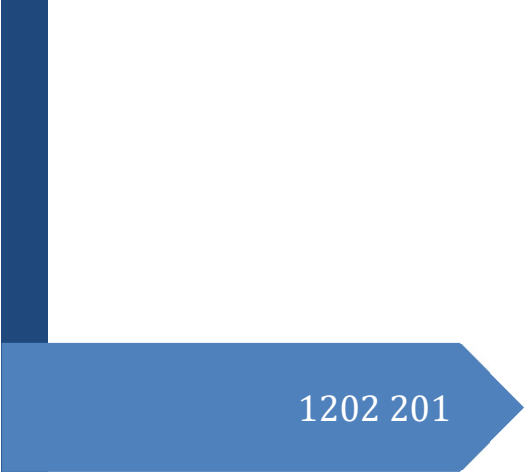




1202 201

เอกสารประกอบการสอนวิชา
การโปรแกรมมิ่งสำหรับงานสารสนเทศ
Programming for Information Work



อาจารย์ ดร.จักรกฤษณ์ แสงแก้ว

หลักสูตรสารสนเทศศาสตร์ คณะวิทยาการสารสนเทศ
มหาวิทยาลัยมหาสารคาม

เอกสารประกอบการสอน
การโปรแกรมมิ่งสำหรับงานสารสนเทศ
(Programming for Information Work)
รหัสวิชา 1202 201

โดย
อาจารย์ ดร.จักรกฤษณ์ แสงแก้ว

หลักสูตรสารสนเทศศาสตร์ ภาควิชาสารสนเทศศาสตร์
คณะวิทยาการสารสนเทศ
มหาวิทยาลัยมหาสารคาม
2564

คำนำ

เอกสารประกอบการสอนรายวิชาการโปรแกรมมิ่งสำหรับงานสารสนเทศ (Programming for Information Work) เล่มนี้จัดทำขึ้นเพื่อใช้สำหรับการเรียนการสอนนิสิตหลักสูตรสารสนเทศศาสตร์บัณฑิต สาขาสารสนเทศศาสตร์ระดับปริญญาตรี แบ่งเนื้อหาการเรียนการสอนออกเป็น 12 บท แต่ละบทใช้เวลาสอนประมาณ 1-2 สัปดาห์

รายวิชานี้มุ่งเน้นให้ผู้เรียนมีความรู้ความสามารถในการโปรแกรมมิ่งสำหรับงานสารสนเทศ ผู้เขียนหวังเป็นอย่างยิ่งว่าเอกสารประกอบการสอนเล่มนี้จะเป็นประโยชน์แก่นิสิต ในรายวิชาการโปรแกรมมิ่งสำหรับงานสารสนเทศและรายวิชาอื่น ๆ ที่เกี่ยวข้อง ตลอดจนผู้สนใจทั่วไป

จักรกฤษณ์ แสงแก้ว

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	๗
สารบัญภาพ	๘
สารบัญตาราง	๘
รายละเอียดรายวิชา (มคอ 3)	๘
บทที่ 1 การเขียนโปรแกรมภาษาไพธอน	1
แนวคิด	1
วัตถุประสงค์	1
1.1 ประวัติภาษาคอมพิวเตอร์.....	1
1.2 การจัดระดับภาษาคอมพิวเตอร์.....	1
1.3 ความนิยมของภาษาคอมพิวเตอร์.....	3
1.4 จุดเด่นของภาษาไพธอน.....	3
1.5 การเขียนโปรแกรมแรกเพื่อแสดงคำว่า "สวัสดีชาวโลก"	4
1.6 การเขียนโปรแกรมแสดงข้อความเดิมหลายครั้ง	4
1.7 การแสดงค่าตัวแปร	5
1.8 รหัสควบคุมรูปแบบการแสดงผล	6
1.9 อักษรควบคุมการแสดงผล.....	6
1.10 ตัวแปร	6
1.11 ชนิดข้อมูลและขนาดจำนวนไบต์.....	7
1.12 การตรวจสอบตัวแปรและอ็อบเจกต์ต่าง ๆ ที่กำลังใช้งาน	8
1.13 การลบตัวแปรและอ็อบเจกต์เพื่อคืนพื้นที่หน่วยความจำให้ระบบปฏิบัติการ (OS).....	8
1.14 การตรวจสอบชนิดข้อมูลภายในตัวแปร	8
1.15 การแปลงตัวแปรเลขจำนวนเต็ม ทศนิยม และข้อความ	9
1.16 การรับข้อมูลจากแป้นพิมพ์มาเก็บในตัวแปร	9
1.17 การเขียนหมายเหตุ (Comment).....	10
แบบฝึกหัดท้ายบท	10

บทที่ 2 เครื่องหมายดำเนินการและนิพจน์	11
แนวคิด	11
วัตถุประสงค์	11
2.1 เครื่องหมายดำเนินการและนิพจน์.....	11
2.2 เครื่องหมายทางคณิตศาสตร์ (Arithmetic Operators)	11
2.3 เครื่องหมายเปรียบเทียบ (Comparison operators)	12
2.4 เครื่องหมายตรรกะ (Logical operators).....	12
2.5 ลำดับและความสำคัญของตัวดำเนินการ	12
2.6 การแปลงเลขฐานสิบและฐานสอง	13
2.7 ลอจิกเกต (Logic gate).....	14
2.8 And Gate	14
2.9 OR Gate	14
2.10 Not Gate.....	15
2.11 วงจรบวกเลข	15
2.12 เลขลบ	16
2.13 การเลื่อนบิตไปทางซ้ายและขวา.....	17
2.14 การเขียนโปรแกรมคำนวณเส้นรอบวงกลม	17
2.15 การเขียนโปรแกรมคำนวณพื้นที่วงกลม	18
2.16 การเขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยม	19
2.17 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้ความยาวด้านทั้งสาม	20
2.18 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้มุมและความยาวสองด้าน	21
2.19 การเขียนโปรแกรมคำนวณรัศมีของวงกลมภายในสามเหลี่ยม	22
2.20 การเขียนโปรแกรมคำนวณพื้นที่รูปหลายเหลี่ยมด้านเท่า	23
2.21 การเขียนโปรแกรมคำนวณปริมาตรทรงกระบอกโดยรับค่ารัศมีและความสูง	24
2.22 การเขียนโปรแกรมคำนวณปริมาตรกรวยโดยรับค่ารัศมีและความสูง	25
2.23 การเขียนโปรแกรมคำนวณปริมาตรทรงกลมและพื้นผิว	26
2.24 การเขียนโปรแกรมคำนวณปริมาตรทรงรี	27

2.25 โปรแกรมคำนวณทางคณิตศาสตร์ MathCAD5	27
แบบฝึกหัดท้ายบท	29
บทที่ 3 ฟังก์ชันและการควบคุม.....	30
แนวคิด	30
วัตถุประสงค์	30
3.1 ความหมายและประเภทของฟังก์ชัน.....	30
3.2 ฟังก์ชันที่สร้างขึ้นด้วยตนเอง.....	30
3.3 การสร้างฟังก์ชันด้วยคำสั่ง lambda	31
3.4 คำสั่ง for.....	32
3.5 คำสั่ง while.....	33
3.6 คำสั่ง if / else.....	33
3.7 คำสั่ง break และ continue	34
3.8 ตัวแปรแบบ Global และ Local	35
3.9 การตรวจสอบความผิดพลาด (Exception Handling).....	36
3.10 การเขียนโปรแกรมคำนวณผลรวมของข้อมูลยกกำลังสองด้วย for().....	37
แบบฝึกหัดท้ายบท	37
บทที่ 4 ตัวแปรลิสต์และทูเพิล.....	38
แนวคิด	38
วัตถุประสงค์	38
4.1 การประกาศตัวแปรชนิดลิสต์.....	38
4.2 ตำแหน่งสมาชิกย่อยภายในลิสต์	38
4.3 การเข้าถึงสมาชิกภายในลิสต์	39
4.4 การตรวจสอบขนาดของลิสต์ด้วยคำสั่ง len()	39
4.5 การวนเข้าไปอ่านค่าภายในลิสต์ด้วยคำสั่ง for	40
4.6 การวนเข้าไปอ่านค่าภายในลิสต์ด้วยคำสั่ง for และ range	40
4.7 การแก้ไขค่าตัวแปรภายในสมาชิกย่อย.....	41
4.8 การเลื่อนลิสต์.....	41

4.9 การลบสมาชิกภายในตัวแปรลิสต์.....	42
4.10 การเพิ่มข้อมูลเข้าไปในลิสต์ด้วยคำสั่ง append.....	43
4.11 การ pop สมาชิกออกจากตัวแปรลิสต์.....	43
4.12 คำสั่ง index() ใช้ตรวจสอบหมายเลขสมาชิกภายในตัวแปรลิสต์.....	43
4.13 คำสั่ง count() สำหรับตรวจสอบจำนวนข้อมูลที่ตรงกับค่าที่กำหนด.....	44
4.14 คำสั่ง sort() สำหรับเรียงลำดับรายการในลิสต์จากน้อยไปมาก	44
4.15 คำสั่ง reverse() ใช้กลับลำดับสมาชิกภายในตัวแปรลิสต์	44
4.16 คำสั่ง set() ใช้กำจัดสมาชิกที่ซ้ำกันภายในลิสต์.....	45
4.17 ตัวแปรทูเพิล (Tuple).....	45
4.18 ฟังก์ชันเกี่ยวกับเวลา.....	46
4.19 ฟังก์ชันการสุ่มตัวเลข.....	46
4.20 ฟังก์ชันคณิตศาสตร์.....	46
แบบฝึกหัดท้ายบท	47
บทที่ 5 การใช้งานสตริง.....	48
แนวคิด	48
วัตถุประสงค์	48
5.1 การประกาศตัวแปรสตริง	48
5.2 การบวกและการคูณสตริง	49
5.3 ตำแหน่งสมาชิกภายในสตริง.....	49
5.4 การตรวจสอบความยาวของสตริง	49
5.5 การวนเข้าไปในสตริงเพื่อเข้าถึงสมาชิกย่อย	50
5.6 การเข้าถึงข้อมูลบางส่วนในสตริง.....	50
5.7 ตัวแปรสตริงเป็น Immutable คือแก้ไขข้อมูลโดยตรงไม่ได้.....	51
5.8 การแทนที่ค่าภายในตัวแปรสตริงด้วยคำสั่ง replace().....	51
5.9 การใช้คำสั่ง in เพื่อตรวจสอบข้อความภายในสตริง	52
5.10 การเติมเลขศูนย์หน้าตัวเลขด้วยคำสั่ง zfill().....	52
5.11 การลบเครื่องหมาย white space ด้านหน้าและหลังตัวแปรสตริงด้วยคำสั่ง strip().....	53

5.12 การแยกสตริงด้วยคำสั่ง split()	53
5.13 การรวมสมาชิกในตัวแปรลิสต์เป็นข้อความด้วยคำสั่ง join()	54
5.14 การค้นหาข้อความย่อยภายในข้อความทั้งหมดด้วยคำสั่ง find() และ rfind()	55
5.15 การค้นหาข้อความภายในข้อความด้วยคำสั่ง index() และ rindex()	56
5.16 การนับจำนวนคำภายในสตริงด้วยคำสั่ง count	56
5.17 การตรวจสอบข้อความถูกต้องตามหลักการตั้งชื่อตัวแปรและฟังก์ชันด้วย isidentifier()	57
5.18 การจัดการเกี่ยวกับตัวพิมพ์ใหญ่และเล็ก	57
5.19 การเปลี่ยนอักษรด้านหน้าคำเป็นตัวพิมพ์ใหญ่ด้วยฟังก์ชัน capwords	58
5.20 การกำหนดตำแหน่งการแสดงผลข้อความ	58
5.21 การตรวจสอบค่าลงท้ายของสตริง	59
5.22 การเข้ารหัสข้อความด้วยคำสั่ง encode()	59
5.23 การค้นหาและแยกข้อความเป็นส่วนด้วยคำสั่ง partition() และ rpartition()	61
5.24 การแทนที่คำด้วย maketrans() และ translate()	61
5.25 การตรวจสอบสตริง	62
5.26 การสกัดอีเมลออกจากเอกสารด้วย Regular Expression	62
5.27 การอ่านข้อมูลจาก Wikipedia เก็บไว้ในตัวแปรสตริง	63
5.28 การตัดคำภาษาไทย	64
5.29 การแปลงอักษรไทยรหัส TIS-620 เป็น UTF-8	64
แบบฝึกหัดท้ายบท	65
บทที่ 6 แฟ้มข้อมูล (File)	66
แนวคิด	66
วัตถุประสงค์	66
6.1 หลักการอ่านและเขียนแฟ้มข้อมูล	66
6.2 การเปิดแฟ้มข้อมูล (Open File)	66
6.3 การอ่านและเขียนแฟ้มข้อมูล (Read and Write File)	67
6.4 บัฟเฟอร์ (Buffer)	68
6.5 ความแตกต่างระหว่างแฟ้มข้อมูลแบบ Binary และ Text Mode	68

6.6 การอ่านไฟล์ในบรรทัดเดียว	69
6.7 อ่านบรรทัดที่ต้องการภายในไฟล์.....	70
6.8 การอ่านรายชื่อไฟล์ภายในไดเรกทอรีด้วย glob().....	70
6.9 การตรวจสอบขนาดไฟล์	70
6.10 การลบไฟล์และไดเรกทอรี.....	71
6.11 ฟังก์ชันภายในอ็อบเจกต์เพิ่มข้อมูล	72
6.12 การอ่านไฟล์ด้วยการระบุตำแหน่งภายในไฟล์ seek และ tell.....	72
6.13 การตัดข้อมูลท้ายไฟล์ทิ้งด้วย truncate()	73
6.14 การอ่านไฟล์จากเว็บด้วย urllib	74
แบบฝึกหัดท้ายบท	74
บทที่ 7 ตัวแปรดิกชันนารีและแม่พริตวีส.....	76
แนวคิด	76
วัตถุประสงค์	76
7.1 การประกาศและเข้าถึงตัวแปรชนิดดิกชันนารี.....	76
7.2 การเพิ่มและแก้ไขข้อมูลในตัวแปรดิกชันนารีด้วยคำสั่ง update().....	77
7.3 การสร้างดิกชันนารีใหม่จากตัวแปรลิสต์	78
7.4 การอ่านข้อมูลภายในตัวแปรดิกชันนารีด้วยคำสั่ง get().....	78
7.5 การ pop ข้อมูลภายในดิกชันนารีด้วยคำสั่ง pop().....	79
7.6 การรวมตัวแปรดิกชันนารีเข้าด้วยกันด้วยคำสั่ง update().....	79
7.7 คำสั่ง map และ lambda.....	80
7.8 คำสั่ง filter	80
7.9 คำสั่ง reduce	81
7.10 การสืบค้นข้อมูลรายชื่อวิทยานิพนธ์ 10,000 รายการ.....	82
แบบฝึกหัดท้ายบท	83
บทที่ 8 การเขียนโปรแกรมภาษาไพธอนสำหรับสถิติเบื้องต้น	84
แนวคิด	84
วัตถุประสงค์	84

8.1 การเขียนโปรแกรมคำนวณค่าเฉลี่ย	84
8.2 การเขียนโปรแกรมคำนวณค่ามัธยฐาน (Median)	85
8.3 การเขียนโปรแกรมคำนวณค่าฐานนิยม (Mode).....	86
8.4 การเขียนโปรแกรมคำนวณค่าเบี่ยงเบนมาตรฐาน	86
8.5 การเขียนโปรแกรมคำนวณค่าความแปรปรวน	87
8.6 การเขียนโปรแกรมคำนวณระดับความเบ้ของข้อมูล (Skewness).....	88
8.7 การเขียนโปรแกรมคำนวณค่าความโด่งของข้อมูล (kurtosis).....	89
8.8 การเขียนโปรแกรมคำนวณพื้นที่ใต้โค้งปกติด้วยการอินทิเกรต.....	90
8.9 การคำนวณพื้นที่ใต้เส้นโค้งปกติ	91
แบบฝึกหัดท้ายบท	94
บทที่ 9 การเขียนโปรแกรมภาษาไพธอนติดต่อบริบบฐานข้อมูล MySQL.....	95
แนวคิด	95
วัตถุประสงค์	95
9.1 ระบบฐานข้อมูล.....	95
9.2 การสร้างตารางด้วย phpmyadmin.....	95
9.3 การติดตั้ง mysql-connector	97
9.4 การเชื่อมต่อฐานข้อมูล MySQL เข้ากับ Python.....	98
9.5 การเพิ่มข้อมูลด้วยคำสั่ง Insert	98
9.6 การสืบค้นด้วยคำสั่ง Select.....	99
9.7 การแก้ไขข้อมูล	99
9.8 การลบข้อมูล	100
9.9 การนำข้อมูลวิทยานิพนธ์ 10,000 รายการใส่ลงในฐานข้อมูล	100
9.10 การนำข้อมูลวิทยานิพนธ์ใส่เข้าไปในตาราง thesis	101
แบบฝึกหัดท้ายบท	102
บทที่ 10 การเขียนโปรแกรมภาษาไพธอนในงานภูมิสารสนเทศ.....	103
แนวคิด	103
วัตถุประสงค์	103

10.1 ระบบพิกัดในแผนที่ (Coordinate System).....	103
10.2 ระบบพิกัดกริด (Grid Coordinate) หรือ UTM (Universal Transverse Mercator).....	104
10.3 โปรแกรม Quantum GIS (QGIS).....	105
10.4 การติดต่อผู้ใช้ด้วยภาพกราฟิก (GUI : Graphic User Interface).....	105
10.5 การเพิ่มแผนที่จังหวัดและแสดงรายชื่อด้วย QGIS.....	106
10.6 เพิ่มข้อมูลอำเภอลงบนแผนที่ด้วย QGIS.....	106
10.7 การนำเข้าข้อมูลภาพถ่ายดาวเทียมพื้นที่ประเทศไทยจากดาวเทียมโมดิส (MODIS)....	106
10.8 การสกัดรายชื่อจังหวัดที่อยู่ภายในพื้นที่รัศมีเรดาร์หาคลิและอมก๋อย	107
10.9 การตรึงภาพถ่ายและแผนที่ด้วย Georeference	110
10.10 การแสดงแผนที่ด้วย folium.....	110
10.11 การใช้งาน GeoPandas.....	116
10.12 การสร้างไฟล์ .shp ด้วย GeoPandas	117
10.13 การเข้ารหัส Province.shp เป็น UTF8.....	117
10.14 การสกัดข้อมูลราชกิจจานุเบกษางบประมาณประจำปี 2564 ลงในแผนที่จังหวัด.....	118
10.15 การปรับโครงสร้างแผนที่จังหวัดและงบประมาณรายจ่ายประจำปี 2564.....	121
10.16 การแสดงแผนที่ตามระดับสีจากงบประมาณรายจ่ายประจำปี 2564	122
10.17 การเรียงลำดับข้อมูล GeoDataFrame ด้วย sort_values().....	124
10.17 การติดตั้งและใช้งาน GeoServer ผ่าน Docker	124
10.18 การเผยแพร่ข้อมูลบน GeoServer	126
แบบฝึกหัดท้ายบท	129
บทที่ 11 การเขียนโปรแกรมภาษาไพธอนในงานประมวลผลภาพ	130
แนวคิด	130
วัตถุประสงค์	130
11.1 พิกเซลบนจอภาพคอมพิวเตอร์ (Pixel).....	130
11.2 โมเดลภาพแบบ RGB (Red Green Blue).....	131
11.3 ค่าตัวเลขในแต่ละพิกเซลของภาพ RGB.....	131
11.4 การเขียนจุด เส้นตรงและวงกลม (Output primitive)	131

11.5 การบวกและลบค่าสี.....	133
11.6 การดำเนินการเกี่ยวกับบิต (Bit).....	135
11.7 การแยกและรวมสี RGB.....	136
11.8 การกลับสีภาพ (Invert Color).....	138
11.9 การเขียนโปรแกรมเพื่อปรับค่าแกมมาของภาพ.....	139
11.10 การเลือกดำเนินการบางส่วนภายในภาพ.....	141
11.11 การกลับภาพซ้ายขวา/บนล่าง.....	142
11.12 คอนโวลูชันและเคอร์เนล (Convolution และ Kernel).....	143
11.13 การตรวจจับใบหน้าด้วย Haar Cascade.....	146
11.14 การตรวจจับใบหน้าขนาดเล็กด้วย Tinyface.....	147
11.15 การอ่านพิกัดแลนด์มาร์คใบหน้าด้วย dlib.....	149
11.16 การตรวจสอบคนในภาพหันหน้าไปทางซ้ายหรือขวา.....	150
11.17 การตรวจจับวัตถุด้วย Yolo3.....	152
แบบฝึกหัดท้ายบท.....	154
บทที่ 12 การเขียนโปรแกรมภาษาไพธอนสำหรับ Machine Learning เบื้องต้น.....	155
แนวคิด.....	155
วัตถุประสงค์.....	155
12.1 ความหมายและประเภทของ Machine Learning.....	155
12.2 การเขียนโปรแกรมเพื่อตรวจสอบเงื่อนไขและแสดงตัวเลขเจ็ดส่วน (7-Segment).....	156
12.3 การเขียนโปรแกรมตรวจสอบเลข 7-Segment ด้วยการเรียนรู้ของเครื่องจักร.....	157
12.4 การบันทึกและโหลดโมเดลกลับมาใช้งาน.....	158
12.5 การสร้างโมเดลด้วยวิธี Simple Linear Regression.....	159
12.6 การสร้างสมการ Simple Linear Regression จากการทดสอบแรงดึงของกาว.....	162
12.7 Metric สำหรับวัดประสิทธิภาพของโมเดล.....	165
12.8 การคำนวณค่าความผิดพลาดของโมเดลด้วย RMSE (Root Mean Square Error).....	165
12.9 โครงข่ายประสาทเทียม (Artificial Neural Network).....	167
12.10 การคำนวณ Single Layer Perceptron ของ And Gate.....	170

12.11 การสร้างโครงข่ายประสาทเทียม Single Layer Perceptron ของ And Gate.....	171
12.11 การสร้างโมเดลจากโครงข่ายประสาทเทียมที่สร้างขึ้น	172
12.11 Gradient Descent.....	173
12.12 การแยกภาพภูเขาและทะเลด้วย Logistic Regression	175
แบบฝึกหัดท้ายบท	178
บรรณานุกรม.....	179

สารบัญภาพ

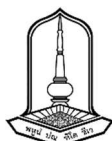
	หน้า
ภาพประกอบที่ 1.1 แผนภาพวิวัฒนาการของภาษาคอมพิวเตอร์	2
ภาพประกอบที่ 1.2 แผนภาพความนิยมของภาษาคอมพิวเตอร์	3
ภาพประกอบที่ 1.3 หน้าต่างการทำงานของโปรแกรม colab.....	4
ภาพประกอบที่ 1.4 แสดงรหัสควบคุมรูปแบบการแสดงผล.....	5
ภาพประกอบที่ 2.1 แสดงค่าประจำหลักของระบบเลขฐานสอง	13
ภาพประกอบที่ 2.2 แนวคิดของ And Gate	14
ภาพประกอบที่ 2.3 แนวคิดของ OR Gate	14
ภาพประกอบที่ 2.4 แนวคิดของ Not Gate	15
ภาพประกอบที่ 2.5 วงจรบวกเลข (Adder Logic).....	16
ภาพประกอบที่ 2.6 แสดงการเปรียบเทียบเลขลบด้วยวิธีการแบบต่าง ๆ.....	16
ภาพประกอบที่ 7.1 แสดงการ reduce ด้วยฟังก์ชันการบวก ลบ และคูณ	81
ภาพประกอบที่ 8.1 แสดงอักขระควบคุม %s และ %.2f แสดงข้อความและทศนิยมสองตำแหน่ง	85
ภาพประกอบที่ 8.2 แสดงเส้นโค้งปกติในช่วงค่า z มีค่า -5 ถึง 5.....	91
ภาพประกอบที่ 8.3 แสดงเส้นโค้งปกติที่มีค่าเฉลี่ย 300 และส่วนเบี่ยงเบนมาตรฐาน 100	92
ภาพประกอบที่ 9.1 สัดส่วนการใช้งานฐานข้อมูลแบบต่าง ๆ.....	95
ภาพประกอบที่ 9.2 การสร้างตารางด้วย phpmyadmin	96
ภาพประกอบที่ 9.3 แสดงการเพิ่มข้อมูลด้วยโปรแกรม phpmyadmin.....	97
ภาพประกอบที่ 9.4 แสดงการติดตั้งโปรแกรม mysql-connector-python	97
ภาพประกอบที่ 9.5 การสร้างตารางวิทยานิพนธ์ thesis	101
ภาพประกอบที่ 10.1 แสดงแนวตั้งเมริเดียน (Meridian) และเส้นศูนย์สูตร (Equator)	104
ภาพประกอบที่ 10.2 พิกัดกริด (Grid Coordinate) หรือ UTM(Universal Transverse Mercator)	104
ภาพประกอบที่ 10.3 ส่วนประกอบของโปรแกรม Qgis	105
ภาพประกอบที่ 10.4 แสดงการเพิ่มข้อมูลแผนที่อำเภอ.....	106
ภาพประกอบที่ 10.5 การเพิ่มชั้นภาพถ่ายดาวเทียม MODIS	107
ภาพประกอบที่ 10.6 แสดงขอบเขตรัศมีเรดาร์ดาวเทียมและอวกาศ	108
ภาพประกอบที่ 10.7 แสดงการเลือกการแอตทริบิวต์ของชั้น province	108
ภาพประกอบที่ 10.8 แสดงการเลือกการแอตทริบิวต์ของชั้น Buffer	109
ภาพประกอบที่ 10.9 แสดงจังหวัดที่อยู่ในพื้นที่ intersec ระหว่าง province และ buffer.....	109
ภาพประกอบที่ 10.10 การตรึงภาพถ่ายดาวเทียม MSAT2 ด้วย Georeferencing	110
ภาพประกอบที่ 10.11 แสดงแผนที่พิกัดตำแหน่งมหาวิทยาลัยมหาสารคาม.....	111
ภาพประกอบที่ 10.12 แสดงการวาดเส้นตรงระหว่าง มมส.เก่าและ มมส.ใหม่.....	113
ภาพประกอบที่ 10.13 แสดงการวาดตรามหาวิทยาลัยลงบนแผนที่และแสดงรายละเอียด.....	115
ภาพประกอบที่ 10.14 แสดงรายละเอียดของไฟล์ province.shp.....	116
ภาพประกอบที่ 10.15 การแสดงผลรายชื่อจังหวัดด้วยการเข้ารหัส UTF-8.....	118

ภาพประกอบที่ 10.16	ผลลัพธ์จากการแปลงเอกสาร PDF เป็นข้อความและทำความสะอาดข้อมูล	120
ภาพประกอบที่ 10.17	แสดงการปรับโครงสร้าง province2564.shp	122
ภาพประกอบที่ 10.18	แสดงแผนที่ระบายสีตามระดับงบรายจ่ายประจำปี 2564	123
ภาพประกอบที่ 10.19	ผลลัพธ์การเรียงลำดับจากคอลัมน์ budget64.....	124
ภาพประกอบที่ 10.20	แสดงสถานะของ GeoServer.....	125
ภาพประกอบที่ 10.21	แสดงการสร้าง workspace.....	126
ภาพประกอบที่ 10.22	แสดงการเพิ่ม Data Source ชนิด Shapefile.....	126
ภาพประกอบที่ 10.23	แสดงการกำหนดการเข้ารหัสตัวอักษร windows-874	127
ภาพประกอบที่ 10.24	การเผยแพร่ shapefile ของ dsdi workspace	127
ภาพประกอบที่ 10.25	แสดงการกำหนดระบบพิกัด CRS แบบ EPSG:32648	128
ภาพประกอบที่ 10.26	การแสดงผลแผนที่บน OpenLayer และทำงานผ่านเว็บ	128
ภาพประกอบที่ 11.1	แสดงการเรียงตัวของพิกเซลบนจอภาพคอมพิวเตอร์.....	130
ภาพประกอบที่ 11.2	แสดงค่าสีภายในภาพ RGB.....	131
ภาพประกอบที่ 11.3	การแสดงตำแหน่งพิกัดของโครงสร้างภาพสองมิติ	132
ภาพประกอบที่ 11.4	แสดงผลลัพธ์การวาดรูปทรงพื้นฐานสองมิติ.....	133
ภาพประกอบที่ 11.5	แสดงผลลัพธ์การบวกและลบค่าสี.....	134
ภาพประกอบที่ 11.6	แสดงการดำเนินการเกี่ยวกับบิต.....	136
ภาพประกอบที่ 11.7	แสดงการแยกและรวมสีภาพด้วย split() และ merge().....	137
ภาพประกอบที่ 11.8	แสดงการกลับสีภาพ (Invert color).....	139
ภาพประกอบที่ 11.9	แสดงผลลัพธ์การปรับค่าแกมมาของภาพ	140
ภาพประกอบที่ 11.10	แสดงการเข้าถึงพื้นที่บางส่วนภายในภาพ.....	142
ภาพประกอบที่ 11.11	แสดงการกลับภาพซ้ายขวา/บนล่างด้วยคำสั่ง flip().....	143
ภาพประกอบที่ 11.12	แสดงการคอนโวลูชัน	144
ภาพประกอบที่ 11.13	แสดงการคอนโวลูชันด้วยเคอร์เนลแบบต่าง ๆ	145
ภาพประกอบที่ 11.14	แสดงการตรวจจับใบหน้าด้วย Haar Cascade	146
ภาพประกอบที่ 11.15	แสดงการตรวจจับใบหน้าขนาดเล็กด้วย tinyface	148
ภาพประกอบที่ 11.16	ตำแหน่งพิกัดของจุดบนใบหน้า (แลนด์มาร์ค).....	149
ภาพประกอบที่ 11.17	แสดงระยะห่างระหว่างตาซ้ายและขวา (e1 และ e2)	151
ภาพประกอบที่ 11.18	แสดงสถาปัตยกรรมการตรวจจับวัตถุของ YOLO.....	152
ภาพประกอบที่ 11.19	แสดงผลลัพธ์การตรวจจับวัตถุด้วย Yolo v3.....	153
ภาพประกอบที่ 12.1	แสดงส่วนประกอบของเลข 7-Segment.....	156
ภาพประกอบที่ 12.2	น้ำหนักรูขี้ผึ้งและเส้นรอบรูขี้ผึ้งจำนวน 10 ตัว	160
ภาพประกอบที่ 12.3	แสดงผลลัพธ์การพล็อตน้ำหนักและความยาวรอบรูขี้ผึ้ง.....	161
ภาพประกอบที่ 12.4	แสดงผลลัพธ์อุณหภูมิและการทนแรงดึงของชิ้นงาน	163
ภาพประกอบที่ 12.5	แสดงเซลล์ประสาทและเซลล์ประสาทเทียม	167
ภาพประกอบที่ 12.6	สถาปัตยกรรมโครงข่ายประสาทเทียมแบบไม่ย้อนกลับและแบบแพร่ย้อนกลับ..	168
ภาพประกอบที่ 12.7	แสดงโครงสร้างประสาทเทียมแบบชั้นเดียว.....	168

ภาพประกอบที่ 12.8	กราฟตัวอย่างของฟังก์ชัน Threshold	169
ภาพประกอบที่ 12.9	กราฟตัวอย่างของฟังก์ชัน Sigmoid	169
ภาพประกอบที่ 12.10	แสดงฟังก์ชันกระตุ้นแบบต่าง ๆ	169
ภาพประกอบที่ 12.11	สถาปัตยกรรมโครงข่ายประสาทเทียมของ And Gate	170
ภาพประกอบที่ 12.12	แสดงการหาค่าต่ำสุดของ loss function ด้วยการหาอนุพันธ์	174
ภาพประกอบที่ 12.13	แสดงการหาค่าต่ำสุดของสมการหลายตัวแปรด้วยวิธี Gradient Descent	174
ภาพประกอบที่ 12.14	แสดงภาพภูเขาและทะเลเพื่อใช้ train model	175
ภาพประกอบที่ 12.15	แสดงภาพภูเขาและทะเลเพื่อใช้ทำนายผลจากโมเดลที่สร้างขึ้น	176

สารบัญตาราง

	หน้า
ตารางที่ 1.1 อักขระและรหัสควบคุมการแสดงผล	6
ตารางที่ 1.2 ขนาดของตัวแปรประเภทต่าง ๆ	7
ตารางที่ 2.1 เครื่องหมายทางคณิตศาสตร์ (Arithmetic Operators)	11
ตารางที่ 2.2 เครื่องหมายเปรียบเทียบ (Comparison operators).....	12
ตารางที่ 2.3 เครื่องหมายเปรียบเทียบ (Comparison operators).....	12
ตารางที่ 2.4 ลำดับความสำคัญของตัวดำเนินการ.....	12
ตารางที่ 2.5 การแปลงเลขฐานสิบเป็นฐานสอง	13
ตารางที่ 2.6 การพิมพ์สูตรและสมการคณิตศาสตร์ด้วย Matcad5.....	28
ตารางที่ 4.1 คำสั่งเกี่ยวกับเวลาภายในโมดูล time	46
ตารางที่ 4.2 ฟังก์ชันการสุ่มภายในโมดูล random.....	46
ตารางที่ 4.3 ฟังก์ชันคณิตศาสตร์และค่าคงที่ภายในโมดูล math	46
ตารางที่ 5.1 การประกาศตัวแปรสตริงด้วยเครื่องหมาย single double และ triple quote	48
ตารางที่ 5.2 การบวกและการคูณสตริง.....	49
ตารางที่ 5.3 แสดงฟังก์ชันการตรวจสอบสตริงแบบต่าง ๆ	62
ตารางที่ 5.4 แสดง Meta Charater ของ Regular Expression	62
ตารางที่ 10.1 ระบบพิกัดภูมิศาสตร์ที่ใช้ในประเทศไทย	104
ตารางที่ 10. 2 ตารางพารามิเตอร์ Folium.....	111
ตารางที่ 12.1 ส่วนประกอบของเลข 7-Segment.....	156
ตารางที่ 12.2 น้ำหนักหมีและเส้นรอบอกหมีจำนวน 10 ตัว.....	160
ตารางที่ 12.3 การคำนวณค่าจากตัวแปร x และ y.....	160
ตารางที่ 12.4 การทดสอบวัดแรงดึงของชิ้นงานที่อุณหภูมิที่แตกต่างกัน.....	163
ตารางที่ 12.5 ตารางความจริงของ And Gate	170
ตารางที่ 12.6 ผลการเรียนรู้ของโครงข่ายประสาทเทียมแต่ละรอบ	171



รายละเอียดของรายวิชา

ชื่อสถาบันอุดมศึกษา มหาวิทยาลัยมหาสารคาม
วิทยาการสารสนเทศ / สารสนเทศศาสตร์

หมวดที่ 1 ข้อมูลโดยทั่วไป

1. รหัสวิชาและชื่อรายวิชา
1202 201 การโปรแกรมมิ่งสำหรับงานสารสนเทศ
(Programming for Information Work)
2. จำนวนหน่วยกิต
3 (2-2-5)
3. หลักสูตรและประเภทรายวิชา
หลักสูตร ศิลปศาสตรบัณฑิต
สาขาวิชา สารสนเทศศาสตร์
ประเภทกลุ่มรายวิชา วิชาเอกบังคับ
4. อาจารย์ผู้รับผิดชอบรายวิชา
อาจารย์ ดร.จักรกฤษณ์ แสงแก้ว
5. ภาคการศึกษา/ชั้นปีที่เรียน
1/2564 ชั้นปีที่ 2
6. รายวิชาที่ต้องเรียนมาก่อน (Pre-requisite) ไม่มี
7. รายวิชาที่ต้องเรียนพร้อมกัน (Co-requisites) ไม่มี
8. สถานที่เรียน เรียนออนไลน์
9. วันที่จัดทำหรือปรับปรุงรายละเอียดของรายวิชาครั้งล่าสุด
พฤษภาคม พ.ศ. 2564

หมวดที่ 2 จุดมุ่งหมายและวัตถุประสงค์

2.1 จุดมุ่งหมายรายวิชา

1. เพื่อให้นิสิตมีความรู้ความเข้าใจเกี่ยวกับหลักการเขียนโปรแกรมภาษาไพธอน
ในงานสารสนเทศศาสตร์
2. เพื่อให้นิสิตเข้าใจเครื่องหมายดำเนินการและนิพจน์
3. เพื่อให้นิสิตเข้าใจฟังก์ชัน การควบคุมและตัวแปรชนิดต่าง ๆ ภายในภาษาไพธอน

4. เพื่อให้นิสิตเข้าใจการเขียนโปรแกรมคำนวณสูตรคณิตศาสตร์และสถิติ
5. เพื่อให้นิสิตสามารถประยุกต์ใช้การโปรแกรมกับงานสารสนเทศได้
6. เพื่อให้นิสิตสามารถเขียนโปรแกรมสำหรับงานสารสนเทศอย่างน้อย 1 ภาษา
7. เพื่อให้นิสิตเข้าใจการแสดงข้อมูลบนแผนที่และการเชื่อมต่อฐานข้อมูล MySQL
8. เพื่อให้นิสิตเข้าใจหลักการประมวลผลภาพและการเรียนรู้ของเครื่องจักร

2.2. วัตถุประสงค์ในการพัฒนาปรับปรุงรายวิชา

1. เพื่อให้นิสิตได้เรียนรู้และนำความรู้ไปใช้ในชีวิตประจำวันได้
2. เพื่อให้ตอบสนองต่อผลการเรียนรู้ของหลักสูตรและปรับปรุงเนื้อหาให้มีความทันสมัยและส่งเสริมการเรียนการสอน

หมวดที่ 3 ลักษณะและการดำเนินการ

3.1 คำอธิบายรายวิชา

ความรู้เบื้องต้นเกี่ยวกับการพัฒนาโปรแกรม ส่วนประกอบของการพัฒนาโปรแกรม หลักการพัฒนาโปรแกรมและการออกแบบอัลกอริธึม โครงสร้างข้อมูลที่ใช้ในการเขียนโปรแกรมและอัลกอริธึม การประยุกต์ใช้กับงานสารสนเทศ ฝึกปฏิบัติเขียนโปรแกรมสำหรับงานสารสนเทศด้วยภาษาโปรแกรม อย่างน้อย 1 ภาษา

Introduction to programming development, programming development component, principles of programming development and algorithm design, data structure for programming and algorithm, apply for information work, programming experiment for information work with less than 1 programming language.

3.2 จำนวนชั่วโมงที่ใช้ต่อภาคการศึกษา

บรรยาย	สอนเสริม	การฝึกปฏิบัติ	การศึกษาด้วยตนเอง
30	ตามความต้องการนิสิตเฉพาะราย	30	75

3.3 จำนวนชั่วโมงต่อสัปดาห์ที่อาจารย์ให้คำปรึกษาและแนะนำทางวิชาการแก่นิสิตเป็นรายบุคคล

1 ชั่วโมงต่อสัปดาห์

3.4 การบูรณาการเรียนการสอนกับงานวิจัย / งานบริการวิชาการ / ทำนุบำรุงศิลปะ (ถ้ามี)

1. การวิจัยเรื่อง ประสิทธิภาพการสืบค้นข้อมูลวิทยานิพนธ์ใน ThaiLIS ด้วยการประมวลผลแบบขนานบนคลัสเตอร์คอมพิวเตอร์โดยใช้เทคนิคแม็ปรีดิวซ์ https://www.tci-thaijo.org/index.php/tla_research/article/download/97842/83883

หมวดที่ 4 การพัฒนาการเรียนรู้ของนิสิต

การพัฒนาผลการเรียนในกลุ่มมาตรฐานผลการเรียนรู้ที่มุ่งหวังตามหลักสูตร

4.1 คุณธรรม จริยธรรม

4.1.1 คุณธรรม จริยธรรมที่ต้องพัฒนา

- 4.1.1.1 ตระหนักในคุณค่าและคุณธรรม จริยธรรม เสียสละ และซื่อสัตย์สุจริต
- 4.1.1.2 มีวินัย ตรงต่อเวลา และความรับผิดชอบต่อตนเอง วิชาชีพและสังคม
- 4.1.1.3 สามารถวิเคราะห์ผลกระทบจากการใช้เทคโนโลยีสารสนเทศต่อบุคคลองค์กรและสังคม

4.1.2 วิธีการสอน

- 4.1.2.1 สอดแทรกคุณธรรมจริยธรรม การเสียสละและซื่อสัตย์สุจริต
- 4.1.2.2 ชี้แจงเรื่องระเบียบวินัย การปฏิบัติตนและความรับผิดชอบต่อตนเองและสังคม
- 4.1.2.3 ให้คิดวิเคราะห์ผลกระทบการใช้เทคโนโลยีสารสนเทศต่อบุคคล องค์กรและสังคม

4.1.3 วิธีการประเมินผล

- 4.1.3.1 ประเมินจากการเข้าชั้นเรียนตรงเวลาและการปฏิบัติตามกิจกรรมในกรอบเวลาที่ได้รับมอบหมาย
- 4.1.3.2 ประเมินจากการให้สถิติวิเคราะห์ผลกระทบของเทคโนโลยีการโปรแกรมมิ่งในงานสารสนเทศที่มีผลกระทบต่อบุคคล องค์กรและสังคม

4.2 ความรู้

4.2.1 ความรู้ที่ต้องได้รับ

- 4.2.1.1 มีความรู้ความเข้าใจเกี่ยวกับหลักการและทฤษฎีที่สำคัญในการเขียนโปรแกรมสารสนเทศศาสตร์
- 4.2.1.2 มีความรู้และความเข้าใจในการวิเคราะห์ปัญหาในงานสารสนเทศศาสตร์
- 4.2.1.3 มีความรู้ความเข้าใจในการวิเคราะห์ ออกแบบ ติดตั้ง ปรับปรุงและประเมินองค์ประกอบต่าง ๆ ของระบบสารสนเทศได้
- 4.2.1.4 มีความรู้ความเข้าใจในความก้าวหน้าทางวิทยาการเทคโนโลยีสารสนเทศที่ทันสมัย

4.2.2 วิธีการสอน (โดยเน้นผู้เรียนเป็นศูนย์กลาง)

- 4.2.2.1 เน้นผู้เรียนเป็นศูนย์กลางและเรียนรู้จากการปฏิบัติ (Learning by doing)
- 4.2.2.2 แทรกหลักการทฤษฎีการเขียนโปรแกรม ให้วิเคราะห์ปัญหาและแนวทางแก้ปัญหา
- 4.2.2.3 เพิ่มเนื้อหาให้มีความทันสมัยได้แก่การเขียนโปรแกรมในงานภูมิสารสนเทศ

4.2.3 วิธีการประเมินผล

- 4.2.3.1 สอบกลางภาคและปลายภาค
- 4.2.3.2 การประเมินจากการปฏิบัติงานของนิสิตในชั้นเรียน
- 4.2.3.3 ประเมินจากโครงงานประจำรายวิชา

4.3 ทักษะทางปัญญา

4.3.1 ทักษะทางปัญญาที่ต้องพัฒนา

4.3.1.1 คิดอย่างมีวิจารณญาณและอย่างเป็นระบบ

4.3.1.2 สามารถประยุกต์ความรู้และทักษะทางเทคโนโลยีสารสนเทศกับการแก้ไขปัญหาได้

4.3.2 วิธีการสอน

4.3.2.1 คิดอย่างมีวิจารณญาณและอย่างเป็นระบบ

4.3.2.2 สามารถประยุกต์ความรู้และทักษะทางเทคโนโลยีสารสนเทศกับการแก้ไขปัญหาได้

4.3.3 วิธีการประเมินผล

4.3.3.1 ประเมินจากสอบกลางภาคและปลายภาค และการปฏิบัติงานที่ได้รับมอบหมาย

4.3.3.2 ประเมินจากการปฏิบัติกิจกรรมในห้องปฏิบัติการคอมพิวเตอร์

4.3.3.2 ประเมินจากโครงงานประจำรายวิชา

4.4 ทักษะความสัมพันธ์ระหว่างบุคคลและความรับผิดชอบ

4.4.1 ทักษะความสัมพันธ์ระหว่างบุคคลและความรับผิดชอบที่ต้องพัฒนา

4.3.1.1 มนุษยสัมพันธ์ที่ดีกับผู้ร่วมงานในองค์กร และบุคคลทั่วไป

4.3.1.2 มีความรับผิดชอบในการกระทำของตนเองและรับผิดชอบงานในกลุ่ม

4.4.2 วิธีการสอน

4.3.2.1 ให้นิสิตทำงานกลุ่มและแลกเปลี่ยนความคิดเห็นระหว่างกันและกัน

4.4.3 วิธีการประเมินผล

4.3.3.1 สังเกตพฤติกรรมและการแสดงออกของนิสิตในการทำงานร่วมกันเป็นกลุ่ม

4.5 ทักษะการวิเคราะห์เชิงตัวเลข การสื่อสาร และการใช้เทคโนโลยีสารสนเทศที่ต้องพัฒนา

4.5.1 ทักษะการวิเคราะห์เชิงตัวเลข การสื่อสาร และการใช้เทคโนโลยีสารสนเทศ

4.5.1.1 สามารถใช้สารสนเทศและเทคโนโลยีสารสนเทศอย่างเหมาะสม

4.5.2 วิธีการสอน

4.5.2.1 ออกแบบการสอนเน้นให้ฝึกคิดวิเคราะห์และแก้ปัญหาทางตัวเลข อาทิ สูตรทางคณิตศาสตร์และการเขียนโปรแกรมเพื่อแก้ปัญหา

4.5.3 วิธีการประเมินผล

4.5.3.1 ประเมินจากการสอบปลายภาค/กลางภาค

4.5.3.1 ประเมินจากการปฏิบัติกิจกรรมในห้องปฏิบัติการคอมพิวเตอร์

หมวดที่ 5 แผนการสอนและการประเมินผล

5.1 แผนการสอน

ลำดับที่	หัวข้อ/รายละเอียด	จำนวนชั่วโมง		กิจกรรม การเรียนการสอน	สื่อการสอน	ผู้สอน
		ทฤษฎี	ปฏิบัติ			
1	ยินดีต้อนรับสู่วิชาโปรแกรมมิ่ง แนวคิด วัตถุประสงค์ ประวัติ คุณลักษณะเด่นของภาษา ตัว แปลภาษา คำสั่งแสดงผล เงื่อนไข การวนรอบ	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา เรียนรู้ผ่านระบบ ออนไลน์	- เอกสารประกอบการ สอน - สไลด์ประกอบการ บรรยาย - เว็บไซต์รายวิชา https://dsdi.msu.ac.th/ ?article=programming	จักรกฤษณ์
2	พื้นฐานการคำนวณด้วย Mathcad ฟังก์ชันทาง คณิตศาสตร์ การพล็อตกราฟ ฟังก์ชันการคำนวณ การหา อนุพันธ์และอินทิเกรต	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
3	องค์ประกอบพื้นฐานของ ภาษาไพธอน การเขียนคำสั่ง เครื่องหมายดำเนินการ ตัว แปรลิสต์ การรับค่าจาก คีย์บอร์ด การพล็อตกราฟ	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
4	เครื่องหมายดำเนินการและ นิพจน์ การเปรียบเทียบ ตรรกะความเป็นเหตุผล เลขฐานสอง วงจรบวกเลข เลขลบ การเขียนโปรแกรม คำนวณสูตรทางคณิตศาสตร์	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
5	การใช้งานสตริงและ แฟ้มข้อมูล การประกาศตัว แปรสตริง การตัดคำ ภาษาไทย การอ่านข้อมูลจาก เว็บไซต์เก็บไว้ในตัวแปรสตริง การอ่านเขียนไฟล์	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
6	ฟังก์ชันและการควบคุม ตัว แปร global และ local การ ควบคุม คำสั่ง range คำสั่ง for คำสั่ง if / else คำสั่ง break / continue คำสั่ง while	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
7	ตัวแปรดิกชันนารี และแม่พรีดิคส์ การอัปเดต การแสดงคีย์ การแม่พฟังก์ชัน การรีดิคส์ การฟิลเตอร์ การ สักรัศมีนิพจน์และสับคั่น	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
8	สอบกลางภาค	3				จักรกฤษณ์

สัปดาห์ที่	หัวข้อ/รายละเอียด	จำนวนชั่วโมง		กิจกรรม การเรียนการสอน	สื่อการสอน	ผู้สอน
		ทฤษฎี	ปฏิบัติ			
9	การเขียนโปรแกรมติดต่อฐานข้อมูล MySQL การสร้างตาราง การเพิ่ม ลบ แก้ไข และสืบค้นผ่านโปรแกรมอินเทอร์เน็ตเฟสกับข้อมูลวิทยานิพนธ์	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา	- เอกสารประกอบการสอน - สไลด์ประกอบการบรรยาย - เว็บไซต์รายวิชา https://dsdi.msu.ac.th/?article=programming	จักรกฤษณ์
10	การเขียนโปรแกรมคำนวณสถิติ ค่าเฉลี่ย มัธยฐาน ฐานนิยม เบี่ยงเบนมาตรฐาน ความแปร ความโค้ง การอินทิเกรตพื้นที่ใต้โค้งปกติ	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
11	การเขียนโปรแกรมภาษาไพธอนในงานภูมิสารสนเทศ ระบบพิกัด ซอฟต์แวร์ GIS GeoPandas และ Folium การสร้าง shapfile การสกัดข้อมูลงบประมาณรายจ่ายประจำปี 2564 ลงในแผนที่ การระบายพล็อตระดับสีจากงบประมาณ 2564	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
12	การประมวลผลภาพด้วยไพธอน การดำเนินการระดับพิกเซลกับภาพ การคอนโวลูชัน และเคอร์เนล การตรวจจับใบหน้าขนาดเล็ก การตรวจจับวัตถุด้วย YOLOv3	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
13 - 14	การเรียนรู้ของเครื่องจักรการสร้างโมเดลด้วย simple linear regression การตรวจจับเลขเจ็ดส่วนด้วยวิธีเก่าและวิธีการเรียนรู้ด้วยเครื่องจักร เมตริกซ์วัดประสิทธิภาพโมเดล การสร้างโครงข่ายประสาทเทียม Gradient Descent การแยกภาพภูเขาและทะเลด้วย Logistic Regression	4	4	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต - บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		จักรกฤษณ์
15	นำเสนอโครงงาน	2	2	- บรรยาย - ตั้งโจทย์/ซักถามนิสิต		จักรกฤษณ์

สัปดาห์ที่	หัวข้อ/รายละเอียด	จำนวนชั่วโมง		กิจกรรม การเรียนการสอน	สื่อการสอน	ผู้สอน
		ทฤษฎี	ปฏิบัติ			
				- บันทึกวีดิโอการสอน นิสิตสามารถศึกษา		
16	สอบปลายภาค	3				จักรกฤษณ์

5.2 แผนการประเมินผลการเรียนรู้

5.2.1 วิธีการ

ผลการเรียนรู้	วิธีการประเมิน	สัปดาห์ที่ ประเมิน	สัดส่วนของการ ประเมิน
1	การประเมินเข้าชั้นเรียนและจิตพิสัย	ทุกสัปดาห์	5 %
2, 3, 5	สอบกลางภาค, สอบปลายภาค	8, 16	20, 20 %
2, 3	กิจกรรม, การบ้าน	ทุกสัปดาห์ยกเว้น การสอบ	40 %
4, 5	โครงงานรายวิชา	15	15 %

5.2.2 เกณฑ์ผ่านรายวิชา ผู้ที่จะผ่านรายวิชานี้จะต้อง

5.2.2.1 มีเวลาเข้าชั้นเรียนไม่ต่ำกว่าร้อยละ 80 ของเวลาเรียน

5.2.2.2 ได้คะแนนรวมทั้งรายวิชาไม่ต่ำกว่าร้อยละ 50 ของคะแนนรวม

5.2.3 เกณฑ์ค่าระดับคะแนน

5.2.2.1 ผู้ที่ไม่ผ่านเกณฑ์ข้อ 5.2.2.2 จะได้รับค่า ระดับคะแนน F

5.2.2.2 ผู้ที่ผ่านเกณฑ์ข้อ 5.2.2.2 จะได้รับค่าระดับคะแนน A B+ B C+ C D+ และ D

ประเมินผลตามเกณฑ์ที่กำหนด

คะแนน 80 – 100 เกรด A

คะแนน 75 – 79 เกรด B+

คะแนน 70 – 74 เกรด B

คะแนน 65 – 69 เกรด C+

คะแนน 60 – 64 เกรด C

คะแนน 55 – 59 เกรด D+

คะแนน 50 – 54 เกรด D

คะแนนต่ำกว่า 50 เกรด F

หมวดที่ 6 ทรัพยากรประกอบการเรียนการสอน

6.1 เอกสารและตำราหลัก

- จักรกฤษณ์ แสงแก้ว, การเขียนโปรแกรมภาษาไพธอนด้วยตนเอง, สำนักพิมพ์ ส.ส.ท. – สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2549

- จักรกฤษณ์ แสงแก้ว, การเขียนโปรแกรมภาษาไพธอนด้วยตนเอง, สำนักพิมพ์ ส.ส.ท. – สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2551

6.2. เว็บไซต์รายวิชาโปรแกรมมิ่งในงานสารสนเทศ

<http://dsdi.msu.ac.th/?article=programming>

หมวดที่ 7 การประเมินและปรับปรุงการดำเนินการของรายวิชา

7.1 กลยุทธ์การประเมินประสิทธิผลของรายวิชาโดยนิสิต

การประเมินประสิทธิผลในรายวิชานี้จัดโดยมหาวิทยาลัยมหาสารคาม ให้นิสิตเข้าประเมินผลการเรียนการสอนผ่านออนไลน์ โดยนำความคิดเห็นจากนิสิต ดังนี้ 1) แบบประเมินผู้สอน 2) ให้นิสิตประเมินด้วยการเขียนลงบนกระดาษโดยไม่ระบุตัวบุคคล

7.2 กลยุทธ์การประเมินการสอน

ใช้กลยุทธ์ในการเก็บข้อมูลเพื่อประเมินการสอนดังนี้ 1) การประเมินผู้สอนและผลการเรียนของนิสิต 2) การทวนสอบผลประเมินผลการเรียนรู้

7.3 การปรับปรุงการสอน

หลังจากได้รับผลการประเมินการสอนในข้อ 2 ทำการปรับปรุงเนื้อหาให้เนื้อหามีความทันสมัยและประยุกต์ใช้ได้มากยิ่งขึ้น

7.4 การทวนสอบมาตรฐานผลสัมฤทธิ์ของนิสิตในรายวิชา

การทวนสอบผลสัมฤทธิ์จากคณะกรรมการในภาควิชา ได้แก่ 1) ตรวจสอบผลการประเมินการเรียนรู้ของนิสิต (คะแนน/เกรด) 2) ข้อสอบ รายงาน โครงการ การให้คะแนนสอบ และการให้คะแนนพฤติกรรม

7.5 การดำเนินการทบทวนและการวางแผนปรับปรุงประสิทธิผลของรายวิชา

ในทางทฤษฎีปรับปรุงรายวิชาทุก 5 ปี หรือตามข้อเสนอแนะปรับปรุงการสอนในข้อ 3 และผลการทวนสอบมาตรฐานผลสัมฤทธิ์ตามข้อ 4 แต่ในทางปฏิบัติผู้สอนมีการปรับปรุงเนื้อหาการสอนทุกปีเพื่อให้เนื้อหาทันสมัยและทันต่อความเปลี่ยนแปลงของโลกาภิวัตน์

บทที่ 1

การเขียนโปรแกรมภาษาไพธอน

แนวคิด

ทักษะการเขียนโปรแกรมเป็นทักษะที่จำเป็นในอนาคต ไม่ว่าจะเป็นผู้เรียนสายไหน ทำงานเกี่ยวกับอะไร หรืออายุเท่าไร ต่อไปจะเริ่มเรียนรู้และหัดเขียนโปรแกรมมากขึ้น ส่งผลให้ตลาดแรงงานโปรแกรมเมอร์มีการแข่งขันสูงขึ้น ในปัจจุบันภาษาไพธอนได้รับความนิยมอย่างแพร่หลาย ทั้งในงานวิทยาศาสตร์ข้อมูล งานด้านการประมวลผลภาพและการเรียนรู้ของเครื่องจักร นอกจากนั้นความก้าวหน้าด้านไมโครคอนโทรลเลอร์รุ่นใหม่อนุญาตให้นำภาษาไพธอนไปทำงานบนฮาร์ดแวร์เหล่านั้นได้และนำไปสู่การพัฒนา IoT (Internet Of Thing) ทำให้สามารถประยุกต์ใช้งานได้อย่างกว้างขวาง

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ ความรู้เบื้องต้นการเขียนโปรแกรมภาษาไพธอน ประวัติภาษาคอมพิวเตอร์ องค์ประกอบพื้นฐานของโปรแกรมภาษาไพธอน การเขียนคำสั่ง การเขียนหมายเหตุ รูปแบบการเขียนโปรแกรมภาษาไพธอน อักขระที่ใช้ในภาษาไพธอน ข้อมูลและชนิดข้อมูล ตัวแปร การกำหนดค่าให้กับตัวแปร การแสดงผลลัพธ์ภายในตัวแปร การตรวจสอบตัวแปรและอ็อบเจ็กต์ต่าง ๆ ที่กำลังใช้งาน การตรวจสอบชนิดตัวแปร การแปลงชนิดตัวแปร การลบตัวแปรและการรับข้อมูลจากคีย์บอร์ด

วัตถุประสงค์

1. อธิบายความเป็นมาของภาษาคอมพิวเตอร์จากแผนภาพวิวัฒนาการภาษาโปรแกรมมิ่งได้
2. อธิบายองค์ประกอบพื้นฐานของโปรแกรมภาษาไพธอนได้
3. อธิบายการเขียนหมายเหตุและอักขระที่ใช้ในภาษาไพธอนได้
4. อธิบายชนิดข้อมูล ตัวแปรและแสดงผลลัพธ์ภายในตัวแปรได้
5. อธิบายการตรวจสอบชนิดตัวแปร ลบตัวแปร แปลงตัวแปรและรับค่าจากผู้ใช้เก็บลงตัวแปรได้

1.1 ประวัติภาษาคอมพิวเตอร์

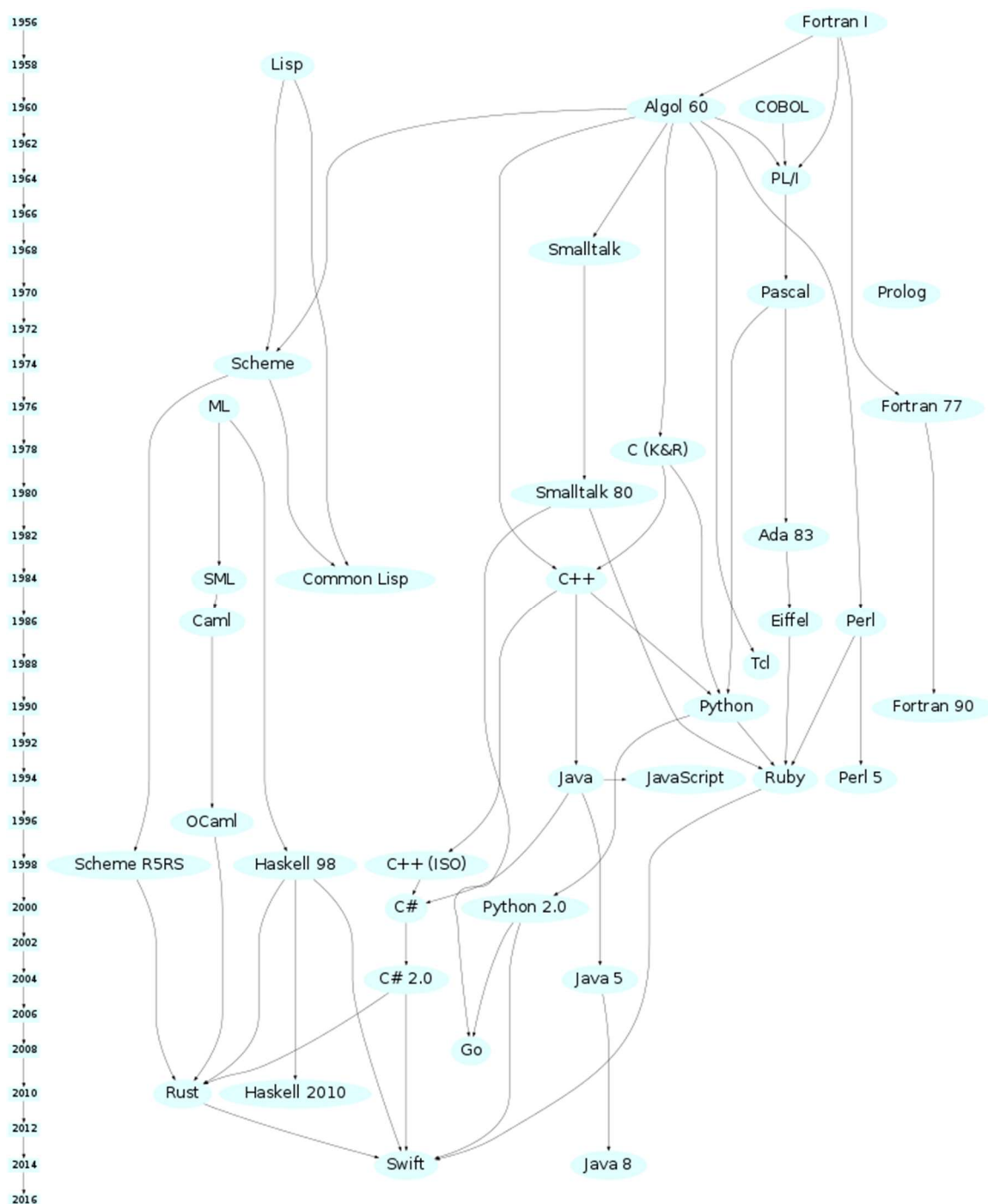
ความเป็นมาของภาษาคอมพิวเตอร์ระดับสูงเริ่มจากคริสต์ทศวรรษ 1950 บริษัท IBM ได้พัฒนาภาษาฟอร์แทรน และในปี ค.ศ. 1958 สถาบันเทคโนโลยีแมสซาชูเซตส์ได้พัฒนาภาษา LISP และต่อมาได้มีการพัฒนาภาษาคอมพิวเตอร์ออกมาเป็นจำนวนมาก แสดงดังภาพประกอบที่ 1.1

1.2 การจัดระดับภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์แบ่งระดับออกเป็น 2 ระดับ คือ ภาษาระดับต่ำ (Low Level Language) และภาษาระดับสูง (High Level Language)

1) ภาษาระดับต่ำมี 2 ภาษา คือ ภาษาเครื่อง (Machine Language) เขียนคำสั่งจากหน่วยประมวลผลกลางด้วยรหัสเลขฐานสอง และภาษาแอสเซมบลีซึ่งเป็นการเปลี่ยนภาษาเครื่องที่อยู่ในรหัสฐานสองมาเป็นคำสั่งเพื่อให้เขียนโปรแกรมง่ายขึ้นแต่ยังจำเป็นต้องเข้าใจโครงสร้างของหน่วยประมวลผลกลางที่กำลังเขียนเป็นอย่างดี

2) ภาษาระดับสูง (High Level Language) ถูกสร้างขึ้นเพื่อช่วยให้การเขียนโปรแกรมทำได้ง่ายขึ้น เช่น ในปี ค.ศ. 1956 ได้มีการพัฒนาภาษา Fortran I และในเวลาได้มีการพัฒนาภาษาต่าง ๆ ออกมาเป็นจำนวนมาก ดังภาพประกอบที่ 1.1



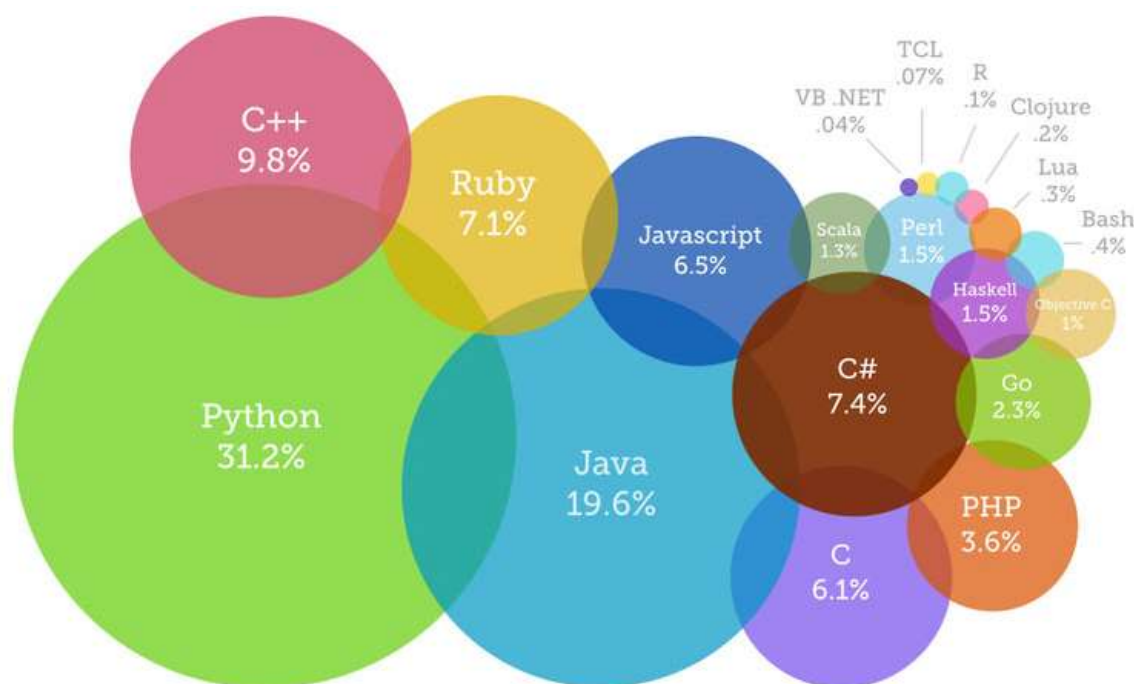
ภาพประกอบที่ 1.1 แผนภาพวิวัฒนาการของภาษาคอมพิวเตอร์

ที่มา: <http://rigaux.org/language-study/diagram.html>

1.3 ความนิยมของภาษาคอมพิวเตอร์

ในปี พ.ศ. 2563 ภาษาไพธอน (python) ได้รับการจัดให้เป็น “ภาษาโปรแกรมมิ่งแห่งปี” โดยได้รับความนิยมเพิ่มขึ้นอย่างก้าวกระโดดในรอบเกือบ 20 ปี ตั้งแต่มีการเก็บสถิติ Programming Language ของ TIOBE programming community index ซึ่งคำนวณจากจำนวนผลการค้นหาจาก Google ,Google Blogs, MSN ,Yahoo, Baidu , Wikipdia และ Youtube

เหตุผลที่ภาษาไพธอนได้รับความนิยมแบบก้าวกระโดดเพราะในปัจจุบันภาษา python ถูกนำมาใช้งานอย่างแพร่หลายในสายงานดาต้า อย่างเช่น Data Science , Machine Learning และยังใช้ไพธอนในงานการพัฒนาเว็บและโมบายแอปพลิเคชันอีกด้วย



ภาพประกอบที่ 1.2 แผนภาพความนิยมของภาษาคอมพิวเตอร์
ที่มา: <https://www.tiobe.com/>

1.4 จุดเด่นของภาษาไพธอน

ภาษาไพธอนมีจุดเด่นหลายประการ แสดงได้ดังนี้

- ภาษาไพธอนเป็นซอฟต์แวร์ระบบเปิด (Open Source) ใช้งานได้โดยไม่เสียค่าลิขสิทธิ์ซอฟต์แวร์
- ภาษาไพธอนมีรูปแบบการเขียนที่เข้าใจง่ายเป็นระเบียบ
- คำสั่งต่าง ๆ ช่วยให้การเขียนโปรแกรมสั้นลงและอำนวยความสะดวก
- ตัวแปรสร้างขึ้นได้ง่ายและใช้งานโดยไม่ต้องประกาศชนิดของตัวแปร
- นำไปใช้กับระบบปฏิบัติการได้หลายแพลตฟอร์ม
- เครื่องมือที่เพิ่มเติมหรือเรียกว่าไลบรารี (library) มีอยู่เป็นจำนวนมาก
- ซอฟต์แวร์หลายตัวนำมาเป็นสคริปต์ให้เรียกใช้ เช่น Maya, QGIS, Fusion 360, Unity เป็นต้น
- เครื่องมือสำหรับการทำงานให้เร็วขึ้นด้วย ระบบการประมวลผลแบบขนาน (Parallel)
- บริษัทกูเกิ้ลให้บริการ <https://colab.research.google.com> เพื่อเขียนโปรแกรมภาษาไพธอน

1.5 การเขียนโปรแกรมแรกเพื่อแสดงคำว่า "สวัสดีชาวโลก"

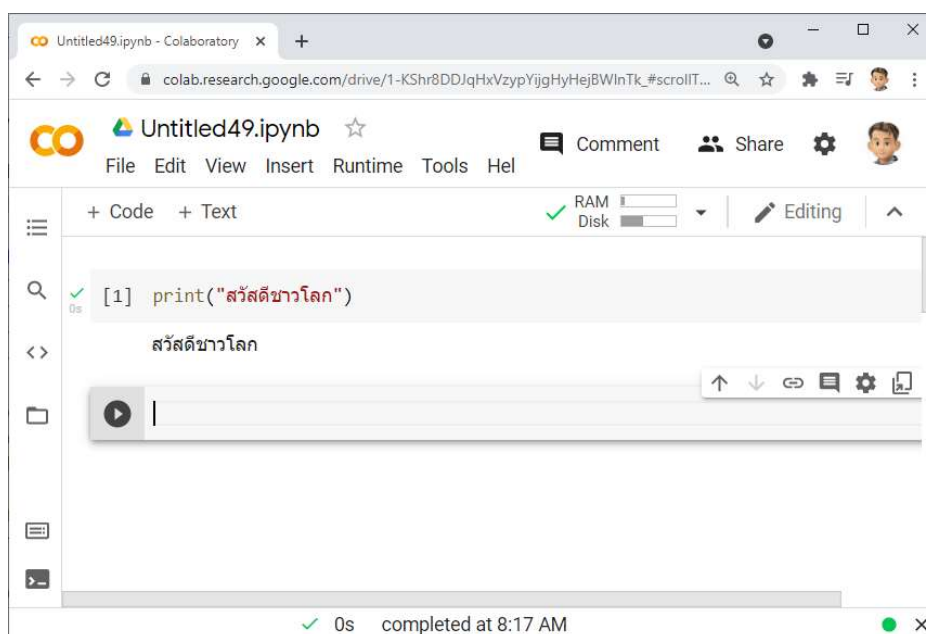
ในการเขียนโปรแกรมของเอกสารฉบับนี้จะเขียนบน google colab ซึ่งทำงานผ่านเว็บเบราว์เซอร์ ดังนั้น ท่านต้องเชื่อมต่ออินเทอร์เน็ตและเป็นสมาชิกอีเมลล์ของ gmail.com โดยให้ล็อกอินเข้าไปที่เว็บ <https://colab.research.google.com> หลังจากนั้น พิมพ์คำสั่ง `print("สวัสดีชาวโลก")` จากนั้นกดปุ่ม ▶ เพื่อเริ่มต้นการทำงาน หลังจากประมวลผลเสร็จเรียบร้อย ผลลัพธ์จะถูกแสดงด้านล่างของคำสั่ง ในตัวอย่างนี้แสดงคำว่า "สวัสดีชาวโลก"

ตัวอย่างที่ 1.1 การแสดงข้อความด้วยคำสั่ง print

```
print("สวัสดีชาวโลก")
```

ผลลัพธ์ :

```
สวัสดีชาวโลก
```



ภาพประกอบที่ 1.3 หน้าต่างการทำงานของโปรแกรม colab

1.6 การเขียนโปรแกรมแสดงข้อความเดิมหลายครั้ง

เครื่องหมาย * ใช้แทนการคูณ ถ้านำข้อความมาคูณกับตัวเลข จะเป็นข้อความเดิมหลายข้อความ ขึ้นอยู่กับตัวเลข เช่น `print("สวัสดีชาวโลก " * 10)` ผลลัพธ์จะแสดงคำว่า สวัสดีชาวโลก ออกมา 10 ครั้ง ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1.2 การแสดงข้อความเดิมหลายครั้ง

```
print("สวัสดีชาวโลก " * 10)
```

ผลลัพธ์ :

สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก
 สวัสดีชาวโลก สวัสดีชาวโลก สวัสดีชาวโลก

1.7 การแสดงค่าตัวแปร

ตัวแปรพื้นฐานได้แก่ 1) ตัวแปรเลขจำนวนเต็ม 2) ตัวแปรทศนิยม 3) ตัวแปรข้อความหรือสตริง (ด้านหน้าและด้านหลังข้อความปิดด้วยเครื่องหมาย คำพูด (")

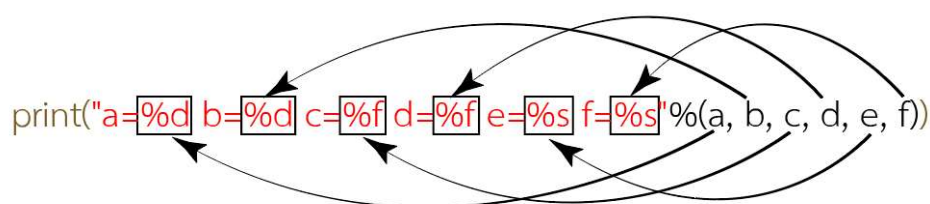
ตัวอย่างที่ 1.3 การแสดงค่าภายในตัวแปร

```
a = 3
b = 2
c = 3.14
d = 2.71
e = "Google"
f = "Youtube"
print("a=%d b=%d c=%f d=%f e=%s f=%s"%(a,b,c,d,e,f))
```

ผลลัพธ์ :

a=3 b=2 c=3.140000 d=2.710000 e=Google f=Youtube

ตัวอย่างที่ 1.3 กำหนดให้ตัวแปร a และตัวแปร b เป็นชนิดเลขจำนวนเต็ม มีค่าเท่ากับ 3 และ 2 ตามลำดับ ตัวแปร c และ d กำหนดให้เป็นทศนิยม มีค่า 3.14 และ 2.71 ตามลำดับ ตัวแปร e และ f กำหนดให้เป็นชนิดข้อความ เก็บคำว่า "Google" และ "Youtube" ตามลำดับ



ภาพประกอบที่ 1.4 แสดงรหัสควบคุมรูปแบบการแสดงผล

การแสดงผลตัวแปรจะใช้ร่วมกับรหัสแสดงค่าตัวแปร ดังตารางที่ 1.1 เมื่อ %d คือการแสดงผลเลขจำนวนเต็ม %s แสดงตัวแปรข้อความ และ %f แสดงตัวเลขทศนิยม ดังนั้น หากต้องการแสดงคำว่า "a=%d b=%d c=%f d=%f e=%s f=%s" เมื่อ %d สองตัวด้านซ้ายมือสุดจะแสดงผลเลขจำนวนเต็มภายในตัวแปร a และ b ส่วน %f สองตัวถัดมาจะแสดงผลเลขทศนิยมในตัวแปร c และ d %f ที่เหลือจะแสดงผลข้อความในตัวแปร e และ f มาแสดงผล ดังนั้น ผลลัพธ์จะแสดงคำว่า a=3 b=2 c=3.140000 d=2.710000 e=Google f=Youtube

1.8 รหัสควบคุมรูปแบบการแสดงผล

รหัสควบคุมรูปแบบ (format code) ใช้สำหรับควบคุมการแสดงผลตัวแปรหรือนิพจน์ออกทางจอภาพโดยรหัสควบคุมรูปแบบในภาษา python มีอยู่หลายชนิด ได้แก่ เลขจำนวนเต็ม ทศนิยม สตริง การเลือกนำไปใช้งานต้องพิจารณาให้เหมาะสมกับค่าของข้อมูลที่ต้องการแสดงผล

ตัวอย่างที่ 1.4 การแสดงรหัสควบคุมรูปแบบการแสดงผล

```
x = 3 ; y = 3.14 ; z = "Google"
print("x=%d y=%.2f z=%s"%(x,y,z))
```

ตัวอย่างที่ 1.4 เป็นการประกาศตัวแปรเลขจำนวนเต็ม ทศนิยม ข้อความ โดยแสดงผลด้วยรหัสควบคุมเลขจำนวนเต็ม (%d) ทศนิยม(%f) และสตริง (%s) ผลลัพธ์มีค่า x=3 y=3.14 z=Google

1.9 อักขระควบคุมการแสดงผล

ในการแสดงผลข้อมูลจำเป็นต้องใช้อย่างยิ่งใช้อักขระพิเศษเพื่อควบคุมการแสดงผล (Carriage Control) ซึ่งจะช่วยให้การแสดงผลนั้นสมบูรณ์มากขึ้น ซึ่งในภาษาไพธอนมีอักขระควบคุมและรหัสควบคุมการแสดงผล ดังนี้

ตารางที่ 1.1 อักขระและรหัสควบคุมการแสดงผล

อักขระ/รหัสควบคุม	หน้าที่	ความหมาย
\n	อักขระควบคุม	ขึ้นบรรทัดใหม่
\t	อักขระควบคุม	เว้นช่องว่างเป็นระยะ 1 แท็บ (6 ตัวอักษร)
\r	อักขระควบคุม	กำหนดให้เคอร์เซอร์ไปอยู่ต้นบรรทัด
\f	อักขระควบคุม	เว้นช่องว่างเป็นระยะ 1 หน้าจอ
\b	อักขระควบคุม	ลบอักขระสุดท้ายออก 1 ตัว
\\	อักขระควบคุม	แทนเครื่องหมาย \
\"	อักขระควบคุม	แทนเครื่องหมาย "
\'	อักขระควบคุม	แทนเครื่องหมาย '
%c	รหัสควบคุม	ตัวแปรอักขระตัวเดียว เช่น 'a' , 'b'
%s	รหัสควบคุม	ตัวแปรสตริง (ข้อความ) เช่น "hello"
%d	รหัสควบคุม	ตัวแปรเลขจำนวนเต็ม เช่น 300
%e	รหัสควบคุม	ตัวแปรทศนิยมในรูป e ยกกำลัง
%x	รหัสควบคุม	เลขฐานสิบหก
%o	รหัสควบคุม	เลขฐานแปด
%f	รหัสควบคุม	เลขทศนิยม

1.10 ตัวแปร

ตัวแปร หมายถึง ชื่อหรือสัญลักษณ์ที่ใช้แทนข้อมูล โดยข้อมูลเหล่านี้จะถูกนำไปใช้ในการอ้างถึงในกระบวนการเขียนโปรแกรม

การตั้งชื่อตัวแปร ห้ามขึ้นต้นด้วยตัวเลขและเครื่องหมายดำเนินตลอดจนสัญลักษณ์พิเศษต่าง ๆ แต่สามารถนำตัวเลขต่อท้ายชื่อตัวแปรได้ และเครื่องหมาย _ (underscore) ถือว่าเป็นตัวอักษร สามารถนำใช้ตั้งชื่อตัวแปรได้

ภาษาไพธอนจะมีตัวแปรพื้นฐาน 6 ชนิดดังต่อไปนี้

- เลขจำนวนเต็ม (Integer) หมายถึง ข้อมูลที่อยู่ภายในตัวแปรจะเป็นเลขจำนวนเต็ม เช่น $a = 5$
- เลขทศนิยม (Floating) หมายถึง ข้อมูลที่อยู่ภายในตัวแปรจะเป็นเลขทศนิยม เช่น $b = 3.14$
- ข้อความ (String) หมายถึง ข้อมูลที่อยู่ภายในตัวแปรเป็นชนิดข้อความ เช่น $c = \text{"Google"}$
- ลิสต์ (List) หมายถึง ตัวแปรรายการ เช่น $d = [\text{"Google"}, \text{"Gmail"}, \text{"Youtube"}]$
- ทูเพิล (Tuple) จะกล่าวถึงในบทที่ 4 เช่น $e = (\text{"Google"}, \text{"Gmail"}, \text{"Youtube"})$
- ดิกชันนารี (Dictionary) จะกล่าวถึงในบทที่ 6 เช่น $f = \{\text{"one": "หนึ่ง"}, \text{"two": "สอง"}\}$

ตัวอย่างที่ 1.5 การประกาศตัวแปรชนิดต่าง ๆ

```
a = 5
b = 3.14
c = "Google"
d = ["Google", "Gmail", "Youtube"]
e = ("Google", "Gmail", "Youtube")
f = {"one": "หนึ่ง", "two": "สอง"}
```

ตัวอย่างที่ 1.5 เป็นการประกาศตัวแปร ชนิดต่าง ๆ ได้แก่ เลขจำนวนเต็ม ทศนิยม ข้อความ รายการลิสต์ ทูเพิล และดิกชันนารี โดยแสดงให้เห็นวิธีการประกาศตัวแปรแต่ละชนิด ซึ่งจะกล่าวถึงรายละเอียดในบทต่อไป

1.11 ชนิดข้อมูลและขนาดจำนวนไบต์

ภาษาไพธอนประกอบด้วยชนิดข้อมูลดังต่อไปนี้

ตารางที่ 1.2 ขนาดของตัวแปรประเภทต่าง ๆ

กลุ่มข้อมูล	ขนาด
Integer	12 bytes
Long integer	$12 \text{ bytes} + (\text{nbit}/16 + 1) * 2 \text{ bytes}$
Floats	16 bytes
Complex	24 bytes
List	$16 \text{ bytes} + 4 \text{ bytes}$ สำหรับแต่ละสมาชิก
Tuple	$20 \text{ bytes} + 4 \text{ bytes}$ สำหรับแต่ละสมาชิก
String	$20 \text{ bytes} + 1 \text{ bytes}$ ต่อตัวอักษร
Unicode string	$24 \text{ bytes} + 2 \text{ bytes}$ ต่อตัวอักษร
Dictionary	$24 \text{ bytes} + 12 * 2n \text{ bytes}$, $n = \log_2(\text{nItems}) + 1$
Class instance	16 bytes + dictionary object

1.12 การตรวจสอบตัวแปรและอ็อบเจ็กต์ต่าง ๆ ที่กำลังใช้งาน

การตรวจสอบตัวแปรและอ็อบเจ็กต์ต่าง ๆ ที่กำลังใช้งานในปัจจุบันสามารถใช้คำสั่ง `dir()` ผลลัพธ์ที่ได้จะเป็นชื่อรายการตัวแปรและอ็อบเจ็กต์ที่กำลังใช้งานอยู่ในปัจจุบัน นอกจากนั้น คำสั่ง `dir()` ยังสามารถใช้ตรวจสอบคำสั่งต่าง ๆ ที่อยู่ภายในอ็อบเจ็กต์หรือตัวแปรได้ เช่น

ตัวอย่างที่ 1.6 การแสดงค่าภายในตัวแปร

```
print(dir())
```

ผลลัพธ์ :

```
['__builtins__', '__doc__', 'a', 'b', 'c', 'd', 'e', 'f']
```

ตัวอย่างที่ 1.6 เป็นการเรียกใช้คำสั่ง `dir()` หลังจากทำตัวอย่างที่ 1.5 แล้ว ซึ่งจะพบว่าตัวแปร `a`, `b`, `c`, `d`, `e` และ `f` นอกจากนั้นยังมีตัวแปรและอ็อบเจ็กต์อื่น ๆ ที่กำลังถูกประกาศใช้งาน เช่น อ็อบเจ็กต์ `__builtins__` เก็บรวบรวมคำสั่งพื้นฐานต่าง ๆ เอาไว้ภายใน เป็นต้น

1.13 การลบตัวแปรและอ็อบเจ็กต์เพื่อคืนพื้นที่หน่วยความจำให้ระบบปฏิบัติการ (OS)

คำสั่ง `del` เป็นคำสั่งที่ใช้สำหรับลบตัวแปรหรืออ็อบเจ็กต์ออกจากหน่วยความจำ

ตัวอย่างที่ 1.7 การลบตัวแปร

```
del a,b,c,d,e,f
print(dir())
```

ผลลัพธ์ :

```
['__builtins__', '__doc__']
```

ในตัวอย่างที่ 1.7 ตัวแปร `a,b,c,d,e,f` เป็นผลจากตัวอย่างที่ 1.5 หลังจากนั้นจะทำการลบตัวแปรด้วยคำสั่ง `del a,b,c,d,e,f` เมื่อเรียกคำสั่ง `dir()` เพื่อตรวจสอบตัวแปรจะพบว่าตัวแปรได้ถูกลบและคืนหน่วยความจำให้กับระบบปฏิบัติการ

1.14 การตรวจสอบชนิดข้อมูลภายในตัวแปร

ในภาษาไพธอนการใช้ตัวแปรเป็นแบบไดนามิก คือ ตัวแปรสามารถเปลี่ยนแปลงชนิดได้อย่างอิสระขึ้นอยู่กับผู้เขียนโปรแกรม การตรวจสอบชนิดข้อมูลจะใช้ฟังก์ชัน `Type()` ในการตรวจสอบ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1.8 การตรวจสอบชนิดของตัวแปร

```
a = 3
c = 3.14
e = "Google"
print(type(a), type(c), type(e))
```

ผลลัพธ์ :

```
<class 'int'> <class 'float'> <class 'str'>
```

ตัวอย่างที่ 1.8 เป็นการประกาศตัวแปร a,c และ e เป็นชนิด เลขจำนวนเต็ม ทศนิยม และสตริง ตามลำดับ เมื่อตรวจสอบชนิดตัวแปร ด้วยคำสั่ง type(a), type(c) และ type(e) จะให้ผลลัพธ์เป็นชนิด int คือเลขจำนวนเต็ม , float คือเลขทศนิยม และ str คือข้อความสตริง

1.15 การแปลงตัวแปรเลขจำนวนเต็ม ทศนิยม และข้อความ

ตัวแปรแต่ละชนิดจะต้องแปลงให้อยู่ในรูปแบบที่ตรงกับการใช้งาน เช่น ถ้าเป็นตัวแปรสตริงจะไม่สามารถนำไปใช้คำนวณเชิงตัวเลข เช่น บวก ลบ คูณ หาร ดังนั้น จะต้องแปลงตัวแปรข้อความเป็นตัวเลขเสียก่อน คำสั่งประกอบด้วย str() , int() และ float() พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 1.9 การลบตัวแปร

```
c = "3.14"
d = float(c)
e = d + 1.2
print(d,e)
```

ผลลัพธ์ :

```
3.14 4.34
```

ตัวอย่างที่ 1.9 เป็นการประกาศตัวแปร c ให้มีค่าเป็นข้อความ "3.14" ซึ่งจะไม่สามารถนำไปใช้คำนวณได้ เนื่องจากเป็นข้อความ ดังนั้น จะต้องแปลงเป็นตัวแปรทศนิยมก่อน โดยใช้ฟังก์ชัน float() จึงจะสามารถนำไปใช้คำนวณเชิงตัวเลขได้ โดยให้ตัวแปร d เป็นผลลัพธ์ที่ได้จากการแปลงสตริงเป็นทศนิยม ซึ่งตัวแปร d จะเป็นตัวแปรชนิดทศนิยมสามารถนำไปใช้คำนวณบวก ลบ คูณ หารได้ โดยให้ตัวแปร e เป็นผลบวกของตัวแปร d กับ 1.2 ทำให้ค่า e มีค่า 4.34 ซึ่งเป็นตัวแปรทศนิยม สามารถตรวจสอบตัวแปรด้วยคำสั่ง type(d) และ type(e) จะพบว่าชนิดเป็นตัวแปรทศนิยม ส่วนตัวแปร c ตรวจสอบด้วยคำสั่ง type(c) จะพบว่าชนิดตัวแปรเป็นชนิดสตริงหรือข้อความ

1.16 การรับข้อมูลจากแป้นพิมพ์มาเก็บในตัวแปร

การรับข้อมูลจากแป้นพิมพ์ใน Python 2 มีคำสั่งรับค่าข้อมูลจากคีย์บอร์ด 2 คำสั่ง คือ raw_input() รับค่าสตริง และ input() รับค่าตัวเลข แต่ใน Python 3 เหลือเพียงคำสั่งเดียวคือ คำสั่ง input() รับค่าเป็นข้อความเพียงอย่างเดียว ดังนั้น เมื่อต้องการนำค่าที่รับเข้ามาจากคีย์บอร์ดไปคำนวณเชิงตัวเลขต้องทำการแปลงข้อมูลที่ป้อนเข้ามาซึ่งอยู่ในตัวแปรชนิดข้อความและแปลงเป็นชนิดตัวเลขก่อนนำไปใช้งาน ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1.10 การรับค่าตัวแปร

```
q = input("กรุณาป้อนเลขท้ายสองตัวของคุณ :")
```

```
print(type(q))
```

ผลลัพธ์ :

```
q = input("กรุณาป้อนเลขท้ายสองตัววงนี้ :")
print(type(q))
```

กรุณาป้อนเลขท้ายสองตัววงนี้ :

<class 'str'>

ตัวอย่างที่ 1.10 คำสั่ง input() จะรอรับการป้อนค่าจากผู้ใช้ และแสดงคำว่า "กรุณาป้อนเลขท้ายสองตัววงนี้ :" เมื่อป้อนค่าลงไปค่าที่ป้อนจะเก็บไว้ในตัวแปร q และมีชนิดเป็นตัวอักษร ตรวจสอบชนิดตัวแปรด้วยคำสั่ง type(q) เมื่อต้องการใช้งานคำนวณจะต้องแปลงชนิดตัวแปรให้เป็นตัวเลข ด้วยคำสั่ง float() และ int() ซึ่งจะแปลงเป็นชนิดทศนิยมและเลขจำนวนเต็มตามลำดับ

1.17 การเขียนหมายเหตุ (Comment)

การเขียนหมายเหตุ (Comment) เป็นการเขียนข้อความลงในซอร์สโค้ดของโปรแกรม นิยมเขียนเพื่ออธิบายความหมาย โดยที่หมายเหตุจะไม่ถูกนำไปประมวลผล การเขียนหมายเหตุด้วยภาษาไพธอน ใช้เครื่องหมาย # (sharp) ไว้ด้านหน้าข้อความที่ต้องการ เช่น

ตัวอย่างที่ 1.11 การลบตัวแปร

```
# นี่คือการเขียนหมายเหตุ เขียนไว้ในโปรแกรม
a = 39760 # ตัวแปร a ใช้เก็บเงินเดือน
print(a) # แสดงค่าเงินเดือนในตัวแปร a ออกมา
```

ผลลัพธ์ :

34000

ตัวอย่างที่ 1.11 เขียนหมายเหตุ (comment) 3 ตำแหน่ง โดยการเขียนหมายเหตุจะใส่เครื่องหมาย # ไว้บริเวณด้านหน้าข้อความที่ต้องการเขียนหมายเหตุ ปกติการเขียนคอมเมนต์เพื่อเตือนความจำหรือเขียนหมายเหตุทิ้งเอาไว้ในซอร์สโค้ด

แบบฝึกหัดท้ายบท

1. จากแผนภาพวิวัฒนาการภาษาโปรแกรมมิ่ง จงตอบว่าภาษาไพธอนพัฒนาขึ้น ในปี ค.ศ. ไດ
2. อธิบายการเขียนหมายเหตุด้วยภาษาไพธอน
3. อธิบายชนิดข้อมูล ตัวแปร และแสดงค่าภายในตัวแปร
4. อธิบายการตรวจสอบชนิดตัวแปร ลบตัวแปร แปลงตัวแปร
5. อธิบายการรับข้อมูลผ่านแป้นพิมพ์ใช้คำสั่งใดและรับค่าเข้ามาเป็นชนิดใด
6. เขียนโปรแกรมเพื่อคำนวณเส้นรอบวงโดยรับค่ารัศมีและคำนวณเส้นรอบวงวงกลม
7. เขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยมโดยรับค่าจากแป้นพิมพ์ความกว้างและความสูง

บทที่ 2

เครื่องหมายดำเนินการและนิพจน์

แนวคิด

เครื่องหมายทางคณิตศาสตร์ เช่น การบวก (+) การลบ (-) การคูณ (x) การหาร (÷) วงเล็บ()ปีกกา {} สามารถดำเนินการได้หลายแบบและอาจทำให้คำตอบไม่ตรงกัน เนื่องจากลำดับความสำคัญของตัวดำเนินการ ดังนั้น เพื่อการคำนวณนิพจน์ให้ได้คำตอบที่ถูกต้องตรงกันจำเป็นต้องเข้าใจระดับความสำคัญของตัวดำเนินการแต่ละตัวและความรู้พื้นฐานที่เกี่ยวข้อง

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ เครื่องหมายดำเนินการและนิพจน์ เครื่องหมายดำเนินการทางคณิตศาสตร์ เครื่องหมายดำเนินการทางตรรกศาสตร์ การแปลงเลขฐานสอง ลอจิกเกตพื้นฐาน การคำนวณของดิจิทัลคอมพิวเตอร์ ตัวเลขลบ และการเขียนโปรแกรมคำนวณสมการทางคณิตศาสตร์พื้นฐาน

วัตถุประสงค์

1. อธิบายเครื่องหมายดำเนินการและนิพจน์ได้
2. อธิบายเครื่องหมายดำเนินการทางคณิตศาสตร์ได้
3. อธิบายเครื่องหมายดำเนินการทางตรรกศาสตร์ได้
4. อธิบายการทำงานของวงจรบวกเลขได้
5. อธิบายวิธีการแปลงสมการคณิตศาสตร์เป็นนิพจน์เพื่อคำนวณได้

2.1 เครื่องหมายดำเนินการและนิพจน์

นิพจน์ (Expression) คือ การนำตัวแปรและตัวเลขมาเชื่อมด้วยเครื่องหมายทางคณิตศาสตร์ หรือเครื่องหมายการเปรียบเทียบ หรือเครื่องหมายตรรกศาสตร์ เช่น “ $3+a*4$ ”

หมายเหตุ: ** คือ เครื่องหมายยกกำลัง เช่น $a**4$ คือ a คูณกัน 4 ครั้ง ($a * a * a * a$)

2.2 เครื่องหมายทางคณิตศาสตร์ (Arithmetic Operators)

ตารางที่ 2.1 เครื่องหมายทางคณิตศาสตร์ (Arithmetic Operators)

เครื่องหมาย	ตัวอย่าง
การบวก (+)	$3 + 2$
การลบ (-)	$3 - 2$
การคูณ (*)	$3 * 2$
การหาร (/)	$3 / 2$
การยกกำลัง (**)	$2 ** 4$
การลดค่าตัวแปร (--)	$a=3; a-=1$ # $a-=1$ คือ $a = a-1$ ดังนั้น $a = 3-1$ ค่า a เท่ากับ 2
การเพิ่มค่าตัวแปร (++)	$a=3; a+=1$ # $a+=1$ คือ $a = a+1$ ดังนั้น $a = 3+1$ ค่า a เท่ากับ 4
การคูณค่าตัวแปร (*=)	$a=3; a*=3$ # $a*=3$ คือ $a = a*3$ ดังนั้น $a = 3*3$ ค่า a เท่ากับ 9
การหารค่าตัวแปร (/=)	$a=4; a/=2$ # $a/=2$ คือ $a = a/2$ ดังนั้น $a = 4/2$ ค่า a เท่ากับ 2
การหารเอาเศษ (%)	$a = 5; a\%2$ ผลลัพธ์ คือ 1

2.3 เครื่องหมายเปรียบเทียบ (Comparison operators)

ตารางที่ 2.2 เครื่องหมายเปรียบเทียบ (Comparison operators)

เครื่องหมาย	ตัวอย่าง
เท่ากับ (==)	3 == 3 ตอบ: จริง (True)
ไม่เท่ากับ (!=)	3 != 3 ตอบ: เท็จ (False)
น้อยกว่า (<)	3 < 2 ตอบ: เท็จ (False)
น้อยกว่าหรือเท่ากับ (<=)	3 <= 3 ตอบ: จริง (True)
มากกว่า (>)	3 > 2 ตอบ: จริง (True)
มากกว่าหรือเท่ากับ (>=)	3 >= 3 ตอบ: จริง (True)

2.4 เครื่องหมายตรรกะ (Logical operators)

เครื่องหมายทางตรรกศาสตร์ เป็นการเปรียบเทียบค่าความเป็นจริง หรือเท็จ ระหว่างสิ่งต่าง ๆ สองสิ่งว่าเป็นจริงหรือเท็จ

ตารางที่ 2.3 เครื่องหมายเปรียบเทียบ (Comparison operators)

เครื่องหมาย	ตัวอย่าง
and (&)	(4 < 3) & (1 < 2) เท่ากับ 0 and 1 ตอบ: เท็จ(False)
or ()	(4 < 3) (1 < 2) เท่ากับ 0 or 1 ตอบ: จริง (True)
not (!)	not 0 ตอบ: จริง (True) และ not 1 ตอบ: เท็จ (False)

2.5 ลำดับและความสำคัญของตัวดำเนินการ

ลำดับความสำคัญของตัวดำเนินการ แบ่งเป็น 4 ระดับ ดังนี้

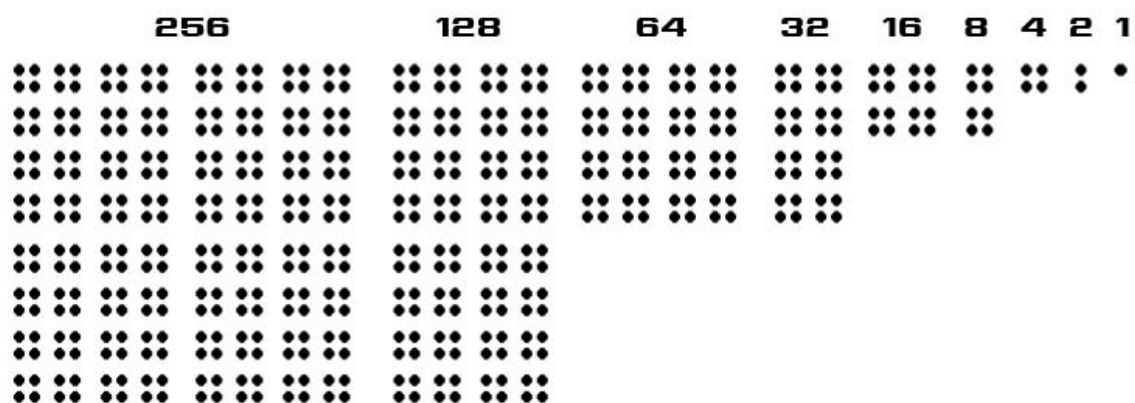
- สำคัญอันดับ 1 คือ เครื่องหมาย วงเล็บ ()
- สำคัญอันดับ 2 คือ เครื่องหมายยูนิอัส เช่น x = 10 ; -x
- สำคัญอันดับ 3 คือ เครื่องหมาย คูณ (*)หาร (/) และการหารเอาเศษ (modulo) (%)
- สำคัญอันดับ 4 คือ เครื่องหมาย บวก (+) ลบ (-)

ตารางที่ 2.4 ลำดับความสำคัญของตัวดำเนินการ

เครื่องหมาย	ลำดับความสำคัญ
(5 - 2) * 3	ก) ทำในวงเล็บก่อน (5-2) = 3 ข) นำผลลัพธ์ไปบวก 3 ได้เท่ากับ 6
5 + 2 * 2	ก) นำ 2 * 2 ข) นำผลลัพธ์ไปบวกกับ 5
(2 + 5) * 4 % 20	ก) ทำในวงเล็บก่อน (2+5) = 7 ข) ทำคูณ 7x4 = 28 ค) ทำการหารเอาเศษด้วย 20 ได้ 8
2 + 5 * 4 % 20	ก) ทำคูณก่อน จะได้ 5 * 4 = 20 ข) ทำหารเอาเศษ จะได้ 0 ค) นำ 0 บวก 2 จะได้ 2

2.6 การแปลงเลขฐานสิบและฐานสอง

ระบบคอมพิวเตอร์แทนข้อมูลด้วยรหัสฐานสองอันเนื่องจากการแปลงข้อมูลให้อยู่ในรูปของสัญญาณทางไฟฟ้าที่มีสองสถานะ คือ 1) มีไฟ (on) 2) ไม่มีไฟ (off)



ภาพประกอบที่ 2.1 แสดงค่าประจำหลักของระบบเลขฐานสอง

พิจารณาดังต่อไปนี้ เลข 7 มีหลักใดบ้างที่รวมกันแล้วได้ 7 ตอบ 4 2 1 ดังนั้นเลข 7 เขียนเป็นฐานสองได้เท่ากับ 0111

ตารางที่ 2.5 การแปลงเลขฐานสิบเป็นฐานสอง

8 ⬢ ⬢	4 ⬢ ⬢	2 ⬢ ⬢	1 ⬢ ⬢	คำอธิบาย
0	0	0	0	เลข 0 ตรงกับฐานสอง คือ 0 0 0 0
0	0	0	1	เลข 1 ตรงกับฐานสอง คือ 0 0 0 1
0	0	1	0	เลข 2 ตรงกับฐานสอง คือ 0 0 1 0
0	0	1	1	เลข 3 ตรงกับฐานสอง คือ 0 0 1 1
0	1	0	0	เลข 4 ตรงกับฐานสอง คือ 0 1 0 0
0	1	0	1	เลข 5 ตรงกับฐานสอง คือ 0 1 0 1
0	1	1	0	เลข 6 ตรงกับฐานสอง คือ 0 1 1 0
0	1	1	1	เลข 7 ตรงกับฐานสอง คือ 0 1 1 1
1	0	0	0	เลข 8 ตรงกับฐานสอง คือ 1 0 0 0
1	0	0	1	เลข 9 ตรงกับฐานสอง คือ 1 0 0 1
1	0	1	0	เลข 10 ตรงกับฐานสอง คือ 1 0 1 0
1	0	1	1	เลข 11 ตรงกับฐานสอง คือ 1 0 1 1
1	1	0	0	เลข 12 ตรงกับฐานสอง คือ 1 1 0 0
1	1	0	1	เลข 13 ตรงกับฐานสอง คือ 1 1 0 1
1	1	1	0	เลข 14 ตรงกับฐานสอง คือ 1 1 1 0

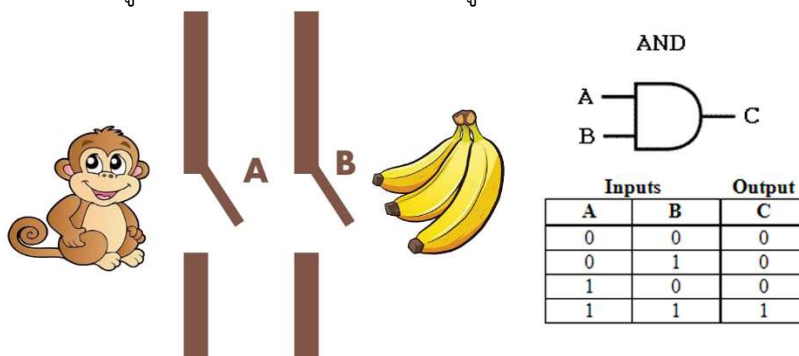
8 ⋮ ⋮ ⋮	4 ⋮ ⋮	2 ⋮	1 •	คำอธิบาย
1	1	1	1	เลข 15 ตรงกับฐานสอง คือ 1 1 1 1

2.7 ลอจิกเกต (Logic gate)

ในโลกของระบบดิจิทัลมีลอจิกเกต 3 อย่างที่เป็นพื้นฐานสำคัญ คือ ประตูแบบ And Or และ Not เมื่อนำลอจิกเกตมาต่อเรียงกันไปอย่างมีระเบียบแบบแผน จะสามารถสร้างวงจรการบวกเลข และสามารถนำวงจรการบวกนั้นมาสร้างเป็นการลบ การคูณ และหาร ซึ่งจะกล่าวถึงในลำดับต่อไป

2.8 And Gate

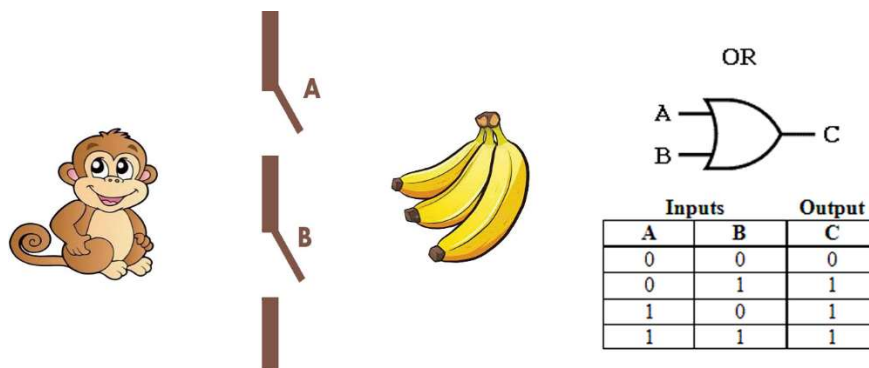
ถ้ามีประตูสองบานตั้งขนานกันดังภาพจะพบว่าลิงเข้าไปเอากล้วยได้ต่อเมื่อประตูทั้งสองบานต้องเปิด ถ้ามีบานใดบานหนึ่งปิดจะไม่สามารถไปเอากล้วยมากินได้ ปรากฏการณ์นี้เป็นแนวคิดของ and gate เมื่อแทน 0 คือการปิดประตู และแทน 1 ด้วยการเปิดประตู



ภาพประกอบที่ 2.2 แนวคิดของ And Gate

2.9 OR Gate

ถ้าในห้องมีประตูสองบานดังภาพด้านล่าง พบว่าลิงจะไปเอากล้วยมากินได้เมื่อประตูบานใดบานหนึ่งเปิด ปรากฏการณ์นี้เป็นแนวคิดของ OR gate เมื่อแทน 0 คือการปิดประตู และแทน 1 ด้วยการเปิดประตู



ภาพประกอบที่ 2.3 แนวคิดของ OR Gate

2.10 Not Gate

น็อตเกต คือ การสลับค่าของสัญญาณขาเข้า เช่น ถ้าสัญญาณขาเข้าเป็น 0 สัญญาณขาออกจะเป็น 1 หรือถ้าสัญญาณขาเข้าเป็น 1 สัญญาณขาออกเป็น 0 ในธรรมชาติถ้าตั้งทางแมวมาด้านหลังมันจะพยายามดันตัวเองไปข้างหน้า ถ้าพยายามผลักกันแมวไปด้านหน้าแมวจะพยายามถอยหลัง เป็นปรากฏการณ์ที่นำมาเทียบกับการทำงานของน็อตเกต สัญลักษณ์น็อตเกตเป็นรูปลูกศรชี้ไปทางขวาและมีวงกลมที่ปลายลูกศร

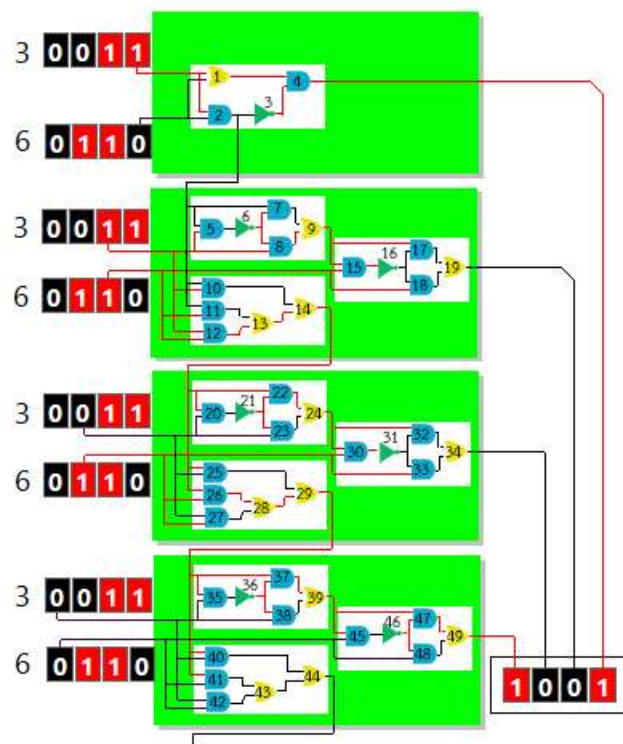


ภาพประกอบที่ 2.4 แนวคิดของ Not Gate

2.11 วงจรบวกเลข

วงจรลอจิกที่ทำหน้าที่คำนวณในทางคณิตศาสตร์ ได้แก่ วงจรบวกเลข (Adder Logic) สร้างขึ้นจากลอจิกเกต พิจารณาการคำนวณของวงจรบวกเลขต่อไปนี้ ในตัวอย่างเป็นการบวกเลข $3 + 6$ ในขั้นแรกจะทำการแปลงเลข 3 เป็นฐานสอง มีค่า 0011 และแปลงเลข 6 เป็นฐานสองมีค่า 0110 หลังจากนั้นป้อนเข้าไปยังวงจรบวกเลขโดยบล็อกบนสุดเป็น Half Adder คือการบวกแบบไม่คิดตัวทด ส่วนบล็อกที่ 2 และ 4 เป็น Full Adder คือบวกแบบมีบิตตัวทด

ในการประมวลผลจะเริ่มตั้งแต่บล็อกบนสุด เรียกว่า Half Adder คือการบวกแบบไม่มีตัวทด เริ่มต้นที่ Or gate (1) โดยมีอินพุตมีค่าเป็น 1 และ 0 ให้ผลลัพธ์ 1 ลำดับต่อมาเป็น And gate (2) มีอินพุตเป็น 1 และ 0 ให้ผลลัพธ์ 0 ลำดับถัดไปคือ not gate (3) มีอินพุตเป็น 0 ให้ผลลัพธ์ 1 และในลำดับสุดท้ายของบล็อกบนสุด คือ and gate(4) มีอินพุต คือ 1 และ 1 ให้ผลลัพธ์เป็น 1 การดำเนินการจะทำซ้ำไปจนกระทั่งถึงลำดับสุดท้าย คือ or gate (49) ผลลัพธ์ที่ได้จะมีค่าเท่ากับ 1001 ซึ่งอยู่ในเลขฐานสอง ดังนั้นต้องแปลงกลับมาเป็นเลขฐาน 10 ซึ่งมีค่าเท่ากับ 9 ซึ่งเป็นความจริง ที่ $3+6$ เท่ากับ 9 ตอนนี้สามารถสร้างวงจรการบวกเลข ในขั้นต่อไปเป็นการลบ คูณ และหาร ซึ่งสามารถใช้เพียงวงจรบวกอย่างเดียวทำการ ลบ คูณ หาร ได้ โดยการคูณ คือ การบวกซ้ำ ๆ กัน เช่น 3×4 คือ การนำ 3 บวกกัน 4 ครั้ง คือ $3+3+3+3$ หรือ 4 บวกกัน 3 ครั้ง คือ $4+4+4$



ภาพประกอบที่ 2.5 วงจรบวกเลข (Adder Logic)

ที่มา: <http://dsdi.msu.ac.th/programming/Adder.zip>

2.12 เลขลบ

การวิจัยเรื่องตัวเลขลบ มีวิวัฒนาการมา 3 รอบ คือ 1) Sign Magnetude 2) One's complement 3) Two's complement และในปัจจุบันนี้ระบบคอมพิวเตอร์ใช้เลขลบด้วยวิธี Two's complement พิจารณาภาพต่อไปนี้

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Unsigned	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Sign & Magnitude	0	1	2	3	4	5	6	7	-0	-1	-2	-3	-4	-5	-6	-7
1's Compl.	0	1	2	3	4	5	6	7	-7	-6	-5	-4	-3	-2	-1	-0
2's Compl.	0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1

ภาพประกอบที่ 2.6 แสดงการเปรียบเทียบเลขลบด้วยวิธีการแบบต่าง ๆ

- Unsigned คือ ตัวเลขที่ไม่มีบิตเครื่องหมาย ในตัวอย่างนี้เก็บตัวเลขขนาด 4 บิต ดังนั้น ถ้าเป็น unsigned จะสามารถเก็บค่าตัวเลขได้ 0 - 15

- Sign Magnitude เป็นวิธีทำเลขลบด้วยการนำบิตซ้ายสุดเป็นเครื่องหมาย (1 คือลบ 0 คือบวก) ถ้าต้องการแทนเลข +1 จะมีค่า 0001 และหากต้องการแทนเลข -1 จะแทนด้วย 1001
- One's complement เป็นการแทนเลขลบด้วยการสลับบิต เช่น เลข 1 มีค่า 0001 เมื่อทำเลขลบด้วยวิธีนี้จะสลับจาก 0 เป็น 1 และจาก 1 เป็น 0 ดังนั้น เลข -1 ของวิธีนี้มีค่า 1110

จากภาพด้านบนจะพบว่าวิธีการทำเลขลบด้วยวิธี Sign magnitude และ one's complement มีปัญหาที่ เลข 0 มีทั้ง +0 และ -0 ซึ่งในความเป็นจริงคือตัวเลขเดียวกัน คือ 0 จึงได้มีการคิดวิธีการสร้างเลขลบแบบใหม่ เพื่อแก้ปัญหาการมีเลข +0 และ -0 เรียกวิธีนี้ว่า two's complement วิธีการคือหลังจากการทำ one's complement แล้วให้บวกเพิ่มด้วย 1

จากหัวข้อวงจรบวกเลขที่ผ่านมา ถ้าต้องการลบเลข เช่น $5 - 3$ โดยใช้วงจรบวกเลข สามารถทำได้โดย บวกอินพุต คือ $5 + (-3)$ กล่าวคือ ทำการแปลงเลข 3 เป็น -3 จากนั้นส่งผ่านเข้าไปในวงจรบวกเลข จะสามารถคำนวณการลบเลขโดยใช้วงจรบวกเลขตัวเดิมได้

2.13 การเลื่อนบิตไปทางซ้ายและขวา

ในภาษาไพธอนใช้เครื่องหมาย << แทนการเลื่อนบิตไปทางซ้าย และใช้เครื่องหมาย >> แทนการเลื่อนบิตไปทางขวา

ตัวอย่างที่ 2.1 การเลื่อนบิตไปทางซ้ายและทางขวา

```
x = 4
print(x << 1)
print(x >> 2)
```

ผลลัพธ์ :

```
8
1
```

ตัวอย่างที่ 2.1 เลข 4 ในฐาน 2 คือ 0100 เมื่อเลื่อนไปทางซ้าย 1 ตำแหน่งจะมีค่า 1000 มีค่าเท่ากับ 8 ในขณะที่เลข 4 นำมาเลื่อนบิตไปทางขวา 2 ตำแหน่ง จะมีค่า 0100 เป็น 0001 ซึ่งมีค่าเท่ากับ 1

2.14 การเขียนโปรแกรมคำนวณเส้นรอบวงกลม

ในหัวข้อนี้เป็นการคำนวณ เส้นรอบวงกลม จากสูตร $R = 2\pi r$ เมื่อ R คือความยาวเส้นรอบวงกลม และ r คือ รัศมี

ตัวอย่างที่ 2.2 การคำนวณเส้นรอบวงกลม

```
from math import pi
def main():
```

```

r = input("ป้อนรัศมี : ")
r = float(r)
R = 2 * pi * r
print("เส้นรอบวง ที่มีรัศมี %f มีค่าเท่ากับ %f หน่วย"%(r,R))

main()

```

ผลลัพธ์ :

```

from math import pi
def main():
    r = input("ป้อนรัศมี : ")
    r = float(r)
    R = 2 * pi * r
    print("เส้นรอบวง ที่มีรัศมี %f มีค่าเท่ากับ %f หน่วย"%(r,R))

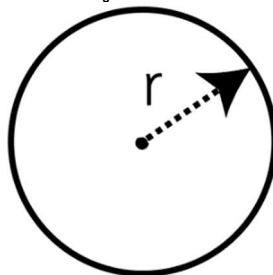
main()
... ป้อนรัศมี : 25
เส้นรอบวง ที่มีรัศมี 25.000000 มีค่าเท่ากับ 157.079633 หน่วย

```

ตัวอย่างที่ 2.2 คำสั่ง `from math import pi` เป็นการขอใช้ค่า `pi` จากโมดูล `math` คำสั่ง `def main()` เป็นการสร้างฟังก์ชัน `main()` ภายในประกอบด้วยคำสั่ง `input()` เพื่อรับค่ารัศมีจากผู้ใช้ จากนั้นนำมาเก็บไว้ในตัวแปร `r` หลังจากนั้นแปลงให้เป็นทศนิยมด้วยคำสั่ง `float(r)` และนำมาคำนวณด้วยนิพจน์ `R = 2 * pi * r` เมื่อคำนวณแล้วเก็บค่าในตัวแปร `R` และพิมพ์ออกมาบนจอภาพด้วยคำสั่ง `print()` เมื่อกำหนดให้ `%f` เป็นรหัสควบคุมรูปแบบการแสดงผลเพื่อแสดงค่าตัวเลขทศนิยม ผลลัพธ์ที่ได้คือ “เส้นรอบวง ที่มีรัศมี 25.000000 มีค่าเท่ากับ 157.079633 หน่วย”

2.15 การเขียนโปรแกรมคำนวณพื้นที่วงกลม

การคำนวณหาพื้นที่วงกลม คำนวณจากสูตร $area = \pi \times r^2$



ตัวอย่างที่ 2.3 การเขียนโปรแกรมคำนวณพื้นที่วงกลม

```

from math import pi
r = float(input("ป้อนรัศมี: "))

```

```
area = pi * r ** 2
print("พื้นที่วงกลมที่มีความยาวรัศมี %.2f เท่ากับ %.2f ตารางหน่วย"%(r,area))
```

ผลลัพธ์ :

```
ป้อนรัศมี: 1.5
พื้นที่วงกลมที่มีความยาวรัศมี 1.50 เท่ากับ 7.07 ตารางหน่วย
```

ตัวอย่างที่ 2.3 การคำนวณพื้นที่วงกลมเมื่อรู้ความยาวรัศมี คำนวณได้จากสูตร $area = \pi \times r^2$ ในขั้นแรกให้รับค่าจากแป้นพิมพ์ด้วยคำสั่ง `input("ป้อนรัศมี: ")` โปรแกรมจะแสดงคำว่า "ป้อนรัศมี: " จากนั้นผู้ใช้จะป้อนค่าเข้ามา ซึ่งตอนนี้จะมีค่าเป็นข้อความ จากนั้นให้นำข้อความมาแปลงเป็นทศนิยมด้วยคำสั่ง `float()` และกำหนดค่าเข้าให้ตัวแปร `r` ต่อมาให้คำนวณพื้นที่ ด้วยสูตร `pi * r ** 2` เมื่อ `**` คือการยกกำลังสอง และค่า `pi` เป็นค่าคงที่ นำมาใช้งานด้วยคำสั่ง `from math import pi` หลังจากหาคำนวณพื้นที่วงกลมได้แล้วให้นำมาแสดงผล ด้วยคำสั่ง `print` และใช้อักขระควบคุมการแสดงผลด้วย `%.2f` คือแสดงทศนิยมสองตำแหน่ง ในตัวอย่างนี้ป้อนรัศมีเท่ากับ 1.5 ผลลัพธ์ คือ "พื้นที่วงกลมที่มีความยาวรัศมี 1.50 เท่ากับ 7.07 ตารางหน่วย"

2.16 การเขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยม

การคำนวณหาพื้นที่สี่เหลี่ยมที่รู้ค่าความยาวฐานและความสูง คำนวณจากสูตร $area = a \times b$



ตัวอย่างที่ 2.4 การเขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยม

```
a = float(input("ป้อนระยะ a: "))
b = float(input("ป้อนระยะ b: "))
area = a*b
print("พื้นที่สี่เหลี่ยมที่มีความยาวฐาน %.2f และสูง %.2f เท่ากับ %.2f ตารางหน่วย"%(a,b,area))
```

ผลลัพธ์ :

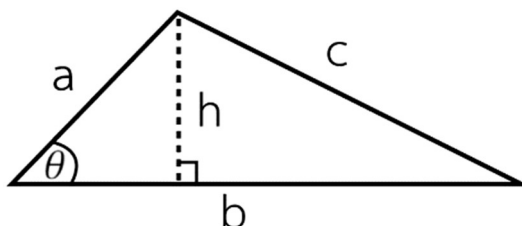
```
ป้อนระยะ a: 3.4
ป้อนระยะ b: 12.8
พื้นที่สี่เหลี่ยมที่มีความยาวฐาน 3.40 และสูง 12.80 เท่ากับ 43.52 ตารางหน่วย
```

ตัวอย่างที่ 2.4 การคำนวณพื้นที่สี่เหลี่ยมที่รู้ความกว้างและความสูง คำนวณจากสูตรความยาวฐานคูณความสูง ในขั้นแรกให้รับค่าจากผู้ใช้มาเก็บไว้ในตัวแปร `a` และ `b` ด้วยคำสั่ง `input()` ให้เขียนข้อความบอกผู้ใช้ภายในคำสั่ง `input` เช่น "ป้อนระยะ : " จากนั้นนำค่าที่ได้มาแปลงเป็นทศนิยมด้วย

ฟังก์ชัน `float()` และเก็บไว้ในตัวแปร `a` และ `b` ตามลำดับ ในขั้นต่อมาคำนวณพื้นที่ด้วยคำสั่ง `area = a*b` จากนั้นพิมพ์ผลลัพธ์โดยใช้รหัสควบคุมการแสดงผล `%.2f` ซึ่งเป็นการแสดงทศนิยมสองตำแหน่ง จากตัวอย่างนี้ป้อนค่า `a = 3.4` และค่า `b = 12.8` ผลลัพธ์ คือ "พื้นที่สี่เหลี่ยมที่มีความยาวฐาน 3.40 และสูง 12.80 เท่ากับ 43.52 ตารางหน่วย"

2.17 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้ความยาวด้านทั้งสาม

การคำนวณหาพื้นที่สามเหลี่ยมที่รู้ค่าความยาวด้านทั้งสาม คำนวณจากสูตร



$$s = \frac{1}{2}(a + b + c)$$

$$area = \sqrt{s(s-a) \times (s-b) \times (s-c)}$$

ตัวอย่างที่ 2.5 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้ความยาวของด้านทั้งสาม

```
from math import sqrt
a = float(input("ป้อนระยะ a: "))
b = float(input("ป้อนระยะ b: "))
c = float(input("ป้อนระยะ c: "))
s = 1/2 * (a+b+c)
area = sqrt(s*(s-a)*(s-b)*(s-c))
print("พื้นที่สามเหลี่ยมที่รู้ความยาวทั้งสามด้านมีค่าเท่ากับ %.2f ตารางหน่วย"%(area))
```

ผลลัพธ์ :

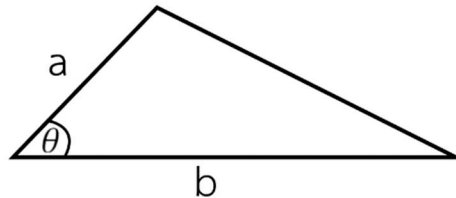
```
ป้อนระยะ a: 17
ป้อนระยะ b: 35
ป้อนระยะ c: 25
พื้นที่สามเหลี่ยมที่รู้ความยาวทั้งสามด้านมีค่าเท่ากับ 197.77 ตารางหน่วย
```

ตัวอย่างที่ 2.5 การคำนวณพื้นที่สามเหลี่ยมที่รู้ความยาวของด้านทั้งสาม ในขั้นแรกเขียนคำสั่งรับค่าความยาวของด้านทั้งสามโดยด้วยคำสั่ง `input()` และแสดงคำว่า "ป้อนระยะ :" ค่าที่รับเข้ามาแล้วมีชนิดเป็นสตริง ให้แปลงเป็นตัวเลขทศนิยมด้วยคำสั่ง `float()` จากนั้นนำไปคำนวณหาตัวแปร `s = (a+b+c)/2` และนำค่า `s` ที่ได้ไปคำนวณหาพื้นที่ด้วยสูตร `area = sqrt(s*(s-a)*(s-b)*(s-c))` แล้วทำการถอดรากที่สองด้วยคำสั่ง `sqrt` ซึ่งอยู่ภายในโมดูล `math` นำเข้ามาใช้งานด้วยคำสั่ง `from math import sqrt` จากนั้นนำค่าที่ได้มาแสดงผลด้วยคำสั่ง `print()` โดยใช้อักขระควบคุมการแสดงผลด้วย `%.2f` เพื่อแสดงทศนิยมสอง

ตำแหน่ง ผลลัพธ์มีค่า "พื้นที่สามเหลี่ยมที่รู้ความยาวทั้งสามด้านมีค่าเท่ากับ 197.77 ตารางหน่วย" เมื่อป้อนค่า a =17 b=35 และ c=25 คำนวณพื้นที่สามเหลี่ยมที่รู้ค่าทั้งสามด้านมีค่า 197.77 ตารางหน่วย

2.18 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้มุมและความยาวสองด้าน

การคำนวณหาพื้นที่สามเหลี่ยมเมื่อรู้มุมและความยาวของด้านเพียง 2 ด้าน คำนวณจากสูตร



$$\text{area} = \frac{1}{2} \times b \times a \times \sin(\theta)$$

ตัวอย่างที่ 2.6 การเขียนโปรแกรมคำนวณพื้นที่สามเหลี่ยมเมื่อรู้มุมและความยาวสองด้าน

```
from math import sin, pi
a = float(input("ป้อนระยะ a: "))
b = float(input("ป้อนระยะ b: "))
theta = float(input("ป้อนมุม theta: "))
area = 1/2 * b * a * sin(theta* pi/180)
print("พื้นที่สามเหลี่ยมที่รู้มุมและความยาวสองด้านมีค่าเท่ากับ %.2f ตารางหน่วย"%(area))
```

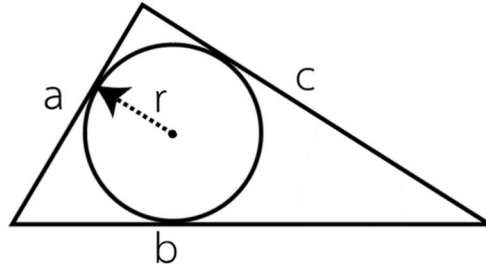
ผลลัพธ์ :

```
ป้อนระยะ a: 0.75
ป้อนระยะ b: 2
ป้อนมุม theta: 35
พื้นที่สามเหลี่ยมที่รู้มุมและความยาวสองด้านมีค่าเท่ากับ 0.43 ตารางหน่วย
```

ตัวอย่างที่ 2.6 การหาพื้นที่สามเหลี่ยมเมื่อรู้มุมและความยาวด้านสองด้าน ให้รับค่าตัวแปร a และ b เข้ามาซึ่งเป็นความยาวของด้าน โดยค่าที่รับเข้ามาแปลงเป็นทศนิยม จากนั้นรับค่ามุมเข้ามาและแปลงเป็นทศนิยม และคำนวณพื้นที่จากสูตร $\text{area} = 1 / 2 * b * a * \sin(\text{theta})$ ที่มุม theta แปลงเป็นองศาด้วยการคูณด้วยค่า pi / 180 ในตัวอย่างนี้มีการขอใช้ฟังก์ชัน sin และค่าคงที่ pi จากโมดูล math สามารถอิมพอร์ตด้วยคำสั่ง `from math import sin, pi` หลังจากคำนวณพื้นที่จากนั้นนำมาแสดงผลด้วยอักขระควบคุมการแสดงผล `%.2f` คือแสดงทศนิยมสองตำแหน่ง ผลลัพธ์มีค่า "พื้นที่สามเหลี่ยมที่รู้มุมและความยาวสองด้านมีค่าเท่ากับ 0.43 ตารางหน่วย"

2.19 การเขียนโปรแกรมคำนวณรัศมีของวงกลมภายในสามเหลี่ยม

การคำนวณหาความยาวรัศมีของวงกลมที่อยู่ภายในสามเหลี่ยมที่กำหนดความยาวของด้านทั้ง 3 ของสามเหลี่ยม คือ



$$s = \frac{1}{2}(a + b + c)$$

$$r = \frac{\sqrt{s(s-a) \times (s-b) \times (s-c)}}{s}$$

ตัวอย่างที่ 2.7 การเขียนโปรแกรมคำนวณรัศมีของวงกลมภายในสามเหลี่ยม

```
from math import sqrt
a = float(input("ป้อนระยะ a: "))
b = float(input("ป้อนระยะ b: "))
c = float(input("ป้อนระยะ c: "))
s = 1/2 * (a+b+c)
r = sqrt(s*(s-a)*(s-b)*(s-c)) / s
print("รัศมีของวงกลมภายในสามเหลี่ยมมุมฉากมีค่าเท่ากับ %.2f หน่วย"%(r))
```

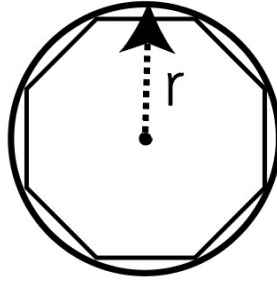
ผลลัพธ์ :

```
ป้อนระยะ a: 2
ป้อนระยะ b: 2.75
ป้อนระยะ c: 2.6
รัศมีของวงกลมภายในสามเหลี่ยมมุมฉากมีค่าเท่ากับ 0.67 หน่วย
```

ตัวอย่างที่ 2.7 รับค่าตัวแปร a,b,c และแปลงเป็นทศนิยม นำมาคำนวณเพื่อหาค่า s และ r จากนั้นแสดงผลลัพธ์ มีค่า รัศมีของวงกลมภายในสามเหลี่ยมมุมฉากมีค่าเท่ากับ 0.67 หน่วย เมื่อป้อน a=2, b=2.75 และ c=2.6

2.20 การเขียนโปรแกรมคำนวณพื้นที่รูปหลายเหลี่ยมด้านเท่า

การคำนวณหาพื้นที่รูปหลายเหลี่ยมด้านเท่าจากสูตร $area = \frac{1}{2} \times n \times r \times \sin(\frac{2\pi}{n})$



ตัวอย่างที่ 2.8 การเขียนโปรแกรมคำนวณพื้นที่รูปหลายเหลี่ยมด้านเท่า

```
from math import sin
r = float(input("ป้อนรัศมี: "))
n = int(input("ป้อนจำนวนด้าน: "))
area = 1/2*n*(r**2) * sin( ((2*pi) /n))
print("พื้นที่รูปหลายเหลี่ยมที่มีรัศมี %.2f หน่วยและจำนวนด้าน %d ด้าน เท่ากับ %.2f ตารางหน่วย"%(r,n,area))
```

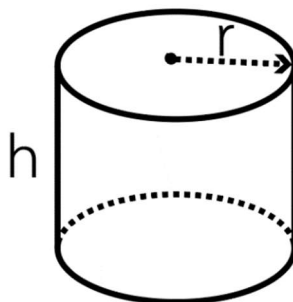
ผลลัพธ์ :

```
ป้อนรัศมี: 9
ป้อนจำนวนด้าน: 8
พื้นที่รูปหลายเหลี่ยมที่มีรัศมี 9.00 หน่วยและจำนวนด้าน 8 ด้าน เท่ากับ 229.10 ตารางหน่วย
```

ตัวอย่างที่ 2.8 การคำนวณพื้นที่รูปหลายเหลี่ยมด้านเท่าเมื่อทราบรัศมีและจำนวนด้านคำนวณจากสูตร $area = \frac{1}{2} \times n \times r \times \sin(\frac{2\pi}{n})$ เมื่อ n คือจำนวนด้าน และ r คือรัศมี ในขั้นแรกรับค่ารัศมีจากผู้ใช้ด้วยคำสั่ง `input()` และแปลงเป็นเลขทศนิยมด้วยคำสั่ง `float()` และกำหนดค่าให้ตัวแปร `r` ต่อมารับค่าจำนวนด้าน ซึ่งแปลงเป็นเลขจำนวนเต็มด้วยคำสั่ง `int()` เพราะจำนวนด้านต้องเป็นเลขจำนวนเต็มเท่านั้น หลังจากนั้นคำนวณพื้นที่ ด้วยนิพจน์ $area = 1/2*n*(r**2) * \sin(((2*pi) /n))$ และแสดงผลลัพธ์ด้วยคำสั่ง `print` ด้วยอักขระควบคุมการแสดงผล `%.2f` คือแสดงผลลัพธ์เป็นเลขทศนิยมสองตำแหน่ง และ `%d` แสดงเลขจำนวนเต็ม ในตัวอย่างนี้ป้อนค่า $r = 9$ และ $n = 8$ ผลลัพธ์มีค่า "พื้นที่รูปหลายเหลี่ยมที่มีรัศมี 9.00 หน่วยและจำนวนด้าน 8 ด้าน เท่ากับ 229.10 ตารางหน่วย"

2.21 การเขียนโปรแกรมคำนวณปริมาตรทรงกระบอกโดยรับค่ารัศมีและความสูง

การคำนวณหาปริมาตรทรงกระบอกจากสูตร $\text{volume} = \pi \times r^2 \times h$



ตัวอย่างที่ 2.9 การเขียนโปรแกรมคำนวณปริมาตรทรงกระบอกโดยรับค่ารัศมีและความสูง

```
from math import pi
r = float(input("ป้อนรัศมี: "))
h = float(input("ป้อนความสูง: "))
volume = pi * r ** 2 * h
print("ปริมาตรทรงกระบอกที่มีรัศมี %.2f หน่วยและสูง %d หน่วย เท่ากับ %.2f ลูกบาศก์หน่วย"%(r,h,volume))
```

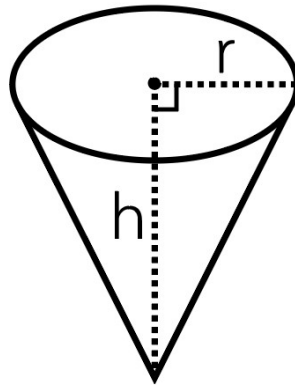
ผลลัพธ์ :

```
ป้อนรัศมี: 25.6
ป้อนความสูง: 70.5
ปริมาตรทรงกระบอกที่มีรัศมี 25.60 หน่วยและสูง 70 หน่วย เท่ากับ 145150.63 ลูกบาศก์หน่วย
```

ตัวอย่างที่ 2.9 การคำนวณปริมาตรทรงกระบอกเมื่อทราบรัศมีและความสูง คำนวณจากสูตรปริมาตรเท่ากับ $\pi \times r^2 \times h$ เมื่อ r คือรัศมีและ h คือความสูง ในขั้นแรกรับค่ารัศมีและความสูงจากผู้ใช้ด้วยคำสั่ง `input()` และแปลงเป็นเลขทศนิยมด้วยคำสั่ง `float()` และกำหนดค่าให้ตัวแปร r และ h ตามลำดับ จากนั้นคำนวณปริมาตรด้วยนิพจน์ `volume = pi * r ** 2 * h` และแสดงผลด้วยคำสั่ง `print` ด้วยอักขระควบคุมการแสดงผล `%.2f` คือแสดงผลลัพธ์เป็นเลขทศนิยมสองตำแหน่ง ในตัวอย่างนี้ป้อนค่า $r = 25.6$ และ $h = 70.5$ ผลลัพธ์มีค่า "ปริมาตรทรงกระบอกที่มีรัศมี 25.60 หน่วยและสูง 70 หน่วย เท่ากับ 145150.63 ลูกบาศก์หน่วย"

2.22 การเขียนโปรแกรมคำนวณปริมาตรกรวยโดยรับค่ารัศมีและความสูง

การคำนวณหาปริมาตรทรงกระบอกจากสูตร $\text{volume} = \frac{1}{3} \pi \times r^2 \times h$



ตัวอย่างที่ 2.10 การเขียนโปรแกรมคำนวณปริมาตรกรวยโดยรับค่ารัศมีและความสูง

```
from math import pi
r = float(input("ป้อนรัศมี: "))
h = float(input("ป้อนความสูง: "))
volume = 1/3 * (pi * r ** 2 * h)
print("ปริมาตรกรวยที่มีรัศมี %.2f หน่วยและสูง%d หน่วย เท่ากับ %.2f ลูกบาศก์หน่วย"%(r,h,volume))
```

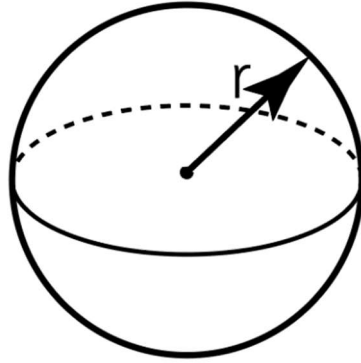
ผลลัพธ์ :

```
ป้อนรัศมี: 4.97
ป้อนความสูง: 12.5
ปริมาตรกรวยที่มีรัศมี 4.97 หน่วยและสูง12 หน่วย เท่ากับ 323.33 ลูกบาศก์หน่วย
```

ตัวอย่างที่ 2.10 การคำนวณปริมาตรกรวยเมื่อทราบรัศมีและความสูง คำนวณจากสูตรปริมาตรเท่ากับ $\frac{1}{3} \pi \times r^2 \times h$ เมื่อ r คือรัศมีและ h คือความสูง ในขั้นแรกรับค่ารัศมีและความสูงจากผู้ใช้ด้วยคำสั่ง `input()` และแปลงเป็นเลขทศนิยมด้วยคำสั่ง `float()` และกำหนดค่าให้ตัวแปร r และ h ตามลำดับ จากนั้นคำนวณปริมาตรด้วยนิพจน์ $\text{volume} = 1/3 * (\pi * r ** 2 * h)$ และแสดงผลด้วยคำสั่ง `print` ด้วยอักขระควบคุมการแสดงผล `%.2f` คือแสดงผลลัพธ์เป็นเลขทศนิยมสองตำแหน่ง ในตัวอย่างนี้ป้อนค่า $r = 4.97$ และ $h = 12.5$ ผลลัพธ์มีค่า "ปริมาตรกรวยที่มีรัศมี 4.97 หน่วยและสูง 12.5 หน่วย เท่ากับ 323.33 ลูกบาศก์หน่วย"

2.23 การเขียนโปรแกรมคำนวณปริมาตรทรงกลมและพื้นผิว

การคำนวณหาปริมาตรทรงกลมคำนวณจากสูตร $\text{volume} = \frac{4}{3}\pi \times r^3$ และพื้นที่ผิวจากสูตร $4 \times \pi \times r^2$



ตัวอย่างที่ 2.11 การเขียนโปรแกรมคำนวณปริมาตรทรงกลม

```
from math import pi
r = float(input("ป้อนรัศมี: "))
volume = 4/3 * pi * r ** 3
SA = 4 * pi * r ** 2
print("ปริมาตรทรงกลมที่มีรัศมี %.2f หน่วย เท่ากับ %.2f ลูกบาศก์หน่วย และพื้นที่ผิวเท่ากับ %.2f ตารางหน่วย"%(r,volume,SA))
```

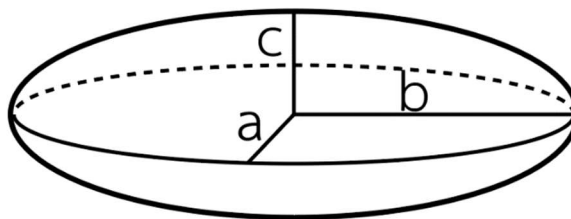
ผลลัพธ์ :

```
ป้อนรัศมี: 6.45
ปริมาตรทรงกลมที่มีรัศมี 6.45 หน่วย เท่ากับ 1124.00 ลูกบาศก์หน่วย และพื้นที่ผิวเท่ากับ 522.79 ตารางหน่วย
```

ตัวอย่างที่ 2.11 การคำนวณปริมาตรทรงกลมเมื่อทราบรัศมีคำนวณจากสูตรปริมาตรทรงกลมเท่ากับ $\frac{4}{3}\pi \times r^3$ และพื้นที่พื้นผิวเท่ากับ $4 \times \pi \times r^2$ เมื่อ r คือรัศมี ในขั้นแรกรับค่ารัศมีจากผู้ใช้งานด้วยคำสั่ง `input()` และแปลงเป็นเลขทศนิยมด้วยคำสั่ง `float()` และกำหนดค่าให้ตัวแปร `r` จากนั้นคำนวณปริมาตรด้วยนิพจน์ `volume = 4/3 * pi * r ** 3` และคำนวณพื้นที่พื้นผิว (Surface Area) ด้วยนิพจน์ `SA = 4 * pi * r ** 2` จากนั้นแสดงผลลัพธ์ด้วยคำสั่ง `print` ด้วยอักขระควบคุมการแสดงผล `%.2f` คือแสดงผลลัพธ์เป็นเลขทศนิยมสองตำแหน่ง ในตัวอย่างนี้ป้อนค่า `r = 6.45` ผลลัพธ์มีค่า "ปริมาตรทรงกลมที่มีรัศมี 6.45 หน่วย เท่ากับ 1124.00 ลูกบาศก์หน่วย และพื้นที่ผิวเท่ากับ 522.79 ตารางหน่วย"

2.24 การเขียนโปรแกรมคำนวณปริมาตรทรงรี

การคำนวณหาปริมาตรทรงรีคำนวณจากสูตร $\text{volume} = \frac{4}{3}\pi \times a \times b \times c$



ตัวอย่างที่ 2.12 การเขียนโปรแกรมคำนวณปริมาตรทรงกลม

```
from math import pi
a = float(input("ป้อน a: "))
b = float(input("ป้อน b: "))
c = float(input("ป้อน c: "))
volume = 4/3 * pi * a * b * c
print("ปริมาตรทรงรีที่มีระยะ a=%2f b=%2f และ c=%2f เท่ากับ %2f ลูกบาศก์หน่วย"%(a,b,c,volume))
```

ผลลัพธ์ :

```
ป้อน a: 16
ป้อน b: 41
ป้อน c: 24
ปริมาตรทรงรีที่มีระยะ a=16.00 b=41.00 และ c=24.00 เท่ากับ 65948.31 ลูกบาศก์หน่วย
```

ตัวอย่างที่ 2.12 การคำนวณปริมาตรทรงรีเมื่อทราบระยะ a b และ c คำนวณจากสูตรปริมาตรทรงรีเท่ากับ $\frac{4}{3}\pi \times a \times b \times c$ ขั้นแรกรับค่าระยะ a,b,c ด้วยคำสั่ง input() และแปลงเป็นทศนิยมด้วยคำสั่ง float() จากนั้นนำมาคำนวณหาปริมาตรทรงรีด้วยนิพจน์ $\text{volume} = \frac{4}{3} * \pi * a * b * c$ และพิมพ์ผลลัพธ์ ในตัวอย่างนี้กำหนด ค่า a = 16 b=41 และ c=24 ผลลัพธ์เท่ากับ "ปริมาตรทรงรีที่มีระยะ a=16.00 b=41.00 และ c=24.00 เท่ากับ 65948.31 ลูกบาศก์หน่วย"

2.25 โปรแกรมคำนวณทางคณิตศาสตร์ MathCAD5

โปรแกรม MathCAD5 เป็นเครื่องมือช่วยการคำนวณและแสดงผลด้วยสัญลักษณ์ทางคณิตศาสตร์ ช่วยให้การเรียนรู้คณิตศาสตร์เป็นเรื่องเข้าใจได้ง่ายสามารถพล็อตกราฟ การหาอนุพันธ์และอินทิเกรตได้

- 1) ดาวน์โหลดโปรแกรม mathcad5 ที่ลิงค์ <http://dsdi.msu.ac.th/programming/MathCAD5.zip> มีขนาด 5 MB จากนั้น extract และเรียกไฟล์  MCAD.EXE
- 2) พื้นที่ทำงานของโปรแกรม Matcad5 จะเป็นพื้นที่เปล่าและมีเครื่องหมาย + สีแดงสำหรับเป็นจุดเริ่มต้นของการเขียนสูตรและสมการทางคณิตศาสตร์

ตารางที่ 2.6 การพิมพ์สูตรและสมการคณิตศาสตร์ด้วย Matcad5

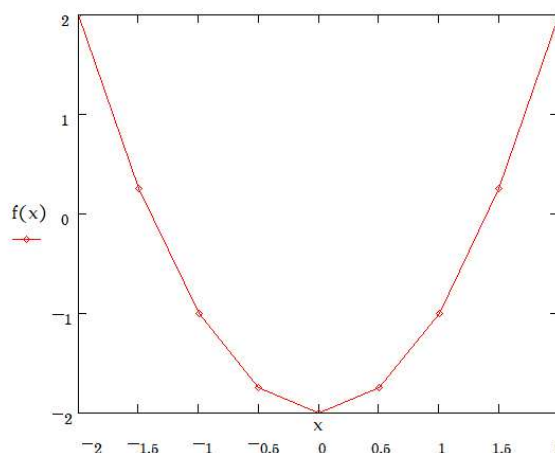
พิมพ์	ผลลัพธ์บนจอภาพ
      	$58 + 12 = 70$
          	$3.26 - 16.5 = -13.24$
    	$2^2 = 4$
        	$\sqrt{450 + 2} = 21.26$
          	$\sin\left(\frac{\pi}{2}\right) = 1$
              	$\text{adder}(x,y) := x + y$
           	$\text{adder}(3,2) = 5$
เลือกเมนู Math -> SmartMath จากนั้นเลือกไอคอน            	$\frac{d}{dx} \sin(x) \rightarrow \cos(x)$
เลือกไอคอน            	$\int \cos(x) dx \rightarrow \sin(x)$

3) การพล็อตกราฟด้วย Mathcad5

$$x := -2, -1.5 \dots 2$$

$$f(x) := x^2 - 2$$

x	f(x)
-2	2
-1.5	0.25
-1	-1
-0.5	-1.75
0	-2
0.5	-1.75
1	-1
1.5	0.25
2	2



ตัวแปร x กำหนดให้มีค่าระหว่าง -2 ถึง 2 โดยเพิ่มค่าทีละ 0.5 ฟังก์ชัน $f(x)$ คำนวณค่า $x^2 - 2$ เมื่อกำหนดค่า $x = 0$ ผลลัพธ์ $f(0)$ จะมีค่า -2 จากนั้นทำการพล็อตกราฟให้แนวนอนแทนตัวแปร x และแนวตั้งแทนตัวแปร $f(x)$ ผลลัพธ์แสดงดังภาพด้านบน โดยมีขั้นตอนดังนี้

ขั้นที่ 1: พิมพ์ 

ขั้นที่ 2: พิมพ์ 

ขั้นที่ 3: พิมพ์ 

ขั้นที่ 4: พิมพ์      

ขั้นที่ 5: เลือกไอคอน  พิมพ์       

แบบฝึกหัดท้ายบท

1. จงอธิบายความหมายของเครื่องหมายดำเนินการและนิพจน์
2. จงอธิบายความหมายของเครื่องหมายดำเนินการทางคณิตศาสตร์แต่ละประเภท
3. จงอธิบายความหมายของเครื่องหมายดำเนินการทางตรรกศาสตร์
4. ให้นิสิตอธิบายหลักการทำงานของวงจรวกเลข
5. จงอธิบายความหมายและของเลขลบแต่ละชนิด
6. จงอธิบายวิธีการแปลงสมการคณิตศาสตร์เป็นนิพจน์และคำนวณผลลัพธ์โดยตรวจสอบคำตอบกับโปรแกรม Matcad5
7. ตัวแปร a b และ c ให้เขียนนิพจน์เพื่อหาค่าของ a ซึ่งเกิดจาก b ยกกำลังสองบวกกับ c
8. ตัวแปร a b และ c ให้เขียนนิพจน์เพื่อหาค่าของ a ซึ่งเกิดจาก a คูณกับ c แล้วหารด้วย b
9. ตัวแปร a b และ c ให้เขียนนิพจน์เปรียบเทียบ a บวก b มีค่ามากกว่า a บวกด้วย c หรือไม่

บทที่ 3

ฟังก์ชันและการควบคุม

แนวคิด

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ การสร้างฟังก์ชัน การสร้างช่วงตัวเลขด้วย range() การวนรอบด้วย for การเปรียบเทียบด้วย if/else ตัวดำเนินการตรวจสอบเงื่อนไข การยุติหรือดำเนินการต่อกับคำสั่ง break และ continue การวนรอบด้วยคำสั่ง while และการควบคุมความผิดพลาดด้วย Exception

วัตถุประสงค์

1. อธิบายความหมายและการสร้างฟังก์ชันได้
2. อธิบายการสร้างช่วงตัวเลขด้วย range() ได้
3. อธิบายการวนรอบด้วย for และ while ได้
4. อธิบายการเปรียบเทียบด้วย if/else ได้
5. อธิบายตัวดำเนินการตรวจสอบเงื่อนไขได้
6. อธิบายการยุติหรือดำเนินการต่อกับคำสั่ง break และ continue ได้
7. อธิบายและการควบคุมความผิดพลาดด้วย Exception ได้

3.1 ความหมายและประเภทของฟังก์ชัน

ฟังก์ชัน (Function) คือ คำสั่ง มีอยู่ 2 แบบ คือ 1) คำสั่งที่สร้างขึ้นจากคนอื่น โดยที่ผู้ใช้ไม่ต้องเขียนชุดคำสั่งขึ้นเองเพียงแต่เรียกใช้งานเท่านั้น 2) คำสั่งที่เขียนขึ้นมาใช้งานเอง

ฟังก์ชันจะมีการคืนค่าเพื่อนำค่าไปใช้งานต่อไป โดยการคืนค่าจะเขียนคำสั่ง return ภายในฟังก์ชัน นอกจากนั้นฟังก์ชันจะมีชื่อเรียก ชื่อของฟังก์ชันจะอยู่หน้าเครื่องหมาย() เช่น print() dir() sqrt() sin() เป็นต้น การเรียกใช้งานฟังก์ชันจะมีสิ่งที่ส่งเข้าไปในฟังก์ชัน อยู่ภายในเครื่องหมายวงเล็บ ซึ่งจะมีหรือไม่ขึ้นกับผู้ออกแบบฟังก์ชัน

ฟังก์ชันที่สร้างโดยคนอื่นเป็นฟังก์ชันที่พัฒนาขึ้นมาจากนักพัฒนาคนอื่น ๆ และเผยแพร่ให้ใช้งาน ฟังก์ชันที่รวมรวมและเก็บเข้าไว้ด้วยกันจะเรียกว่า Library หรือ Module หากต้องการเรียกใช้ฟังก์ชันจะใช้คำสั่ง import ซึ่งมีอยู่ 2 รูปแบบ

- import math คือ การอิมพอร์ตคำสั่งทั้งหมดภายใน math เข้ามาในโปรแกรมหากต้องการใช้คำสั่ง sin จะต้องเรียกชื่อโมดูลและตามด้วยฟังก์ชันที่ต้องการใช้งาน เช่น math.sin(45) เป็นต้น
- from math import sin คือ การอิมพอร์ตเฉพาะคำสั่ง sin() เข้ามาเท่านั้น เรียกใช้ด้วยคำสั่ง sin(45)

3.2 ฟังก์ชันที่สร้างขึ้นด้วยตนเอง

การสร้างฟังก์ชัน มีรูปแบบ ดังนี้

รูปแบบการสร้างฟังก์ชัน:

```
def ชื่อฟังก์ชัน():
    คำสั่ง.....
    return ผลลัพธ์
```

ตัวอย่างที่ 3.1 การสร้างฟังก์ชัน adder

```
def adder(a, b):
    return a+b
c = adder(3,2)
d = adder(adder(1,1), adder(2,2))
print("ตัวแปร c มีค่า %d"%c)
print("ตัวแปร d มีค่า %d"%d)
```

ผลลัพธ์ :

```
ตัวแปร c มีค่า 5
ตัวแปร d มีค่า 6
```

ตัวอย่างที่ 3.1 เป็นการสร้างฟังก์ชันชื่อ adder() มีค่าที่ส่งเข้าไปให้ฟังก์ชันหรืออาร์กิวเมนต์ 2 ตัว คือ a และ b ภายในฟังก์ชันมีคำสั่ง return a+b เมื่อเรียกใช้คำสั่ง c = adder(3,2) เลข 3 คืออาร์กิวเมนต์ a และเลข 2 คืออาร์กิวเมนต์ b ผลลัพธ์จะแสดงคำว่า "ตัวแปร c มีค่า 5" จากนั้นตัวแปร d เรียกฟังก์ชัน adder(adder(1,1), adder(2,2)) ให้พิจารณา adder(1,1) มีค่าเท่ากับเลข 2 และ adder(2,2) เท่ากับ 4 และจะถือว่าเลข 2 และ 4 เป็นอาร์กิวเมนต์ของฟังก์ชัน adder(2,4) ซึ่ง คือ 6 ดังนั้น ผลลัพธ์ในลำดับต่อมาแสดงผลคำว่า "ตัวแปร d มีค่า 6"

3.3 การสร้างฟังก์ชันด้วยคำสั่ง lambda

การสร้างฟังก์ชันด้วยคำสั่ง lambda เป็นฟังก์ชันที่ไม่มีชื่อ เรียกว่า แอนโนนิมัสฟังก์ชันมีข้อดีคือ สามารถสร้างฟังก์ชันขึ้นมาได้ในขณะที่โปรแกรมกำลังทำงาน (runtime)

ไวยากรณ์

```
lambda arguments : expression
```

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
arguments	คือ อาร์กิวเมนต์ของฟังก์ชัน
expression	คือ การดำเนินการและคืนค่าของฟังก์ชัน

ตัวอย่างที่ 3.2 การสร้างฟังก์ชันด้วยคำสั่ง lambda

```
adder = lambda a,b : a+b
c = adder(3,2)
```

```
print("ตัวแปร c มีค่า %d"%c)
```

ผลลัพธ์ :

```
ตัวแปร c มีค่า 5
```

ตัวอย่าง 3.2 จะเห็นว่าตัวอย่างที่ผ่านมาเป็นการสร้างฟังก์ชัน adder ด้วยคำสั่ง def ซึ่งรับอาร์กิวเมนต์ 2 ตัว คือ a และ b โดยคืนค่าเป็นผลรวมของ a+b ในตัวอย่างนี้เป็นการสร้างฟังก์ชันเหมือนตัวอย่างที่ผ่านมาแต่ใช้คำสั่ง lambda

3.4 คำสั่ง for

ในภาษาไพธอนมีการวนรอบ 2 วิธี คือ for และ while คำสั่ง for มีรูปแบบคำสั่งดังนี้

```
for variable in range(start,stop):
    statement
```

ตัวอย่างที่ 3.3 การวนรอบสร้างแม่สูตรคูณ

```
for i in range(1,6):
    print("9 x %d = %d"%(i, 9*i))
```

ผลลัพธ์ :

```
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
```

ตัวอย่างที่ 3.3 คำสั่ง range(1,6) จะสร้างตัวเลข [1,2,3,4,5] โดยค่าปลายทางจะถูกกลบด้วย 1 เสมอ บรรทัดคำสั่ง print() จะต้องเยื้องออกมาด้านขวา เพื่อบอกว่าเป็นการทำงานภายในการวนรอบของ คำสั่ง for และเรียกคำสั่ง print("9 x %d = %d"%(i, 9*i)) โดยมีอักขระควบคุมการแสดงผลสองตัว คือ %d และ %d เป็นการแสดงผลเลขจำนวนเต็ม โดยค่าที่แสดงใน %d ตัวแรกคือค่าของตัวแปร i ซึ่งในแต่ละรอบจะมีค่าเป็น 1 2 3 4 5 ตามลำดับ ส่วน %d ตัวที่สองด้านขวาจะแสดงค่าตัวแปร 9*i คือผลคูณของเลข 9 กับค่า i ในแต่ละรอบ เช่น 9*1 , 9*2, .. 9*5 ตามลำดับ

การวนรอบโดยระบุค่าเริ่มต้น(start) ค่าสุดท้าย (stop) และค่าการเพิ่มหรือลด (step) ดังนี้

```
for variable in range(start, stop, step):
    statement
```

ตัวอย่างที่ 3.4 การวนรอบสร้างแม่สูตรคูณโดยกำหนดค่าเริ่มต้นและค่าสุดท้ายและค่าการเพิ่ม

```
for i in range(1,12,3):
    print("9 x %d = %d"%(i, 9*i))
```

ผลลัพธ์ :

```
9 x 1 = 9
9 x 4 = 36
9 x 7 = 63
9 x 10 = 90
```

ตัวอย่างที่ 3.4 เป็นการวนรอบ 1 ถึง 12 โดยเพิ่มค่าทีละ 3 ดังนั้น จะวนรอบจาก [1, 4, 7, 10] โดยเพิ่มทีละ 3 โดยการทำงานจะเหมือนกับตัวอย่างที่ 3.2 ต่างกันที่ค่าการเพิ่ม โดยในตัวอย่าง 3.2 จะเพิ่มทีละ 1 ส่วนตัวอย่างนี้จะเพิ่มทีละ 3

3.5 คำสั่ง while

รูปแบบของคำสั่ง while ดังนี้

```
while condition:
    statement
```

ตัวอย่างที่ 3.5 การวนรอบสร้างแม่สูตรคูณด้วย while loop

```
i = 1
while i <= 4:
    print("9 x %d = %d"%(i, 9*i))
    i+=1
```

ผลลัพธ์ :

```
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
```

ตัวอย่าง 3.5 เป็นการวนรอบทำงานจนกว่าเงื่อนไขจะเป็นเท็จจึงจะหยุดการทำงาน ในตอนเริ่มต้น กำหนดค่า $i = 1$ การวนรอบเปรียบเทียบค่า $i \leq 4$ หมายถึงจะยังคงทำงานหากค่า i น้อยกว่าหรือเท่ากับ 4 ในแต่ละรอบจะทำการพิมพ์ `print("9 x %d = %d"%(i, 9*i))` เมื่อ %d ตัวด้านซ้ายแทนด้วยค่า i และ %d ด้านขวาแทนด้วย $9 * i$ คำสั่ง `i+=1` มีค่าคือ $i = i+1$ ดังนั้น ค่า i จะเพิ่มขึ้นทีละ 1 ทุกครั้งในแต่ละรอบ และรอบสุดท้าย i มีค่า = 5 จะยุติการวนรอบเพราะเงื่อนไขเป็นเท็จ

3.6 คำสั่ง if / else

รูปแบบของคำสั่ง while ดังนี้

```
if condition:
    statement
elif condition:
```

```
statement
else:
statement
```

ตัวอย่างที่ 3.6 การเปรียบเทียบหลายทางเลือก

```
ดิน = "แห้ง"
if ดิน == "แห้ง":
    print("รดน้ำต้นไม้")
elif ดิน == "เปียก":
    print("ปิดวาล์วน้ำ")
else:
    print("งง ไม่รู้เรื่อง")
```

ผลลัพธ์ :

```
รดน้ำต้นไม้
```

ตัวอย่าง 3.6 เป็นการเปรียบเทียบหลายทางเลือก โดยมีตัวแปร ดิน = "แห้ง" เครื่องหมาย == หมายถึงการเปรียบเทียบว่าเท่ากันหรือไม่ เช่น ดิน == "แห้ง" มีค่าจริงเพราะตอนเริ่มต้นประกาศว่า ดิน = "แห้ง" ดังนั้นโปรแกรมจะทำงานที่บล็อกที่เป็นจริง ในการตรวจสอบเงื่อนไขหากไม่เป็นจริงใช้คำสั่ง elif ซึ่งโปรแกรมจะดำเนินการในบล็อกที่เป็นจริง และถ้าหากไม่สอดคล้องกับเงื่อนไขใด ๆ เลย โปรแกรมจะเข้าไปทำงานที่บล็อกของคำสั่ง else: ในตัวอย่างนี้ ดิน=="แห้ง" จึงแสดงคำว่า "รดน้ำต้นไม้"

3.7 คำสั่ง break และ continue

รูปแบบของคำสั่ง break และ continue ดังนี้

```
if condition:
    continue
else:
    break
```

ตัวอย่างที่ 3.7 คำสั่ง break และ continue

```
ระยะทาง = range(10)
ใบขับขี่ = False
print("วันนี้ฉันจะกลับบ้าน บ้านฉันห่างจากนี้ %d กม. วันนี้ใบขับขี่ฉันมีสถานะเป็น %d"%(max(
ระยะทาง), ใบขับขี่))
for i in ระยะทาง:
    print(">> สบายใจฉันกำลังกลับบ้าน กิโลเมตรที่ : %d"%(i))
    if i == 5:
        print("\t>> ฉัน : ตายละตำรวจ ๆ")
```

```

print("\t>> ตำรวจ : ขอตรวจใบขับขี่ด้วยครับ :)")
if ใบขับขี่ == True:
    print("\t>> ฉันทัน : แปปนึ่งนะครับ นี๋ครับ !!")
    print("\t>> ตำรวจ : โอเค เดินทางกลับบ้านโดยสวัสดิภาพนะ !!")
    continue
else:
    print("\t>> ฉันทัน : อืม.. ลืมเอามาอะ !!")
    print("\t>> ตำรวจ : งั้นวันนี้ไม่ต้องกลับบ้านนะ !!")
    break

```

ผลลัพธ์ :

```

วันนี้ฉันทันจะกลับบ้าน บ้านฉันทันห่างจากนี้ 9 กม. วันนี้ใบขับขี่ฉันทันมีสถานะเป็น 0
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 0
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 1
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 2
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 3
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 4
>> สบายใจฉันทันกำลังกลับบ้าน กิโลเมตรที่ : 5
    >> ฉันทัน : ตายละตำรวจ ๆ
    >> ตำรวจ : ขอตรวจใบขับขี่ด้วยครับ :)
    >> ฉันทัน : อืม.. ลืมเอามาอะ !!
    >> ตำรวจ : งั้นวันนี้ไม่ต้องกลับบ้านนะ !!

```

ตัวอย่าง 3.7 ประกาศตัวแปร 2 ตัว คือ ระยะทาง = range(10) จะมีค่าเท่ากับ [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] และตัวแปร ใบขับขี่ = False แทนการไม่มีใบขับขี่ คำสั่ง `print("วันนี้ฉันทันจะกลับบ้าน บ้านฉันทันห่างจากนี้ %d กม. วันนี้ใบขับขี่ฉันทันมีสถานะเป็น %d"% (max(ระยะทาง), ใบขับขี่))` จะแสดงคำว่า "วันนี้ฉันทันจะกลับบ้าน บ้านฉันทันห่างจากนี้ 9 กม. วันนี้ใบขับขี่ฉันทันมีสถานะเป็น 0" เลข 9 คือ max(ระยะทาง) และสถานะใบขับขี่มีค่า 0 หรือ False

การวนรอบจะวนรอบทั้งหมด 9 รอบ โดยในรอบที่ 5 จะมีการตรวจสอบใบขับขี่ ถ้าไม่มีใบขับขี่จะยุติการทำงานซึ่งเป็นผลจากคำสั่ง `break` แต่ถ้ามีใบขับขี่คือมีค่าเป็น 1 หรือ True จะทำงานจนครบทุกรอบเป็นผลจากคำสั่ง `continue`

3.8 ตัวแปรแบบ Global และ Local

ในฟังก์ชันสามารถประกาศตัวแปรภายในได้แต่จะมีคุณสมบัติเป็น private คือ ใช้ภายในฟังก์ชันเท่านั้น หากต้องการให้ตัวแปรภายนอกสามารถใช้งานภายในฟังก์ชันได้ ให้ประกาศคำสั่ง global ตามด้วยชื่อตัวแปร พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 3.8 การประกาศตัวแปร global และ local

```
def ฉั่นคือเอ1():
    a = 7
    print("ฉั่นชื่อ a : ฉั่นอยู่ในฟังก์ชัน \"ฉั่นคือเอ1()\" -> ฉั่นมีค่าเท่ากับ ",a)
    return a

def ฉั่นคือเอ2():
    global a
    print("ฉั่นชื่อ a : ฉั่นอยู่ในฟังก์ชัน \"ฉั่นคือเอ2()\" -> ฉั่นมีค่าเท่ากับ ",a)
    return a

a = 20
z = ฉั่นคือเอ1()
z = ฉั่นคือเอ2()
```

ผลลัพธ์ :

```
ฉั่นชื่อ a : ฉั่นอยู่ในฟังก์ชัน "ฉั่นคือเอ1()" -> ฉั่นมีค่าเท่ากับ 7
ฉั่นชื่อ a : ฉั่นอยู่ในฟังก์ชัน "ฉั่นคือเอ2()" -> ฉั่นมีค่าเท่ากับ 20
```

ตัวอย่างที่ 3.8 มีฟังก์ชัน 2 ฟังก์ชัน ชื่อว่า ฉั่นคือเอ1() และ ฉั่นคือเอ2() ในฟังก์ชัน ฉั่นคือเอ1() มีตัวแปร a เป็น local ดังนั้น จะใช้เฉพาะภายในฟังก์ชันฉั่นคือเอ1() เท่านั้น และถูกประกาศให้มีค่าเท่ากับ 7 ในขณะที่ฟังก์ชัน ฉั่นคือเอ2() มีตัวแปร a เป็น global คือ ใช้งานจากภายนอกฟังก์ชัน ในตอนเริ่มต้นทำงาน กำหนดให้ a=20 จากนั้นเรียกฟังก์ชัน ฉั่นคือเอ1() จะแสดงค่าภายในซึ่งเป็น local variable จึงแสดงค่า a มีค่าเท่ากับ 7 หลังจากนั้นเรียกฟังก์ชัน ฉั่นคือเอ2() ซึ่งมีตัวแปร a เป็น global variable ทำให้มีค่าเท่ากับ 20

3.9 การตรวจสอบความผิดพลาด (Exception Handling)

การเขียนโปรแกรมที่มีความผิดพลาดโดยปกติจะยุติการทำงานทันที แต่จะอย่างไรให้โปรแกรมยังคงทำงานต่อไปได้ เรียกวิธีการนี้ว่า Exception Handling โดยใช้คำสั่ง try / except

รูปแบบของคำสั่ง try except ดังนี้

```
try:
    statement
except:
    statement
```

ตัวอย่างที่ 3.9 การตรวจจับความผิดพลาด

```
try:
    7/0
except:
```

```
print("กรุณารหัสด้วยเลขอื่นที่ไม่ใช่ศูนย์")
```

ผลลัพธ์ :

```
กรุณารหัสด้วยเลขอื่นที่ไม่ใช่ศูนย์
```

ตัวอย่างที่ 3.9 เป็นการตรวจจับข้อผิดพลาดที่อาจเกิดขึ้นในขณะที่โปรแกรมกำลังทำงาน เพราะโดยปกติหากเกิดความผิดพลาดจะทำให้โปรแกรมยุติการทำงาน ดังนั้น คำสั่ง try เป็นการให้พยายามทำคำสั่งนั้น ๆ ในตัวอย่างนี้ให้คำนวณ $7/0$ ซึ่งจะเกิดข้อผิดพลาดเพราะเป็นการหารตัวเลขด้วย 0 ซึ่งไม่อนุญาตให้หารเลขด้วย 0 ดังนั้นเมื่อเกิดความผิดพลาดจึงย้ายมาทำงานในส่วนของ except และแสดงคำว่า "กรุณารหัสด้วยเลขอื่นที่ไม่ใช่ศูนย์"

3.10 การเขียนโปรแกรมคำนวณผลรวมของข้อมูลยกกำลังสองด้วย for()

สูตรการคำนวณค่าเฉลี่ย

$$\sum_{i=1}^n x^2$$

ตัวอย่างที่ 3.10 การเขียนโปรแกรมคำนวณผลรวมของข้อมูลยกกำลังสองด้วย for()

```
x=[3, 1, 2]
x2 = []
for i in x:
    x2.append(i ** 2)
print("ค่าผลรวมของข้อมูลยกกำลังสองของ %s = %.2f" % (str(x),sum(x2)))
```

ผลลัพธ์ :

```
ค่าผลรวมของข้อมูลยกกำลังสองของ [3, 1, 2] = 14.00
```

ตัวอย่าง 3.x เป็นการคำนวณผลรวมของข้อมูลแต่ละตัวยกกำลังสอง เมื่อข้อมูลคือ $x=[3,1,2]$ เมื่อนำมายกกำลังสอง จะได้ $x2=[3^2, 1^2, 2^2]$ เท่ากับ $x2=[9, 1, 4]$ เมื่อคำสั่ง `append()` คือการเพิ่มข้อมูลเข้าไปในตัวแปร `x2` โดยเป็นการนำสมาชิกแต่ละตัวในตัวแปร `x` มายกกำลังสอง หลังจากแสดงค่าผลลัพธ์มีค่าคือ "ค่าผลรวมของข้อมูลยกกำลังสองของ $[3,1,2] = 14.00$ " เมื่อคำสั่ง `str(x)` คือการแปลงตัวแปร `x` ซึ่งเป็นชนิดลิสต์ให้เป็นสตริงและแสดงผลในอักขระควบคุมการแสดงผล `%s` และ `%.2f` (แสดงทศนิยมสองตำแหน่ง)

แบบฝึกหัดท้ายบท

1. จงเขียนโปรแกรมสร้างฟังก์ชัน `subtract()` มีอาร์กิวเมนต์ 2 ตัวและนำมาลบกัน
2. จงเขียนแม่สูตรคูณด้วยการวนรอบด้วย `for` และ `while`
3. จงเขียนการเปรียบเทียบด้วย `if/else`
4. จงเขียนโปรแกรมตัดเกรดหากคะแนนเกิน 50 พิมพ์สอบผ่าน นอกนั้นพิมพ์ สอบตก
5. จงอธิบายคำสั่ง `break` และ `continue`
7. จงอธิบายการควบคุมความผิดพลาดด้วย Exception

บทที่ 4

ตัวแปรลิสต์และทูเพิล

แนวคิด

ไพธอนมีความยืดหยุ่นในการสร้างลิสต์ ซึ่งคล้ายกับอาร์เรย์ในภาษาซีและภาษาปาสคาลหรือภาษาอื่น ๆ ภายในตัวแปรลิสต์อนุญาตให้ข้อมูลภายในเป็นตัวแปรชนิดใด ๆ ได้อย่างไม่จำกัด แม้แต่ตัวแปรลิสต์สามารถเก็บภายในตัวแปรลิสต์ได้ ทำให้สามารถสร้างโครงสร้างข้อมูลที่มีความซับซ้อนและสามารถประยุกต์ใช้ในงานต่าง ๆ ได้อย่างกว้างขวาง

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ การประกาศตัวแปรลิสต์และทูเพิล การเข้าถึงตำแหน่งภายในตัวแปรลิสต์และทูเพิล การเพิ่มและลบตัวแปรลิสต์ การเรียง สลับตำแหน่งหลังไปหน้า การจัดสมาชิกที่ซ้ำกันในลิสต์ และฟังก์ชันต่าง ๆ ของตัวแปรลิสต์และทูเพิล

วัตถุประสงค์

1. อธิบายการประกาศตัวแปรลิสต์และทูเพิลได้
2. อธิบายการเข้าถึงตำแหน่งภายในตัวแปรลิสต์และทูเพิลได้
3. อธิบายการเพิ่มและลบตัวแปรลิสต์ได้
4. อธิบายการเรียง สลับตำแหน่งหลังไปหน้าได้
5. อธิบายการจัดสมาชิกที่ซ้ำกันในลิสต์ได้

4.1 การประกาศตัวแปรชนิดลิสต์

การประกาศตัวแปรชนิดลิสต์สามารถทำได้โดยใช้เครื่องหมาย [และ] ล้อมรอบสมาชิกเอาไว้ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.1 การเขียนโปรแกรมคำนวณผลรวมของข้อมูลยกกำลังสองด้วย for()

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]  
b = [ 'Google', 'Gmail', 'Youtube' ]
```

ตัวอย่าง 4.1 ตัวแปร a เก็บรายการกลุ่มตัวเลข ส่วนตัวแปร b เก็บรายการสตริง

4.2 ตำแหน่งสมาชิกย่อยภายในลิสต์

การเข้าถึงสมาชิกภายในตัวแปรลิสต์ระบุได้สองวิธี คือ ระบุด้วยเลขจำนวนเต็มบวก และจำนวนเต็มลบ

ตัวอย่างที่ 4.2 การระบุตำแหน่งของสมาชิกย่อยภายในลิสต์

ตัวแปร a =	[3	4	2	3	4	9	4	7	3	4]
ระบุตำแหน่งด้วยเลขบวก	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
ระบุตำแหน่งด้วยเลขลบ	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]

ตัวอย่าง 4.2 สมาชิกตัวแรกสุดแทนคือตำแหน่ง `a[0]` มีค่า 3 ส่วนสมาชิกตำแหน่งหลังสุดคือ `a[-1]` มีค่า 4

4.3 การเข้าถึงสมาชิกภายในลิสต์

รูปแบบการเข้าถึงสมาชิกจะอยู่ภายในเครื่องหมายก้ามปู ดังนี้

```
list_name[index]
```

เมื่อ ตัวแปร `list_name` แทนชื่อตัวแปรลิสต์ และ `index` แทนหมายเลขลำดับของสมาชิกภายในลิสต์ ต้องเป็นเลขจำนวนเต็มเท่านั้น

ตัวอย่างที่ 4.3 การเข้าถึงสมาชิกภายในตัวแปรลิสต์

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]
b = [ 'Google', 'Gmail', 'Youtube' ]
print(a[0], a[1], a[2])
print(a[-1], a[-2], a[-3])
print(b[0], b[1], b[2])
print(b[-1], b[-2], b[-3])
```

ผลลัพธ์ :

```
3 4 2
4 3 7
Google Gmail Youtube
Youtube Gmail Google
```

ตัวอย่าง 4.3 การเข้าถึงตำแหน่งสมาชิกเลขบวกที่ตำแหน่งที่ `a[0]`, `a[1]`, `a[2]` มีค่า 3 4 และ 2 ตามลำดับ ส่วนการเข้าถึงด้วยเลขลบ ตำแหน่ง `a[-1]`, `a[-2]`, `a[-3]` มีค่า 4 3 และ 7

การเข้าถึงตัวแปร `b` ที่ตำแหน่ง `b[0]`, `b[1]`, `b[2]` มีค่าเท่ากับ Google Gmail Youtube และการเข้าถึงตำแหน่งด้วยเลขลบที่ตำแหน่ง `b[-1]`, `b[-2]`, `b[-3]` มีค่าเท่ากับ Youtube Gmail Google

4.4 การตรวจสอบขนาดของลิสต์ด้วยคำสั่ง `len()`

ฟังก์ชัน `len()` ใช้ตรวจสอบจำนวนสมาชิกของตัวแปรลิสต์ ดังนี้

ตัวอย่างที่ 4.4 การเข้าถึงสมาชิกภายในตัวแปรลิสต์

```
b = [ 'Google', 'Gmail', 'Youtube' ]
print(len(b))
```

ผลลัพธ์ :

```
3
```

ตัวอย่าง 4.4 การตรวจสอบจำนวนสมาชิกของตัวแปรลิสต์ด้วยคำสั่ง len() ในตัวอย่างนี้ตัวแปร b มีจำนวนสมาชิก 3 ตัว คือ "Google" , "Gmail" และ "Youtube"

4.5 การวนเข้าไปอ่านค่าภายในลิสต์ด้วยคำสั่ง for

การวนเข้าไปในสมาชิกของตัวแปรลิสต์ด้วยคำสั่ง for ดังนี้

ตัวอย่างที่ 4.5 การวนเข้าไปในสมาชิกของตัวแปรลิสต์ด้วยคำสั่ง for

```
b = ['Google', 'Gmail', 'Youtube']
for i in b:
    print(i)
```

ผลลัพธ์ :

```
Google
Gmail
Youtube
```

ตัวอย่าง 4.5 การวนเข้าไปในลิสต์ด้วยคำสั่ง for จะเป็นการวนเข้าไปในสมาชิกลำดับที่ 0 ถึงลำดับสุดท้าย ตัวตัวแปร i จะถูกนำมาใช้แทนค่าสมาชิกในแต่ละรอบ ในรอบแรกตัวแปร i มีค่า Google รอบต่อมา มีค่า Gmail และ Youtube ตามลำดับ

4.6 การวนเข้าไปอ่านค่าภายในลิสต์ด้วยคำสั่ง for และ range

การวนเข้าไปในสมาชิกของตัวแปรลิสต์ด้วยคำสั่ง for ด้วยขนาดของตัวแปรลิสต์ จะวนจากลำดับที่ 0 ถึงลำดับสุดท้ายของตัวแปรโดยกำหนดด้วยคำสั่ง len(ชื่อตัวแปร) ดังนี้

ตัวอย่างที่ 4.6 การวนเข้าไปในสมาชิกของตัวแปรลิสต์ด้วยคำสั่ง for และ range

```
b = ['Google', 'Gmail', 'Youtube']
for i in range(len(b)):
    print(i, b[i])
```

ผลลัพธ์ :

```
0 Google
1 Gmail
2 Youtube
```

ตัวอย่าง 4.6 การวนเข้าไปในลิสต์ด้วยคำสั่ง for และ range(len(b)) คือ การวนจากสมาชิกตัวที่ 0 ไปยังตำแหน่งสุดท้าย โดยระบุตำแหน่งสุดท้ายด้วยขนาดของสมาชิกผ่านคำสั่ง len(b) ในการเข้าถึงสมาชิกด้วยตัวเลขจะต้องระบุภายในเครื่องหมายก้ามปู

4.7 การแก้ไขค่าตัวแปรภายในสมาชิกย่อย

การแก้ไขค่าตัวแปรลิสต์สามารถระบุด้วยหมายเลขลำดับและกำหนดค่าผ่านเครื่องหมายเท่ากับ ซึ่งมีรูปแบบดังนี้

```
list_name[index] = value
```

เมื่อ ตัวแปร list_name แทนชื่อตัวแปรลิสต์และ index แทนหมายเลขลำดับของสมาชิกภายในลิสต์ที่ต้องการแก้ไข โดยสามารถกำหนดค่าใหม่ด้วยเครื่องหมายเท่ากับและกำหนดค่าลงได้โดยตรง เช่น

ตัวอย่างที่ 4.7 การแก้ไขค่าตัวแปรภายในสมาชิกย่อย

```
b = [ 'Google', 'Gmail', 'Youtube' ]
b[1] = "Yahoo"
print(b)
```

ผลลัพธ์ :

```
['Google', 'Yahoo', 'Youtube']
```

ตัวอย่าง 4.7 การแก้ไขสมาชิกลำดับที่ 1 ของตัวแปร b จากเดิมมีค่า "Gmail" สามารถเปลี่ยนค่าได้ด้วยคำสั่ง b[1] = "Yahoo" เมื่อพิมพ์ค่าตัวแปร b จะพบว่าตำแหน่งที่ 0 มีค่าใหม่คือคำว่า "Yahoo"

4.8 การเลื่อนลิสต์

การเข้าถึงสมาชิกภายในลิสต์ด้วยการสไลด์จะใช้เครื่องหมายโคลอน (:) เช่น

ตัวอย่างที่ 4.8 การแก้ไขค่าตัวแปรภายในสมาชิกย่อย

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
print(x[:])      # สมาชิกย่อยทั้งหมดในลิสต์
print(x[:5])     # สมาชิกย่อยจากตำแหน่งเริ่มต้นถึงตำแหน่งที่ 4
print(x[5:])     # สมาชิกย่อยจากตำแหน่งที่ 5 ถึงตำแหน่งสุดท้าย
print(x[3:6])    # สมาชิกย่อยจากตำแหน่งที่ 3 ถึงตำแหน่งที่ 6
print(x[1:9:2])  # สมาชิกจากตำแหน่งที่ 1 ถึงตำแหน่งที่ 9 โดยเพิ่มค่าทีละ 2
print(x[-1:-5:-1]) # สมาชิกจากตำแหน่ง -1 ถึงตำแหน่ง -5 โดยลดค่าทีละ 1
print(x[:-3])   # สมาชิกย่อยจากตำแหน่งเริ่มต้น ถึงตำแหน่ง -3
print(x[-3:])   # สมาชิกย่อยจากตำแหน่งเริ่มต้นคือ -3 จนถึงตำแหน่งสุดท้าย
print(x)        # สมาชิกย่อยทั้งหมดในลิสต์
```

ผลลัพธ์ :

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
[1, 2, 3, 4, 5]
[6, 7, 8, 9, 0]
[4, 5, 6]
```

```
[2, 4, 6, 8]
[0, 9, 8, 7]
[1, 2, 3, 4, 5, 6, 7]
[8, 9, 0]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
```

4.9 การลบสมาชิกภายในตัวแปรลิสต์

การลบสมาชิกภายในตัวแปรลิสต์ใช้คำสั่ง `del` ตามด้วยหมายเลขสมาชิกภายในตัวแปรลิสต์สามารถกำหนดโดยใช้การเลื่อนลิสต์ได้ ดังนี้

ตัวอย่างที่ 4.9 การลบสมาชิกภายในตัวแปรลิสต์

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
del x[:]      # ลบสมาชิกย่อยทั้งหมดในลิสต์
print(x)
```

ผลลัพธ์ :

```
[ ]
```

ตัวอย่าง 4.9 เป็นการลบสมาชิกทุกตัวภายในตัวแปรลิสต์ ดังนั้นผลลัพธ์จึงเป็นลิสต์ว่าง []

ตัวอย่างที่ 4.10 ลบสมาชิกย่อยจากตำแหน่งที่ 4 ถึงตำแหน่งสุดท้าย

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
del x[4:]      # ลบสมาชิกย่อยจากตำแหน่งที่ 4 ถึงตำแหน่งสุดท้าย
print(x)
```

ผลลัพธ์ :

```
[1, 2, 3, 4]
```

ตัวอย่าง 4.10 เป็นการลบสมาชิกจากตำแหน่งที่ 4 ถึงตำแหน่งสุดท้าย ผลลัพธ์เหลือสมาชิก ดังนี้
[1, 2, 3, 4]

ตัวอย่างที่ 4.11 ลบสมาชิกย่อยจากตำแหน่ง 3 ถึง 6

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
del x[3:6]      # ลบสมาชิกย่อยทั้งหมดในลิสต์
print(x)
```

ผลลัพธ์ :

```
[1, 2, 3, 7, 8, 9, 0]
```

ตัวอย่าง 4.11 เป็นการลบสมาชิกในตำแหน่งที่ 3 ถึง 6 ผลลัพธ์เหลือสมาชิกดังนี้ [1, 2, 3, 7, 8, 9, 0]

4.10 การเพิ่มข้อมูลเข้าไปในลิสต์ด้วยคำสั่ง append

ฟังก์ชัน append() ทำหน้าที่เพิ่มค่าเข้าต่อท้ายรายการลิสต์ ดังนี้

ตัวอย่างที่ 4.12 การลบสมาชิกภายในตัวแปรลิสต์

```
x = ["Python", "Dart"]
x.append("GoLang")
print(x)
```

ผลลัพธ์ :

```
['Python', 'Dart', 'GoLang']
```

ตัวอย่าง 4.12 ในตอนเริ่มต้นประกาศให้ตัวแปร x มีสมาชิก 2 ตัวคือ "Python" และ "Dart" จากนั้นทำการเพิ่ม "GoLang" ลงในตัวแปร x ด้วยคำสั่ง append() เมื่อพิมพ์คำสั่ง x จะมีสมาชิกเพิ่มขึ้นอีก 1 ตัว คือ ['Python', 'Dart', 'GoLang']

4.11 การ pop สมาชิกออกจากตัวแปรลิสต์

ฟังก์ชัน pop() ทำหน้าที่ดึงสมาชิกตำแหน่งสุดท้ายภายในลิสต์ออกไป

ตัวอย่างที่ 4.13 การ pop สมาชิกออกจากตัวแปรลิสต์

```
x = ["Python", "Dart"]
z = x.pop()
print(z, x)
```

ผลลัพธ์ :

```
Dart ['Python']
```

ตัวอย่าง 4.13 ตัวแปร x มีสมาชิก 2 ตัว คือ ["Python", "Dart"] ต่อมาทำการ pop() ออกมาและนำค่าไปใส่ตัวแปร z ทำให้ตัวแปร x เหลือสมาชิกเพียง 1 ตัว โดยค่า z มีค่า Dart และตัวแปร x มีค่า ['Python']

4.12 คำสั่ง index() ใช้ตรวจสอบหมายเลขสมาชิกภายในตัวแปรลิสต์

ฟังก์ชัน index() ทำหน้าที่บอกตำแหน่งของสมาชิกภายในตัวแปรลิสต์

ตัวอย่างที่ 4.14 การ pop สมาชิกออกจากตัวแปรลิสต์

```
x = ["Python", "Dart", "GoLang"]
print(x.index("Python"))
```

ผลลัพธ์ :

0

ตัวอย่าง 4.14 ตัวแปร x มีสมาชิก 3 ตัว คือ ["Python", "Dart", "GoLang"] การตรวจสอบว่าสมาชิกที่มีค่า "Python" มีลำดับสมาชิกที่เท่าใด ใช้คำสั่ง x.index("Python") ผลลัพธ์แสดงตำแหน่งที่ 0

4.13 คำสั่ง count() สำหรับตรวจสอบจำนวนข้อมูลที่ตรงกับค่าที่กำหนด

คำสั่ง count() ทำหน้าที่บอกจำนวนข้อมูลที่พบภายในตัวแปรลิสต์

ตัวอย่างที่ 4.15 การ count() สำหรับตรวจสอบจำนวนข้อมูลที่ตรงกับค่าที่กำหนด

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]
print(a.count(3))
```

ผลลัพธ์ :

3

ตัวอย่าง 4.13 ตัวแปร a มีสมาชิก 10 ตัว คือ [3, 4, 2, 3, 4, 9, 4, 7, 3, 4] ต้องการนับว่าสมาชิกที่มีค่าเท่ากับเลข 3 มีกี่ตัว ใช้คำสั่ง a.count(3) ผลลัพธ์มีค่าเท่ากับ 3 ตัว

4.14 คำสั่ง sort() สำหรับเรียงลำดับรายการในลิสต์จากน้อยไปมาก

คำสั่ง sort() ใช้สำหรับเรียงลำดับข้อมูลภายในตัวแปรรายการจากน้อยไปมาก ดังนี้

ตัวอย่างที่ 4.16 คำสั่ง sort() ใช้เรียงลำดับข้อมูลภายในตัวแปรลิสต์จากน้อยไปมาก

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]
a.sort()
print(a)
```

ผลลัพธ์ :

[2, 3, 3, 3, 4, 4, 4, 4, 7, 9]

ตัวอย่าง 4.14 คำสั่ง sort() เป็นการเรียงลำดับข้อมูลภายในตัวแปรลิสต์จากน้อยไปมาก ผลลัพธ์ของคำสั่ง a.sort() มีค่าเท่ากับ [2, 3, 3, 3, 4, 4, 4, 4, 7, 9]

4.15 คำสั่ง reverse() ใช้กลับลำดับสมาชิกภายในตัวแปรลิสต์

คำสั่ง reverse() ใช้กลับลำดับสมาชิกภายในตัวแปรลิสต์ ถ้าใช้คำสั่ง reverse() สองรอบลำดับของสมาชิกกลับมาเหมือนเดิมเหมือนยังไม่ได้กลับลำดับ ดังนี้

ตัวอย่างที่ 4.17 คำสั่ง reverse() ใช้กลับลำดับสมาชิกภายในตัวแปรลิสต์

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]
a.reverse()
```

```
print(a)
a.reverse()
print(a)
```

ผลลัพธ์ :

```
[4, 3, 7, 4, 9, 4, 3, 2, 4, 3]
[3, 4, 2, 3, 4, 9, 4, 7, 3, 4]
```

ตัวอย่าง 4.17 คำสั่ง reverse() เป็นการก่ากลับลำดับสมาชิกภายในตัวแปรลิสต์จากหลังมาหน้า และจากหน้าไปหลัง

4.16 คำสั่ง set() ใช้กำจัดสมาชิกที่ซ้ำกันภายในลิสต์

คำสั่ง set() เป็นการกำจัดสมาชิกที่ซ้ำกันออกจากตัวแปรลิสต์

ตัวอย่างที่ 4.18 คำสั่ง set() ใช้กำจัดสมาชิกที่ซ้ำกันออกจากตัวแปรลิสต์

```
a = [ 3, 4, 2, 3, 4, 9, 4, 7, 3, 4 ]
b = list(set(a))
print(b)
```

ผลลัพธ์ :

```
[2, 3, 4, 7, 9]
```

ตัวอย่าง 4.18 คำสั่ง set() ใช้สำหรับกำจัดสมาชิกที่ซ้ำกันภายในตัวแปรลิสต์ ในตัวอย่างนี้ใช้คำสั่ง b = list(set(a)) หมายถึงเมื่อกำจัดสมาชิกที่ซ้ำกันออกไปแล้วแปลงตัวแปรเซตให้กลับมาเป็นตัวแปรลิสต์อีกครั้ง จากนั้นพิมพ์ตัวแปร b ผลลัพธ์มีค่าเท่ากับ [2, 3, 4, 7, 9]

4.17 ตัวแปรทูเพิล (Tuple)

- ตัวแปรทูเพิลประกาศโดยใช้เครื่องหมาย () ล้อมรอบตัวแปร เช่น m = (3,2,1) ในขณะที่ลิสต์ประกาศโดยใช้เครื่องหมาย [] เช่น m = [3,2,1] เป็นต้น
- ตัวแปรลิสต์เป็น mutable คือสามารถแก้ไขข้อมูลภายในสมาชิกเครื่องหมาย [] ได้ แต่ตัวแปร Tuple เป็น immutable คือไม่อนุญาตให้แก้ไขข้อมูลภายในสมาชิก
- ตัวแปรทูเพิลไม่มีฟังก์ชันสนับสนุนเหมือนตัวแปรลิสต์ เนื่องจากความเป็น immutable ดังนั้นจึงไม่สามารถแก้ไข ไม่สามารถ sort(), หรือ reverse() ข้อมูลภายในทูเพิลได้
- ตัวแปรทูเพิลมีการเข้าถึงตำแหน่งภายในสมาชิกย่อยเหมือนตัวแปรลิสต์และมีตำแหน่งการอ้างอิงสมาชิกเหมือนแปรลิสต์สามารถระบุตำแหน่งของสมาชิกด้วยเลขบวก และเลขลบได้

ตัวอย่างที่ 4.19 การประกาศตัวแปรชนิดทูเพิล

```
a = ( 3, 2, 1 )
print(a)
```

ผลลัพธ์ :

3, 2, 1

ตัวอย่าง 4.19 การใช้งานตัวแปรทูปเพิลเหมือนตัวแปรลิสต์ต่างกันที่ไม่สามารถแก้ไขค่าข้อมูลของสมาชิกภายในทูปเพิลได้

4.18 ฟังก์ชันเกี่ยวกับเวลา

การจัดการเกี่ยวกับเวลาภาษาไพธอนใช้โมดูล time ประกอบด้วยคำสั่ง ต่อไปนี้

ตารางที่ 4.1 คำสั่งเกี่ยวกับเวลาภายในโมดูล time

ฟังก์ชัน	คำอธิบาย
import time	ขอใช้โมดูล time
time.time()	แสดงตัวเลขไทม์สแตมป์ เช่น 1634703415.28907
time.ctime()	แสดงวันและเวลา เช่น Wed Oct 20 11:18:50 2021
time.tzname	แสดงโซนเวลา เช่น SE Asia Standard Time
time.sleep(1)	หน่วงเวลา เช่น หน่วงเวลา 1 วินาที
time.localtime()	แสดงเวลาเป็นโครงสร้าง เช่น (tm_year=2021, tm_mon=10, tm_mday=20, tm_hour=11, tm_min=21, tm_sec=24, tm_wday=2, tm_yday=293, tm_isdst=0)

4.19 ฟังก์ชันการสุ่มตัวเลข

การสุ่มเลขในภาษาไพธอนใช้โมดูล random ประกอบด้วยคำสั่ง ดังต่อไปนี้

ตารางที่ 4.2 ฟังก์ชันการสุ่มภายในโมดูล random

ฟังก์ชัน	คำอธิบาย
import random	ขอใช้โมดูล random
random.randint(1,10)	สุ่มเลขจำนวนเต็มในช่วงที่กำหนด เช่น 1 ถึง 10
random.random()	สุ่มเลขทศนิยมในช่วง 0 ถึง 1
random.shuffle([1,3,5,6])	สุ่มค่าภายในตัวแปรลิสต์
random.uniform(10, 20)	สุ่มค่าทศนิยมในช่วงที่กำหนด เช่น 10 ถึง 20
random.weibullvariate(1,1.5)	สุ่มด้วยการกระจายตัวเลขแบบ Weibull distribution

4.20 ฟังก์ชันคณิตศาสตร์

ภาษาไพธอนมีคำสั่งเกี่ยวกับการคำนวณทางคณิตศาสตร์ในโมดูล math

ตารางที่ 4.3 ฟังก์ชันคณิตศาสตร์และค่าคงที่ภายในโมดูล math

ฟังก์ชัน	คำอธิบาย
import math	ขอใช้โมดูล math

ฟังก์ชัน	คำอธิบาย
<code>math.exp(1)</code>	คำนวณค่า e ยกกำลัง เมื่อ e คือ ค่าคงที่ มีค่า 2.71
<code>math.pi</code>	ค่า pi มีค่า 3.1415
<code>math.sin()</code> และ <code>math.asin(x)</code>	คำนวณค่า sin และ arcsin
<code>math.cos(x)</code> และ <code>math.acos(x)</code>	คำนวณค่า cos และ arccos
<code>math.tan()</code> และ <code>math.atan(x)</code>	คำนวณค่า tan และ arctan
<code>math.pow(2,8)</code>	คำนวณเลขยกกำลัง เช่น 2^8 มีค่าเท่ากับค่าสั่ง $2^{**}8$
<code>math.log</code> <code>log10</code> <code>log1p</code> <code>log2</code>	คำนวณค่า logarithm
<code>math.sqrt(9)</code>	คำนวณการถอดสแควร์ท เช่น <code>sqrt(9)</code> มีค่าเท่ากับ 3
<code>math.degrees(x)</code>	แปลง x เป็นหน่วยองศา (degree)
<code>math.radians(x)</code>	แปลง x เป็นหน่วยเรเดียน (radian)

แบบฝึกหัดท้ายบท

1. จงอธิบายการประกาศตัวแปรลิสต์และทูเพิล
2. จงอธิบายการเข้าถึงตำแหน่งภายในตัวแปรลิสต์และทูเพิล
3. จงอธิบายการเพิ่มและลบตัวแปรลิสต์
4. จงอธิบายการเรียง สลับตำแหน่งหลังไปหน้า
5. จงอธิบายการจัดสมาชิกที่ซ้ำกันในลิสต์
6. จงเขียนโปรแกรมรับตัวเลขจากแป้นพิมพ์จำนวน 10 รอบ จากนั้นคำนวณค่าผลรวมและค่าเฉลี่ยของตัวเลขทั้ง 10 ชุด

บทที่ 5

การใช้งานสตริง

แนวคิด

บริษัทกูเกิลเป็นบริษัทยักษ์ใหญ่ด้านการให้บริการการประมวลผลข้อมูล โดยมีบริษัทแม่ชื่อ Alphabet มีความหมายว่า "ตัวอักษร" ย่อมาหมายความว่ากูเกิลให้ความสำคัญต่อตัวอักษรหรือข้อมูลที่อยู่ในรูปตัวอักษร

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ การประกาศตัวแปรสตริง สมาชิกย่อยภายในสตริง การแยกสตริงด้วยคำสั่ง split การตรวจสอบความยาวสตริงด้วย len() การเข้าถึงข้อมูลบางส่วนในสตริง การใช้คำสั่ง in ตรวจสอบข้อความภายในสตริง เครื่องหมาย Escape Sequence คำสั่งภายในโมดูลสตริง การจัดการแฟ้มข้อมูล การอ่านและเขียนแฟ้มข้อมูล (Read and Write File) แฟ้มข้อมูลแบบ Binary และ Text Mode คำสั่ง glob สำหรับอ่านรายชื่อไฟล์และไดเรกทอรี การลบไฟล์ การอ่านไฟล์จากเว็บไซต์ด้วย urlopen การใช้งาน Regular Expression Meta-Characters ภายใน Regular Expression และการค้นหาข้อความที่มีรูปแบบ

วัตถุประสงค์

1. อธิบายการแยกสตริงออกจากกันด้วยคำสั่ง split ได้
2. อธิบายการใช้เครื่องหมาย Escape Sequence ได้
3. อธิบายการอ่านเขียนและแสดงรายชื่อไฟล์ได้
4. อธิบายการอ่านไฟล์จากเว็บไซต์ด้วย urlopen ได้
5. อธิบายการตรวจจรูปแบบข้อความด้วย Regular Expression ได้

5.1 การประกาศตัวแปรสตริง

ตัวแปรสตริงใช้สำหรับเก็บข้อความ สามารถประกาศด้วยเครื่องหมาย single , double และ triple quote ได้ดังนี้

ตารางที่ 5.1 การประกาศตัวแปรสตริงด้วยเครื่องหมาย single double และ triple quote

เครื่องหมาย	ชื่อเรียก	ตัวอย่าง
'	Single quote	'นี่คือข้อความที่เปิดปิดด้วยเครื่องหมาย single quote'
"	Double quote	"นี่คือข้อความที่เปิดปิดด้วยเครื่องหมาย double quote"
'''	triple single quote	'''นี่คือข้อความที่เปิดปิดด้วย triple single quote'''
"""	Triple double quote	"""นี่คือข้อความที่เปิดปิดด้วย triple double quote"""

ข้อดีของการใช้งาน Triple Quote คือ สามารถสร้างตัวแปรสตริงซึ่งมีค่า Single หรือ Double Quote ภายในสตริงได้และสามารถประกาศตัวแปรสตริงหลาย ๆ บรรทัดได้

5.2 การบวกและการคูณสตริง

ตัวแปรสตริงบวกกับสตริง คือการนำสตริงมาต่อกัน ในทางกลับกันตัวแปรสตริง คูณกับตัวเลข เป็นการนำสตริงมาต่อกันเท่ากับจำนวนที่คูณ ดังตัวอย่างต่อไปนี้

ตารางที่ 5.2 การบวกและการคูณสตริง

ชนิดข้อมูล	ตัวอย่าง	ผลลัพธ์
ตัวเลข	1+1	2
สตริง	"1" + "1"	"11"
ตัวเลข	5 * 10	50
สตริง	"5" * 10	"5555555555"

การบวกและการคูณตัวเลขและการคูณสตริงให้ผลลัพธ์แตกต่างกัน ถ้าข้อมูลเป็นตัวเลขกับตัวเลข ผลลัพธ์จะเป็นตัวเลข ส่วนข้อมูลตัวเลขและสตริง ผลลัพธ์จะเป็นสตริง จากตัวอย่างนี้ '5' * 10 จะหมายถึง การต่อสตริง '5' เข้าด้วยกัน 10 ครั้ง การบวกสตริงคือการต่อสตริงเข้าด้วยกัน เช่น '1' + '1' มีค่าเท่ากับ '11'

5.3 ตำแหน่งสมาชิกภายในสตริง

การเข้าถึงตำแหน่งสมาชิกภายในตัวแปรสตริงระบุได้สองวิธี เช่นเดียวกับตัวแปรชนิดอื่น ๆ คือ ระบุด้วยเลขจำนวนเต็มบวก และจำนวนเต็มลบ

ตัวอย่างที่ 5.1 การระบุตำแหน่งของสมาชิกย่อยภายในลิสต์

ตัวแปร s =	P	r	o	g	r	a	m	m	e	r
ระบุตำแหน่งด้วยเลขบวก	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
ระบุตำแหน่งด้วยเลขลบ	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]

สมาชิกตัวแรกสุดแทนคือตำแหน่ง s[0] มีค่า "P" ส่วนสมาชิกตำแหน่งหลังสุดคือ s[-1] มีค่า "r"

5.4 การตรวจสอบความยาวของสตริง

ภาษาไพธอนใช้คำสั่ง len() ในการตรวจสอบจำนวนสมาชิกของตัวแปร การตรวจสอบความยาวของตัวแปรสตริงใช้คำสั่ง len() เช่นเดียวกัน แสดงตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 5.2 การตรวจสอบความยาวสตริง

```
s = "Information Science"
print(len(s))
```

ผลลัพธ์ :

19

ตัวอย่าง 5.2 ตัวแปร s มีชนิดสตริง เก็บข้อความคำว่า "Information Science" การตรวจสอบความยาวของสตริงใช้คำสั่ง len(s) ซึ่งมีจำนวน 19 ตัวอักษร

5.5 การวนเข้าไปในสตริงเพื่อเข้าถึงสมาชิกย่อย

การวนเข้าไปในสตริงเป็นการอ่านอักขระภายในสตริงทีละตัว โดยบอกตำแหน่งสมาชิกภายในเครื่องหมาย [] ดังนี้

ตัวอย่างที่ 5.3 การวนเข้าไปในสตริงเพื่อเข้าถึงสมาชิกย่อย

```
s = "ABC"
for i in range(len(s)):
    print(s[i])
```

ผลลัพธ์ :

```
A
B
C
```

ตัวอย่างที่ 5.3 แสดงการเข้าถึงสมาชิกภายในตัวแปรสตริงทีละตัวอักขระ โดยสร้างตัวเลขความยาวของสตริงด้วยคำสั่ง `range(len(s))` ซึ่งจะมีค่าเท่ากับ `[0, 1, 2]` ในการวนเข้าไปในสตริงจะใช้ลำดับที่ `i` ในการเข้าถึงตำแหน่งสมาชิกภายในสตริง เช่น `s[0]` คือ "A" `s[1]` คือ "B" และ `s[2]` คือ "C"

5.6 การเข้าถึงข้อมูลบางส่วนในสตริง

การเข้าถึงข้อมูลบางส่วนภายในสตริงระบุภายในเครื่องหมาย [] โดยกำหนดค่าเริ่มต้นและค้นด้วย : และค่าลำดับปลายทางที่ต้องการเข้าถึง โดยสามารถระบุการเข้าถึงด้วยเลขบวกและลบได้ เช่น

ตัวอย่างที่ 5.4 การวนเข้าไปในสตริงเพื่อเข้าถึงสมาชิกย่อย

```
s = 'python'
print(s[:])
print(s[2:])
print(s[2:4])
print(s[:4])
print(s[-4:-2])
```

ผลลัพธ์ :

```
Python
thon
th
pyth
th
```

ตัวอย่างที่ 5.4 คำสั่ง `s[:]` คือ การเอาสมาชิกทุกตัวออกมา ดังนั้น ผลลัพธ์จะแสดงคำว่า "python" `s[2:]` คือ เอาสมาชิกตัวที่ 2 จนถึงตัวสุดท้ายออกมา ผลลัพธ์ คือ "thon" `s[2:4]` คือ เอาสมาชิกตัวที่ 2 ถึง

4-1 ออกมา เนื่องจากหมายเลขปลายทางจะต้องลบด้วย 1 เสมอ ผลลัพธ์ คือ "th" s[:4] คือ เอาสมาชิกที่ 0 ถึง 4-1 ผลลัพธ์ คือ "pyth" และการระบุด้วยเลขลบ s[-4:-2] คือ ลำดับที่ -4 ถึง -2 ผลลัพธ์คือ "th"

5.7 ตัวแปรสตริงเป็น Immutable คือแก้ไขข้อมูลโดยตรงไม่ได้

ไพธอนไม่อนุญาตให้แก้ไขข้อมูลส่วนหนึ่งส่วนใดภายในสตริง (Immutable) พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.5 ข้อผิดพลาดที่เกิดจากการพยายามแก้ไขข้อมูลภายในสตริง

```
s= 'python'
s[0] = "H"
```

ผลลัพธ์ :

```
Traceback (most recent call last):
TypeError: object doesn't support item assignment
```

ตัวอย่างที่ 5.5 การพยายามแก้ไขข้อมูลภายในสตริงทำให้เกิดความผิดพลาด ไพธอนรายงานผลว่า อ็อบเจ็กต์ไม่สนับสนุนการ assign การแก้ไขบางส่วนในตัวแปรสตริงใช้การคัดลอกข้อมูลออกมาเก็บในตัวแปรใหม่ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.6 การแก้ไขบางส่วนในตัวแปรสตริง

```
s= 'python'
s1 = "N" + s[1:]
print(s1)
```

ผลลัพธ์ :

```
Nython
```

ตัวอย่างที่ 5.6 ตัวแปร s1 เป็นคนละตัวแปร s ดังนั้น การแก้ไข s1 จะไม่มีผลต่อตัวแปร s

5.8 การแทนที่คำภายในตัวแปรสตริงด้วยคำสั่ง replace()

คำสั่ง replace ใช้แทนที่คำที่ต้องการด้วยคำใหม่ แสดงตัวอย่างการใช้งานดังนี้

ตัวอย่างที่ 5.7 การแทนที่คำที่ต้องการด้วยคำใหม่ในตัวแปรสตริง

```
print("information science".replace("information","data"))
```

ผลลัพธ์ :

```
data science
```

ตัวอย่างที่ 5.7 ตัวแปร s มีชนิดสตริง เก็บข้อความคำว่า "information science" ต้องการเปลี่ยนคำว่า information เป็นคำว่า data ใช้คำสั่ง `replace("information","data")` ผลลัพธ์เป็นคำว่า " data science"

5.9 การใช้คำสั่ง in เพื่อตรวจสอบข้อความภายในสตริง

ไพธอนมีคำสั่ง in เพื่อตรวจสอบข้อมูลภายในสตริง คืนค่าเป็น True หากมีข้อความในตรึงและคืนค่า False หากไม่มีข้อความในสตริง แสดงตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 5.8 การตรวจสอบความยาวสตริง

```
s = "Information Science"
print("information" in s)
print("Information" in s)
```

ผลลัพธ์ :

```
False
True
```

ตัวอย่างที่ 5.8 ตัวแปร s มีชนิดสตริง เก็บข้อความคำว่า "Information Science" การตรวจสอบว่า "information" อยู่ภายในตัวแปร s หรือไม่ ใช้คำสั่ง in เพื่อตรวจสอบ เช่น "information" in s คำตอบเป็น False เพราะตัวตัวพิมพ์ใหญ่และตัวพิมพ์เล็กเป็นคนละตัวกัน (case sensitive) คำสั่ง "Information" in s คำตอบเป็น True เพราะมีคำว่า "Information" ในตัวแปร s

5.10 การเติมเลขศูนย์หน้าตัวเลขด้วยคำสั่ง zfill()

คำสั่ง `zfill()` เป็นการเติมเลขศูนย์หน้าตัวเลข แสดงตัวอย่างการใช้งานดังนี้

ตัวอย่างที่ 5.9 การเติมเลขศูนย์หน้าตัวเลข

```
print("5".zfill(4))
print(str(5).zfill(8))
```

ผลลัพธ์ :

```
0005
00000005
```

ตัวอย่างที่ 5.9 คำสั่ง `"5".zfill(4)` คือ ข้อความ "5" เรียกคำสั่ง `zfill(4)` คือ ให้เติมเลขศูนย์ให้เต็ม 4 หลัก ดังนั้น ผลลัพธ์ คือ "0005" คำสั่ง `str(5).zfill(8)` เป็นการแปลงตัวเลขให้เป็นสตริงด้วยคำสั่ง `str(5)` จากนั้นเรียกใช้คำสั่ง `zfill(8)` เป็นการเติมเลขศูนย์ให้ครบ 8 หลัก ดังนั้น ผลลัพธ์ คือ "00000005"

5.11 การลบเครื่องหมาย white space ด้านหน้าและหลังตัวแปรสตริงด้วยคำสั่ง strip()

คำสั่ง strip() เป็นกำจัด white space เช่น เครื่องหมาย space bar ที่ด้านและหลังสตริงออกไป โดยมี 2 คำสั่งเพิ่มเติม คือ rstrip และ lstrip เป็นการกำจัด white space ด้านขวาและซ้ายของข้อความ แสดงตัวอย่างการใช้งานดังนี้

ตัวอย่างที่ 5.10 การกำจัด white space ด้านหน้าหลังตัวแปรสตริง

```
a = " 000 ".strip()
b = " 000 ".lstrip()
c = " 000 ".rstrip()
print("%s\n%s\n%s"%(a,b,c))
```

ผลลัพธ์ :

```
'000'
'000 '
' 000'
```

ตัวอย่างที่ 5.10 คำสั่ง strip() เป็นการลบ white space ด้านหน้าและหลังสตริง ตัวแปร a = " 000 " มีความยาว 5 ตัวอักษร เพราะมีเครื่องหมาย space bar ด้านหน้าและด้านหลัง คำสั่ง strip() จะลบ white space ด้านหน้าและหลังทำให้ตัวแปร มีความยาว 3 ตัวอักษร คือ "001"

คำสั่ง rstrip() และ lstrip() เป็นการลบ white space ด้านขวาและซ้ายตามลำดับ ทำให้ผลลัพธ์ คำสั่ง lstrip() มีผลลัพธ์ '000 ' และคำสั่ง rstrip() เป็นการลบ white space ด้านขวาของข้อความ มีผลลัพธ์ คือ ' 000'

5.12 การแยกสตริงด้วยคำสั่ง split()

คำสั่ง split() เป็นการแยกข้อความออกเป็นส่วน ๆ ด้วยคำแยก ผลลัพธ์เป็นตัวแปรลิสต์ คำสั่งที่เกี่ยวข้อง ได้แก่ rsplit() และ linesplit() พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.11 การแยกตัวแปรสตริงออกจากกันด้วยคำสั่ง split

```
s = "Google|Gmail|Youtube"
z = s.split("|")
print(z)
```

ผลลัพธ์ :

```
['Google', 'Gmail', 'Youtube']
```

ตัวอย่างที่ 5.11 คำสั่ง split() จะแยกสตริงออกจากกันด้วยคำแยก ในตัวอย่างนี้คือเครื่องหมาย "|" ผลลัพธ์เก็บอยู่ในตัวแปร z คือ ['Google', 'Gmail', 'Youtube']

ตัวอย่างที่ 5.12 การแยกตัวแปรสตริงออกจากกันด้วยคำสั่ง `rsplit`

```
s = "Google|Gmail|Youtube"
z = s.rsplit("|",1)
print(z)
```

ผลลัพธ์ :

```
['Google|Gmail', 'Youtube']
```

ตัวอย่างที่ 5.12 คำสั่ง `rsplit()` จะแยกสตริงเริ่มจากด้านขวาของคำแยก ถ้าไม่กำหนดค่า "max" จะคืนค่าผลลัพธ์เหมือนคำสั่ง `split()` ในตัวอย่างใช้คำแยกด้วยเครื่องหมาย "|" และกำหนด max เท่ากับ 1 ผลลัพธ์ จะมีสมาชิก 2 ตัว คือ 'Google|Gmail' และ 'Youtube'

ตัวอย่างที่ 5.13 การแยกตัวแปรสตริงออกจากกันที่ละบรรทัดด้วยคำสั่ง `splitlines()`

```
s = """Google
Gmail
Youtube"""
z = s.splitlines()
print(z)
```

ผลลัพธ์ :

```
['Google', 'Gmail', 'Youtube']
```

ตัวอย่างที่ 5.13 คำสั่ง `splitlines()` เป็นการแยกสตริงออกจากกันด้วยเครื่องหมายขึ้นต้นบรรทัดใหม่ ผลลัพธ์มีสมาชิก 3 ตัว คือ ['Google', 'Gmail', 'Youtube']

5.13 การรวมสมาชิกในตัวแปรลิสต์เป็นข้อความด้วยคำสั่ง `join()`

คำสั่ง `join()` ใช้รวมสมาชิกในตัวแปรลิสต์เป็นข้อความ เมื่อข้อความของสตริงจะเป็นตัวเชื่อมระหว่างข้อมูลภายในตัวแปรลิสต์ พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.14 การรวมสมาชิกในตัวแปรลิสต์เป็นข้อความด้วยคำสั่ง `join()`

```
z = ['Google', 'Gmail', 'Youtube']
s = "--".join(z)
print(s)
s = ", ".join(z)
print(s)
```

ผลลัพธ์ :

```
Google--Gmail--Youtube
Google, Gmail, Youtube
```

ตัวอย่างที่ 5.14 การเชื่อมสมาชิกในตัวแปรลิสต์เข้าด้วยกันด้วยคำสั่ง join จะใช้ค่าในตัวแปรสตริงเป็นตัวเชื่อม ในตัวอย่างกำหนดตัวแปร z เป็นตัวแปรรายการมีสมาชิกคือ ['Google', 'Gmail', 'Youtube'] เมื่อคำสั่ง `"".join(z)` เป็นการเชื่อมสมาชิกในตัวแปรลิสต์ด้วยเครื่องหมาย "--" ผลลัพธ์ คือ Google--Gmail--Youtube คำสั่ง `", ".join(z)` เป็นการเชื่อมสมาชิกในตัวแปรลิสต์ด้วยเครื่องหมายคอมม่า (,) ผลลัพธ์ คือ "Google, Gmail, Youtube "

5.14 การค้นหาข้อความย่อยภายในข้อความทั้งหมดด้วยคำสั่ง find() และ rfind()

คำสั่ง find() เป็นการค้นหาข้อความย่อยภายในข้อความทั้งหมด ผลลัพธ์จะบอกตำแหน่งเริ่มต้นของคำค้น หากไม่พบคำตอบผลลัพธ์จะคืนค่า -1

คำสั่ง rfind() เป็นการค้นหาคำจากด้านขวาสุดของข้อความ โดยมีรูปแบบไวยากรณ์ คือ `rfind(value, start, end)` เมื่อ value คือ คำค้น และ start คือ ตำแหน่งเริ่มต้นค้น ค่าปริยายกำหนดที่ 0 end คือ ตำแหน่งสุดท้ายของการค้น ค่าปริยายคือความยาวของสตริงที่ค้น

ตัวอย่างที่ 5.15 การค้นหาสตริงภายในสตริงด้วย find()

```
a = 'Information Science'
b = 'Data Science'
print(a.find('on'))
print(b.find('on'))
```

ผลลัพธ์ :

```
9
-1
```

ตัวอย่าง 5.15 ตัวแปร a เก็บคำว่า 'Information Science' ตัวแปร b เก็บคำว่า 'Data Science' ต้องการค้นหามีคำว่า "on" ในตัวแปรทั้งสองใช้คำสั่ง find() และระบุค่าค้น ผลลัพธ์จะคืนค่า 9 คือตำแหน่งเริ่มต้นที่ 9 จากคำว่า "Information**on** Science" ตัวแปร b ไม่มีคำว่า "on" อยู่ภายในจึงคืนค่า -1

ตัวอย่างที่ 5.16 การค้นหาสตริงภายในสตริงด้วย rfind()

```
s = 'Information Science and Data Science'
print(s.rfind('and',10, len(s)))
s = 'Information Science and Data Science'
print(s.rfind('and',23, len(s)))
```

ผลลัพธ์ :

```
20
-1
```

ตัวอย่างที่ 5.16 ตัวแปร s เก็บข้อความ 'Information Science and Data Science' ต้องการค้นหาคำว่า "and" โดยเริ่มค้นจากตำแหน่งที่ 10 ถึงตำแหน่งสุดท้ายของตัวแปร s ด้วยคำสั่ง len(s)

คำตอบพบที่ตำแหน่ง 20 ในขณะที่กำหนดค่าเริ่มต้นการค้นหาที่ 23 เป็นต้นไปจะไม่พบคำว่า "and" จึงคืนค่าเท่ากับ -1

5.15 การค้นหาข้อความภายในข้อความด้วยคำสั่ง index() และ rindex()

คำสั่ง index() ทำหน้าที่เหมือนคำสั่ง find() ต่างกันที่หากไม่พบข้อความจะคืนค่าเป็น ValueError ขณะที่คำสั่ง find() จะคืนค่าเป็น -1

ตัวอย่างที่ 5.17 การค้นหาสตริงภายในสตริงด้วย index()

```
a = 'Information Science'
b = 'Data Science'
print(a.index('on'))
print(b.index('on'))
```

ผลลัพธ์ :

```
9
ValueError: substring not found
```

ตัวอย่างที่ 5.17 ตัวแปร a เก็บข้อความ "Information Science" ตัวแปร b เก็บข้อความ "Data Science" เมื่อค้นหาคำว่า "on" ด้วยคำสั่ง index() ในตัวแปร จะคืนค่าเป็นตำแหน่งที่พบ คือ ตำแหน่งที่ 9 ในกรณีตัวแปร b จะคืนค่าเป็น ValueError

ตัวอย่างที่ 5.18 การค้นหาสตริงภายในสตริงด้วย rindex()

```
s = 'Information Science and Data Science'
print(s.rindex('and',10, len(s)))
s = 'Information Science and Data Science'
print(s.rindex('and',23, len(s)))
```

ผลลัพธ์ :

```
20
ValueError: substring not found
```

ตัวอย่างที่ 5.18 คำสั่ง rindex() ใช้ค้นหาตำแหน่งของคำค้น ในตัวอย่างเป็นการค้นหาคำว่า "and" เริ่มต้นที่ 10 ถึงความยาวสตริง len(s) ผลลัพธ์ 20 คือตำแหน่งเริ่มต้นคำว่า "and" คือตำแหน่งที่ 20 ในขณะที่ถ้าเริ่มต้นที่ตำแหน่ง 23 เป็นต้นไปจะไม่พบคำว่า "and" จึงคืนค่าเป็น ValueError

5.16 การนับจำนวนคำภายในสตริงด้วยคำสั่ง count

ตัวแปรสตริงจะมีฟังก์ชัน count() ในการนับคำภายในสตริง ดังนี้

ตัวอย่างที่ 5.19 การนับจำนวนคำภายในสตริง

```
"information science".count("o")
```

ผลลัพธ์ :

```
2
```

ตัวอย่างที่ 5.19 ตัวแปรสตริงจะมีฟังก์ชัน count() สำหรับนับคำที่ปรากฏอยู่ในสตริง ในตัวอย่างเป็นการนับจำนวนตัว "o" ที่ปรากฏในคำว่า "information science" คำตอบคือ 2 ตัว

5.17 การตรวจสอบข้อความถูกต้องตามหลักการตั้งชื่อตัวแปรและฟังก์ชันด้วย identifier()

คำสั่ง identifier() ใช้สำหรับตรวจสอบว่าข้อความตั้งชื่อถูกหลักการตั้งชื่อตัวแปรหรือชื่อฟังก์ชันหรือชื่อคลาสหรือไม่

สตริงที่เป็น identifier คือประกอบด้วยตัวอักษร (a-z) และตัวเลข (0-9) หรือเครื่องหมาย underscore (_) และต้องไม่เริ่มต้นด้วยตัวเลขและไม่ประกอบด้วย space

ตัวอย่างที่ 5.20 การตรวจสอบข้อความถูกต้องตามหลักการตั้งชื่อด้วย identifier()

```
a = "x"
b = "2x"
c = "_x"
print(a.identifier(), b.identifier(), c.identifier())
```

ผลลัพธ์ :

```
True False True
```

ตัวอย่างที่ 5.20 การตรวจสอบว่าตัวแปรสตริงสามารถนำไปใช้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน ได้หรือไม่ ใช้คำสั่ง identifier() เมื่อตัวแปร a ="x" b="2x" และ c = "_x" ผลลัพธ์คือ "2x" ไม่สามารถนำไปใช้ตั้งชื่อตัวแปรได้เนื่องจากผิดไวยากรณ์เพราะขึ้นต้นด้วยตัวเลข

5.18 การจัดการเกี่ยวกับตัวพิมพ์ใหญ่และเล็ก

คำสั่ง upper() ใช้สำหรับแปลงข้อความให้เป็นตัวพิมพ์ใหญ่

คำสั่ง lower() ใช้สำหรับแปลงข้อความเป็นตัวพิมพ์เล็ก

คำสั่ง swapcase() ใช้สำหรับสลับตัวพิมพ์ใหญ่และเล็ก

คำสั่ง capitalize() ใช้สำหรับแปลงตัวอักษรตัวแรกของข้อความเป็นตัวพิมพ์ใหญ่

ตัวอย่างที่ 5.21 การตรวจสอบข้อความถูกต้องตามหลักการตั้งชื่อด้วย identifier()

```
s = "Information Science"
print(s.upper())
print(s.lower())
print(s.swapcase())
```

```
print(s.capitalize())
```

ผลลัพธ์ :

```
INFORMATION SCIENCE
information science
iNFORMATION sCIENCE
Information science
```

ตัวอย่างที่ 5.21 ตัวแปร s เก็บข้อความ "Information Science" คำสั่ง upper() เป็นการแปลงข้อความให้เป็นตัวพิมพ์ใหญ่ ผลลัพธ์ คือ "INFORMATION SCIENCE" คำสั่ง lower() แปลงข้อความเป็นตัวพิมพ์เล็กทั้งหมด ผลลัพธ์ คือ "information science" และคำสั่ง swapcase() เป็นการสลับตัวตัวพิมพ์ ผลลัพธ์ คือ "iNFORMATION sCIENCE" และคำสั่ง capitalize() คือการแปลงตัวอักษรตัวแรกเป็นตัวพิมพ์ใหญ่ ผลลัพธ์ คือ "Information science"

5.19 การเปลี่ยนอักษรด้านหน้าคำเป็นตัวพิมพ์ใหญ่ด้วยฟังก์ชัน capwords

การแปลงตัวอักษรตัวแรกเป็นตัวพิมพ์ใหญ่ ด้วยคำสั่ง capwords ดังนี้

ตัวอย่างที่ 5.22 การแสดงค่าตัวอักษร ASCII ภายใน ascii_letters

```
import string
string.capwords("information science")
```

ผลลัพธ์ :

```
Information Science
```

ตัวอย่างที่ 5.22 การเปลี่ยนอักษรด้านหน้าคำเป็นตัวพิมพ์ใหญ่ใช้ฟังก์ชัน capwords ภายในโมดูล string ผลลัพธ์ ของคำสั่ง string.capwords("information science") จะเปลี่ยนตัวแรกของคำเป็นตัวพิมพ์ใหญ่ ผลลัพธ์ คือ "Information Science"

5.20 การกำหนดตำแหน่งการแสดงผลข้อความ

- » คำสั่ง center() เป็นการจัดตำแหน่งข้อความให้อยู่กึ่งกลาง
- » คำสั่ง expandtabs() สำหรับกำหนดขนาดแท็บ ค่าปริยาย คือ 8 ตัวอักษร
- » คำสั่ง rjust(length, character) กำหนดข้อความชิดขวา
- » คำสั่ง ljust(length, character) กำหนดข้อความชิดซ้าย

ตัวอย่างที่ 5.23 การแสดงค่าตัวอักษร ASCII ภายใน ascii_letters

```
a = "information science".center(80)
b = "\tinformation science"
c = "\tinformation science".expandtabs(20)
d = "information science".rjust(80, ".")
```

```
e = "information science".ljust(80, ".")
print("%s\n%s\n%s\n%s\n%s"%(a,b,c,d,e))
```

ผลลัพธ์ :

```
information science
information science
information science
.....information science
information science.....
```

ตัวอย่างที่ 5.23 ตัวแปร a กำหนดให้แสดงกึ่งกลางของระยะ 80 ตัวอักษร ตัวแปร b เป็นการพิมพ์ระยะแท้ไปด้วยระยะปกติ ตัวแปร c กำหนดให้ระยะแท้มีความกว้าง 20 ตัวอักษร ตัวแปร d ให้ข้อความชิดด้านขวาของระยะอักษร 80 ตัว ใส่เครื่องหมายจุดให้ครบ 80 ตัวอักษร และตัวแปร e ให้ข้อความชิดซ้ายระยะตัวอักษร 80 ตัวโดยใส่เครื่องหมายจุดให้ครบ 80 ตัว

5.21 การตรวจสอบค่าลงท้ายของสตริง

คำสั่ง startswith() เป็นการตรวจสอบค่าขึ้นต้นสตริง

คำสั่ง endswith() เป็นการตรวจสอบค่าลงท้ายของสตริง

ตัวอย่างที่ 5.24 การตรวจสอบค่าลงท้าย

```
s = "Information Science."
print( s.startswith("Info"), s.endswith("."))
```

ผลลัพธ์ :

```
True True
```

ตัวอย่างที่ 5.24 ตัวแปร s เก็บคำว่า "Information Science." คำสั่ง startswith("Info") เป็นการตรวจสอบว่าเริ่มต้นด้วย "Info" หรือไม่ คำตอบคือ True คำสั่ง endswith(".") เป็นการตรวจสอบว่าลงท้ายด้วยเครื่องหมาย "." หรือไม่ คำตอบคือ True คือลงท้ายด้วยจุด

5.22 การเข้ารหัสข้อความด้วยคำสั่ง encode()

คำสั่ง encode() เป็นการเข้ารหัสข้อความ ถ้าไม่มีการกำหนดจะหมายถึง UTF-8

ไวยากรณ์

```
string.encode(encoding=encoding, errors=errors)
```

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
encoding	รหัสที่ต้องการเข้ารหัส ค่าปริยายคือ UTF-8

พารามิเตอร์	คำอธิบาย	
error	ความผิดพลาดเมื่อพบข้อความที่เข้ารหัสไม่ได้	
	'backslashreplace'	แทนที่ด้วยเครื่องหมาย backslash
	'ignore'	ไม่ต้องสนใจตัวอักษรที่เข้ารหัสไม่ได้
	'namereplace'	แทนที่ด้วยอักขรด้วยคำที่ระบุไว้
	'strict'	สร้างเหตุการณ์ความผิดพลาด
	'replace'	แทนที่ด้วยเครื่องหมายคำถาม
	'xmlcharrefreplace'	แทนที่ด้วยตัวอักษร xml

ตัวอย่างที่ 5.25 การแสดงค่าตัวอักษร ASCII ภายใน `ascii_letters`

```
s = "Informåtion Science"
print(s.encode(encoding="ascii",errors="backslashreplace"))
print(s.encode(encoding="ascii",errors="ignore"))
print(s.encode(encoding="ascii",errors="namereplace"))
print(s.encode(encoding="ascii",errors="replace"))
print(s.encode(encoding="ascii",errors="xmlcharrefreplace"))
print("%s\n%s\n%s\n%s\n%s\n%s"%(a,b,c,d,e))
```

ผลลัพธ์ :

```
b'Inform\\xe5tion Science'
b'Informtion Science'
b'Inform\\N{LATIN SMALL LETTER A WITH RING ABOVE}tion Science'
b'Inform?tion Science'
b'Inform&#229;tion Science'
```

ตัวอย่างที่ 5.25 ประกาศตัวแปร `s` มีค่า `"Informåtion Science"` มีอักขระพิเศษภายในสตริง คือ `"å"` ต้องการเข้ารหัสข้อความเป็นรหัสแอสกี (ascii) โดยกำหนดความผิดพลาดและการซ่อมแซมความผิดพลาดของตัวอักษรดังนี้ `"backslashreplace"` คือ เมื่อพบความผิดพลาดให้ใส่เครื่องหมาย backslash ผลลัพธ์ คือ `b'Inform\\xe5tion Science'` เมื่อกำหนดความผิดพลาดเป็น `errors="ignore"` จะไม่สนใจอักขระพิเศษนั้น เมื่อกำหนด `errors="namereplace"` จะแทนที่สัญลักษณ์ที่ทำให้เกิดความผิดพลาดด้วย `namereplace` ในตัวอย่างนี้คือ `"\\N{LATIN SMALL LETTER A WITH RING ABOVE}"` เมื่อกำหนด `errors="replace"` จะแทนด้วยเครื่องหมายคำถาม ผลลัพธ์ คือ `b'Inform?tion Science'` และเมื่อกำหนด `errors="xmlcharrefreplace"` ผลลัพธ์จะแทนสัญลักษณ์พิเศษด้วยสัญลักษณ์ XML ผลลัพธ์ คือ `b'Informåtion Science'`

5.23 การค้นหาและแยกข้อความเป็นส่วนสามส่วนด้วยคำสั่ง partition() และ rpartition()

คำสั่ง partition() เป็นการแบ่งข้อความด้วยคำค้นออกเป็น 3 ส่วน คือ ส่วนที่อยู่ด้านหน้าคำค้น คำค้น และส่วนด้านหลังคำค้น คำสั่ง partition() จะแยกข้อความด้วยคำค้นที่พบด้านซ้ายมือ ส่วนคำสั่ง rpartition() แยกข้อความด้วยคำค้นที่พบด้านขวามือ

ตัวอย่างที่ 5.26 การค้นหาและแยกข้อความเป็นส่วนสามส่วนด้วยคำสั่ง partition()

```
s = "ลิงกินกล้วยทุกวัน กล้วยเป็นสิ่งที่ลิงชอบกิน"
z = s.partition("กล้วย")
print(z)
z = s.rpartition("กล้วย")
print(z)
```

ผลลัพธ์ :

```
('ลิงกิน', 'กล้วย', 'ทุกวัน กล้วยเป็นสิ่งที่ลิงชอบกิน')
('ลิงกินกล้วยทุกวัน ', 'กล้วย', 'เป็นสิ่งที่ลิงชอบกิน')
```

ตัวอย่างที่ 5.26 คำสั่ง partition() และ rpartition() จะแบ่งสตริงเป็น 3 ส่วน จากตัวอย่างแบ่งด้วยคำว่า "กล้วย" จากซ้ายมือสุด ผลลัพธ์ ('ลิงกิน', 'กล้วย', 'ทุกวัน กล้วยเป็นสิ่งที่ลิงชอบกิน') และคำว่า "กล้วย" จากขวามือสุด ผลลัพธ์ คือ ('ลิงกินกล้วยทุกวัน ', 'กล้วย', 'เป็นสิ่งที่ลิงชอบกิน')

5.24 การแทนที่คำด้วย maketrans() และ translate()

คำสั่ง translate สร้างตารางการแทนค่าคำ

คำสั่ง maketrans() คืนค่าตัวแปรดิกชันนารีใช้อธิบายการแทนที่ตัวอักษร (ใช้รหัส unicode)

คำสั่ง

พารามิเตอร์	คำอธิบาย
<i>maketrans()</i>	คือ การสร้างตารางจับคู่ตัวอักษร
<i>translate()</i>	คือการนำตารางจับคู่ตัวอักษรมาแทนที่ในตัวแปรสตริง

ตัวอย่างที่ 5.27 การเข้ารหัสด้วยตารางจับคู่ตัวอักษร

```
s = "Information Science"
table = s.maketrans("aeiou", "12345")
print(s.translate(table))
```

ผลลัพธ์ :

```
Inf4rm1t34n Sc32nc2
```

ตัวอย่างที่ 5.27 สร้างตัวแปรตารางจับคู่ตัวอักษร เช่น แทน a=1, e=2, i=3, o=4 และ u=5 ด้วยคำสั่ง table = s.maketrans("aeiou", "12345") จากนั้นใช้คำสั่ง translate เพื่อนำตารางจับคู่ที่

สร้างขึ้นมามีงานกับตัวแปร s ด้วยคำสั่ง s.translate(table) ผลลัพธ์การใช้ตารางจับคู่ตัวอักษรของคำว่า "Information Science" คือ "Inf4rm1t34n Sc32nc2"

5.25 การตรวจสอบสตริง

ประกอบด้วยคำสั่ง isalpha isascii isdecimal isdigit islower isnumeric isprintable isspace istitle และ isupper

ตารางที่ 5.3 แสดงฟังก์ชันการตรวจสอบสตริงแบบต่าง ๆ

คำสั่ง	คำอธิบาย
"information Science".isalpha()	เป็นอักษร a-z A-Z หรือไม่: False (เพราะมี space)
"Informãtion Science".isascii()	เป็นแอสกี (ASCII) หรือไม่: False (เพราะมี ã)
"\u0033".isdecimal()	เป็นตัวเลข unicode 0-9 หรือไม่: True
"1234567890".isdigit()	เป็นตัวเลข 0-9 หรือไม่ใช้กับ unicode: True
"information science".islower()	เป็นตัวพิมพ์เล็กหรือไม่: True
"84000".isnumeric()	เป็นตัวเลขหรือไม่: True
"!@#\$%^&*()".isprintable()	เป็นตัวที่พิมพ์ได้หรือไม่: True
"\t\n ".isspace()	เป็น space bar หรือไม่: True
"Information Science".istitle()	ตัวแรกของคำเป็นตัวพิมพ์ใหญ่หรือไม่: True
"INFORMATION SCIENCE".isupper()	เป็นตัวพิมพ์ใหญ่หรือไม่: True

5.26 การสกัดอีเมลออกจากเอกสารด้วย Regular Expression

Regular Expression คือ การตรวจจับค่าที่มีรูปแบบ เช่น อีเมล มีรูปแบบเป็น อะไรก็ได้@อะไรก็ได้.อะไรก็ได้

ตารางที่ 5.4 แสดง Meta Character ของ Regular Expression

Meta Character	ความหมาย	ตัวอย่าง
	หรือ	"คนไทย ชาวไทย" เช่น คนไทย ชาวไทย
.	อักขระใด ๆ จำนวน 1 ตัว	.ar เช่น การ ขาน คาน aar bar car ฯลฯ
^	คำเริ่มต้น	^พระ เช่น พระจอมเกล้า
\$	คำสุดท้าย	เกล้า\$ เช่น พระจอมเกล้า
[]	อักขระภายในก้ามปู	"[bcr]at" เช่น bat cat rat
-	จาก .. ถึง ..	"F[0-9]" เช่น F0 F1 F2 ... F9
*	ซ้ำได้หลายครั้ง	"01*" เช่น 0 01 011 0111 ฯลฯ
+	ซ้ำหนึ่งครั้งขึ้นไป	"01+" เช่น 01 011 0111 ฯลฯ
?	ซ้ำได้เพียง 1 ครั้ง	"01?" เช่น 0 01
{ }	ระบุจำนวนการซ้ำ	"1{4}" เช่น 1111

Meta Character	ความหมาย	ตัวอย่าง
{m,n}	ระบุจำนวนการซ้ำมีเงื่อนไข	"1{2,4}" เช่น 11 111 1111
\w	ตัวอักษร	\wvar เช่น การ ขาน คาน avar bar car ฯลฯ

ตัวอย่างที่ 5.28 การสกัดอีเมลออกจากเอกสารด้วย Regular Expression

```
s = """ในเอกสารนี้มีอีเมลประกอบด้วย chakkrit@gmail.com และ kritsawan@yahoo.com และมี
เพิ่มอีกหนึ่งรายชื่อ คือ woravit@chaiyo.com ขอขอบคุณ"""
from re import findall
findall("\w+@\w+\.\w+", s)
```

ผลลัพธ์ :

```
['chakkrit@gmail.com', 'kritsawan@yahoo.com', 'woravit@chaiyo.com']
```

ตัวอย่างที่ 5.28 ขอใช้คำสั่ง findall() ภายในโมดูล re ด้วยคำสั่ง from re import findall เรียกใช้คำสั่ง findall("\w+@\w+\.\w+",s) เมื่อตัวแปร s เป็นข้อมูลชนิดสตริง ผลลัพธ์จะสกัดเอารายชื่ออีเมลออกมาได้ ได้ดังนี้ ['chakkrit@gmail.com', 'kritsawan@yahoo.com', 'woravit@chaiyo.com']

5.27 การอ่านข้อมูลจาก Wikipedia เก็บไว้ในตัวแปรสตริง

ไลบรารี wikipedia-API ใช้สำหรับสกัดข้อความ , ส่วนต่าง ๆ ในหน้าวิกิ, ลิงค์, หมวดหมู่ (category), การแปลเนื้อหาภาษาอื่น ๆ จากเอกสารวิกิพีเดีย มีวิธีการใช้งานดังนี้ การติดตั้ง ใช้คำสั่งต่อไปนี้

```
!sudo pip install wikipedia
!sudo pip install wikipedia-api
```

ตัวอย่างที่ 5.29 การอ่านข้อมูลจาก Wikipedia เก็บไว้ในตัวแปรสตริง

```
import wikipedia as wiki
wiki.set_lang("th")
s = wiki.summary("โลก", sentences=1)
print(s)
```

ผลลัพธ์ :

โลก (บาลี: loka; อังกฤษ: world) มีความหมายโดยปริยายหมายถึงหมู่มนุษย์ รวมทั้งอารยธรรมมนุษย์ โดยรวมทั้งหมด โดยเฉพาะในด้านประสบการณ์ ประวัติศาสตร์ หรือสภาพของมนุษย์โดยทั่ว ๆ ไป ทั้งนี้ คำว่า ทั่วโลก หมายถึงสถานที่ใด ๆ บนดาวเคราะห์โลกในทางปรัชญามองโลกอยู่ 2 แบบ

ตัวอย่างที่ 5.29 เป็นการสืบค้นคำว่า "โลก" จาก wikipedia ผลลัพธ์จะแสดงดังตัวอย่างด้านบน โดยที่การใช้งานจะต้องเชื่อมต่ออินเทอร์เน็ตเนื่องจากเป็นการร้องขอข้อมูลจากแม่ข่ายของ wikipedia

5.28 การตัดคำภาษาไทย

โปรแกรมตัดคำภาษาไทยมีอยู่หลายตัว อาทิ kucut , swatch, lexitron , deepcut และ pythainlp ในหัวข้อนี้ใช้ไลบรารี pythainlp มีขั้นตอนดังนี้

```
!pip install pythainlp
```

ตัวอย่างที่ 5.30 การตัดคำภาษาไทย

```
from pythainlp import word_tokenize as thaicut
s = "เด็กเอ๋ยเด็กน้อยความรู้เรายังด้อยเร่งศึกษาเมื่อเติบโตใหญ่เราจะได้มีวิชาเป็นเครื่องหาเลี้ยงชีพ  
สำหรับตน"
z = thaicut(s)
print(z)
```

ผลลัพธ์ :

```
['เด็ก', 'เอ๋ย', 'เด็กน้อย', 'ความรู้', 'เรา', 'ยัง', 'ด้อย', 'เร่ง', 'ศึกษา', 'เมื่อ', 'เติบโตใหญ่', 'เรา', 'จะ', 'ได้', 'มี',  
'วิชา', 'เป็น', 'เครื่อง', 'หาเลี้ยงชีพ', 'สำหรับ', 'ตน']
```

ตัวอย่างที่ 5.30 คำสั่ง thaicut ใช้ตัดคำภาษาไทย โดยป้อนข้อความที่ต้องการตัดลงไปจะได้ผลลัพธ์เป็นคำที่ถูกตัดเก็บไว้ในตัวแปรลิสต์

5.29 การแปลงอักษรไทยรหัส TIS-620 เป็น UTF-8

ในการเข้าแปลงรหัสจาก TIS-620 เป็น UTF8 โดยการเข้ารหัสเป็น cp1252 จากนั้นถอดรหัสนั้นกลับมาเป็น TIS-620 ดังนี้

ตัวอย่างที่ 5.31 การแสดงผลภาษาไทยที่เข้ารหัส Window-874 หรือ TIS-620

```
def tis620_utf8(text):
    b = text.encode('cp1252')
    th = b.decode('tis-620')
    return th

s1 = "...ÁÊËÔËÒÃøÒÁ"
s2 = tis620_utf8(s1)
print(s2)
```

ผลลัพธ์

```
จ.มหาสารคาม
```

แบบฝึกหัดท้ายบท

1. จงอธิบายการแยกสตริงออกจากกันด้วยคำสั่ง split
2. จงอธิบายการใช้เครื่องหมาย Escape Sequence
3. จงอธิบายการอ่านเขียนและแสดงรายชื่อไฟล์
4. จงอธิบายการอ่านไฟล์จากเว็บไซต์ด้วย urlopen
5. จงอธิบายการตรวจจับรูปแบบข้อความด้วย Regular Expression
6. จงเขียนโปรแกรมอ่านข้อมูลจากหน้าเว็บไซต์ จำนวน 1 หน้า จากนั้น สกัดเอาเฉพาะ email ภายในหน้าเว็บออกมาแสดงผล
7. จงเขียนโปรแกรมอ่านข้อมูลจากเว็บที่มีเนื้อเพลง เช่น <https://lyric.in.th> จากนั้น สกัดเอาชื่อเพลง ศิลปิน และเนื้อเพลงและแสดงผลลัพธ์ออกมาบนจอภาพ

บทที่ 6

แฟ้มข้อมูล (File)

แนวคิด

เมื่อเปิดเครื่องคอมพิวเตอร์จะทำให้ ตัวแปรและข้อมูลต่าง ๆ ที่อยู่ภายในหน่วยความจำหลักสูญหายไป ระบบปฏิบัติการมีคำสั่งที่ใช้จัดการไฟล์เพื่ออ่านและบันทึกข้อมูลจากหน่วยความจำเก็บไว้ลงบนหน่วยความจำสำรอง เช่น ฮาร์ดดิสก์ ข้อมูลถูกจัดเก็บอยู่ในรูปแบบของไฟล์หรือแฟ้มข้อมูล (File)

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับการจัดการแฟ้มข้อมูล ระบบไฟล์แบบไบนารีและเท็กซ์โหมด การแสดงรายชื่อไฟล์ การสร้างและลบไฟล์ การสร้างและลบไดเรกทอรี การอ่านไฟล์จากเว็บไซต์

วัตถุประสงค์

1. อธิบายการอ่านและเขียนแฟ้มข้อมูลได้
2. อธิบายความแตกต่างระหว่างเท็กซ์และไบนารีไฟล์ได้
3. อธิบายการอ่านบรรทัดที่ต้องการภายในไฟล์ได้
4. อธิบายการแสดงรายชื่อไฟล์และไดเรกทอรีได้
5. อธิบายการสร้างและลบไฟล์และไดเรกทอรีได้
6. อธิบายการอ่านไฟล์จากเว็บไซต์ได้

6.1 หลักการอ่านและเขียนแฟ้มข้อมูล

ระบบปฏิบัติการเป็นทำหน้าที่จัดการเกี่ยวกับการอ่านเขียนไฟล์ผ่าน FCB (File Control Block) ได้แก่ หมายเลขรหัสแฟ้มข้อมูล , ชื่อแฟ้มข้อมูล, ตำแหน่งที่อยู่ทางกายภาพของแฟ้มข้อมูล, วันที่สร้างแฟ้มข้อมูล เป็นต้น ระบบปฏิบัติการจะอ่านข้อมูลที่ลบล็อกซึ่งเป็นหน่วยที่เล็กที่สุดของสื่อบันทึกข้อมูลชนิดนั้น ๆ เช่น ฮาร์ดดิสก์มีหน่วยที่เล็กที่สุดทางกายภาพคือ เซกเตอร์ (sector)

ในการกระทำกับไฟล์มีลำดับขั้นตอนสำคัญ 3 ขั้นตอนคือ 1) เปิดแฟ้มข้อมูล 2) อ่านหรือเขียนแฟ้มข้อมูล 3) ปิดแฟ้มข้อมูล การเปิดแฟ้มข้อมูล คือการส่งคำสั่งเพื่อบอกระบบปฏิบัติการให้ทำการจองพื้นที่บัฟเฟอร์ หรือสตริม ให้กับแฟ้มข้อมูลและหากสามารถเปิดไฟล์ได้สำเร็จระบบปฏิบัติการจะสร้าง FCB ขึ้นมา ระบบปฏิบัติการจะอ่านข้อมูลและเลื่อนตัวชี้ตำแหน่งเท่ากับจำนวนไบต์ที่อ่าน ระบบปฏิบัติการจะเขียนข้อมูลเอาไว้ในบัฟเฟอร์ เมื่อไรที่บัฟเฟอร์เต็ม ระบบปฏิบัติการจึงจะเริ่มต้นเขียนข้อมูลลงอุปกรณ์ฮาร์ดแวร์ที่ทำหน้าที่เป็นสื่อเก็บข้อมูล

6.2 การเปิดแฟ้มข้อมูล (Open File)

การเปิดแฟ้มข้อมูลหมายถึงการเปิดการติดต่อระหว่างแฟ้มข้อมูล โดยใช้ฟังก์ชัน open()

ไวยากรณ์

`File = open(filename,mode,buffer)`

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
<i>filename</i>	คือ สตริงที่ระบุชื่อไดเรกทอรี ชื่อไดเรกทอรี และชื่อแฟ้ม
<i>Buffer</i>	คือ ขนาดหน่วยความจำที่ใช้สำหรับสร้างบัฟเฟอร์เพื่อพักข้อมูลสำหรับการอ่านหรือเขียนไฟล์
<i>encoding</i>	คือ รหัสข้อความ
<i>Mode</i>	คือ พารามิเตอร์เพื่อบอกโหมดการเปิดแฟ้ม
	r เปิดแฟ้มข้อมูลเพื่ออ่าน
	a เปิดแฟ้มข้อมูลเพื่อเขียนต่อท้ายไฟล์
	w เปิดแฟ้มข้อมูลเพื่อเขียน
	x สร้างแฟ้มข้อมูลใหม่ ถ้ามีไฟล์อยู่แล้วจะคืนค่า error
	t อ่านแบบเท็กซ์โหมด
	b อ่านแบบไบนารีโหมด

6.3 การอ่านและเขียนแฟ้มข้อมูล (Read and Write File)

เมื่อเปิดไฟล์แล้วสิ่งที่ต้องทำในขั้นตอนต่อมา คือการอ่านหรือเขียนไฟล์ หากต้องการอ่านจะใช้ฟังก์ชัน `read()` และหากต้องการเขียนข้อมูลลงไปในไฟล์จะใช้ฟังก์ชัน `write()`

ไวยากรณ์การอ่านไฟล์

```
variable = File.read()
```

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
<i>variable</i>	คือ ตัวแปรสำหรับเก็บค่าที่อ่านมาจากแฟ้มข้อมูล
<i>File</i>	คือ ไฟล์อ็อบเจกต์

ไวยากรณ์การเขียนไฟล์

```
File.write(variable)
```

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
<i>variable</i>	คือ ข้อมูลที่ต้องการเขียนลงแฟ้มข้อมูล
<i>File</i>	คือ ไฟล์อ็อบเจกต์

ตัวอย่างที่ 6.1 การเปิดไฟล์เพื่อเขียน

```
f = open("demofile.txt", "w", encoding="utf-8")
```

```
s = "สาขาวิชาสารสนเทศศาสตร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม"
f.write(s)
f.close()
```

ตัวอย่างที่ 6.1 คำสั่ง `f = open("demofile.txt", "w", encoding="utf-8")` เป็นการเปิดไฟล์เพื่อเขียน (w) และเข้ารหัสด้วย utf-8 รองรับทุกภาษาสากล อาทิ ไทย จีน ญี่ปุ่น ฯลฯ ตัวแปร `s` เป็นข้อความที่ต้องการเขียนลงในไฟล์ จากนั้นเรียกคำสั่ง `write(s)` เพื่อเขียนข้อมูลลงในไฟล์ และทำการปิดไฟล์ด้วยคำสั่ง `close()` ตอนนี้อยู่ในระบบปฏิบัติการจะมีไฟล์เกิดขึ้นชื่อว่า `demofile.txt` และภายในไฟล์จะมีข้อความจากตัวแปร `s` เก็บลงในไฟล์

ตัวอย่างที่ 6.2 การเปิดไฟล์เพื่ออ่าน

```
f = open("demofile.txt", "r", encoding="utf-8")
s = f.read()
f.close()
print(s)
```

ผลลัพธ์ :

```
สาขาวิชาสารสนเทศศาสตร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
```

ตัวอย่างที่ 6.2 การเปิดไฟล์เพื่ออ่านใช้คำสั่ง `open("demofile.txt", "r", encoding="utf-8")` จากนั้นทำการอ่านไฟล์ด้วยคำสั่ง `read()` และปิดไฟล์ด้วยคำสั่ง `close()` ผลลัพธ์ที่ได้ คือ ตัวแปรที่มารับค่าจากการอ่านไฟล์ด้วยคำสั่ง `read()` จากนั้นพิมพ์ค่าในตัวแปร `s` ผลลัพธ์ คือ "สาขาวิชาสารสนเทศศาสตร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม"

6.4 บัฟเฟอร์ (Buffer)

ขนาดของบัฟเฟอร์เป็นเลขจำนวนเต็มใช้สำหรับจองพื้นที่หน่วยความจำสำหรับไฟล์ บัฟเฟอร์จึงเป็นหน่วยความจำที่จองเอาไว้สำหรับเขียนหรืออ่านไฟล์ เมื่อข้อมูลเต็มบัฟเฟอร์ ระบบปฏิบัติการจะนำข้อมูลจากบัฟเฟอร์ลงไปเขียนลงในไฟล์ บัฟเฟอร์ช่วยเพิ่มความเร็วในการเข้าถึงอุปกรณ์ IO

6.5 ความแตกต่างระหว่างแฟ้มข้อมูลแบบ Binary และ Text Mode

ระบบปฏิบัติการวินโดวส์เป็นระบบเดียวที่มีความแตกต่างระหว่างเท็กซ์โหมดและไบนารีโหมด ระบบปฏิบัติการอื่น ๆ เช่น Linux , MacOS, Android จะไม่มีความแตกต่างระหว่างเท็กซ์โหมดและไบนารีโหมด

เท็กซ์โหมดในระบบปฏิบัติการวินโดวส์จะเปลี่ยนเครื่องหมายขึ้นต้นบรรทัดใหม่ '\n' เป็น '\r\n' เรียกว่าอักขระขึ้นต้นบรรทัดใหม่ (Carriage-Return Line Feed) หรือ CRLF

ตัวอย่างที่ 6.3 การอ่านและเขียนไฟล์แบบไบนารี

```
s="Information
Science"
```

```

"""
f = open(r"d:\binary-mode.txt",'wt')
f.write(s)
f.close()

f = open(r"d:\text-mode.txt",'wb')
f.write(s.encode())
f.close()

```

ตัวอย่างที่ 6.3 ตัวแปร s มีชนิดเป็นสตริง เก็บคำว่า "Information\nScience\n" เมื่อ "\n" คือเครื่องหมายขึ้นต้นบรรทัดใหม่ นำตัวแปร s ไปเขียนลงไฟล์แบบ เท็กซ์โหมดและไบนารีโหมด พบว่า ในเท็กซ์โหมดจะทำการแปลงเครื่องหมาย "\n" เป็น "\r\n" ทำให้ขนาดพื้นที่เพิ่มขึ้น 2 ตัวอักษร ดังนั้น ขนาดไฟล์เท็กซ์โหมดจะมีขนาด 22 ตัวอักษร และไฟล์ไบนารีโหมดจะมีขนาด 20 ตัวอักษร เนื่องจากระบบปฏิบัติการวินโดวส์จะเปลี่ยนเครื่องหมายขึ้นต้นบรรทัดใหม่ '\n' เป็น '\r\n' เรียกอักขระขึ้นต้นบรรทัดใหม่ (Carriage-Return Line Feed: CRLF)

ไฟล์แบบไบนารีโหมด

I	n	f	o	R	M	a	t	i	o	n	\n	S	c	i	e	n	c	e	\n
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

ไฟล์แบบเท็กซ์โหมด

I	n	f	o	r	m	a	t	i	o	n	\r	\n	S	c	i	e	n	c	e	\r	\n
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

ขนาดของไฟล์ในเท็กซ์โหมดจะมีขนาดใหญ่กว่าไบนารีโหมด ขึ้นอยู่กับจำนวน '\n' ที่ปรากฏภายในไฟล์

6.6 การอ่านไฟล์ในบรรทัดเดียว

ในขั้นตอนที่ผ่านมาแสดงให้เห็นการอ่านไฟล์ซึ่งต้องดำเนินการ 3 ขั้นตอน หัวข้อนี้เสนอวิธีการอ่านไฟล์เพียงคำสั่งบรรทัดเดียว ดังนี้

ตัวอย่างที่ 6.4 การเปิดไฟล์เพื่ออ่าน

```

s = open(r"d:\text-mode.txt",'r').read()
print(s)

```

ผลลัพธ์ :

```

Information
Science

```

ตัวอย่างที่ 6.4 ตัวแปร s สามารถอ่านไฟล์ด้วยคำสั่ง open() ตามด้วยฟังก์ชัน read() เพื่ออ่านไฟล์ทั้งหมดมาเก็บไว้ในตัวแปร s ซึ่งทำได้ด้วยคำสั่งเพียงบรรทัดเดียว

6.7 อ่านบรรทัดที่ต้องการภายในไฟล์

การเข้าถึงตำแหน่งบรรทัดภายในไฟล์ด้วยการระบุหมายเลขบรรทัดใช้โมดูล linecache โดยเรียกใช้ฟังก์ชัน getline()

ตัวอย่างที่ 6.5 การอ่านข้อมูลจากแฟ้มข้อมูลด้วยการกำหนดหมายเลขบรรทัดที่ต้องการ

```
from linecache import getline
s = getline(r"d:\text-mode.txt",2)
print(s)
```

ผลลัพธ์ :

```
Science
```

ตัวอย่างที่ 6.5 เป็นการอ่านบรรทัดที่ 2 ของไฟล์ด้วยคำสั่ง getline() ผลลัพธ์ แสดงคำว่า "Science" (ใช้ไฟล์จากตัวอย่างที่ 6.3)

6.8 การอ่านรายชื่อไฟล์ภายในไดเรกทอรีด้วย glob()

คำสั่ง glob() ใช้อ่านรายชื่อไฟล์และไดเรกทอรี

ตัวอย่างที่ 6.6 การอ่านรายชื่อไฟล์

```
from glob import glob
fn = glob(r"d:\*.txt")
print(fn)
```

ผลลัพธ์ :

```
['d:\\binary-mode.txt', 'd:\\text-mode.txt']
```

ตัวอย่างที่ 6.6 ตัวแปร fn ใช้สำหรับเก็บผลลัพธ์รายชื่อไฟล์ที่ได้จากฟังก์ชัน glob() โดยในตัวอย่างกำหนดการค้นหไฟล์ r"d:*.txt" ซึ่งเป็นไฟล์ในไดเรกทอรี d:\ นามสกุล *.txt ซึ่งมีอยู่ 2 ไฟล์สร้างขึ้นจากตัวอย่างที่ผ่านมา ผลลัพธ์ คือ ['d:\\binary-mode.txt', 'd:\\text-mode.txt']

6.9 การตรวจสอบขนาดไฟล์

การตรวจสอบขนาดไฟล์ด้วยคำสั่ง getsize() ภายในโมดูล os (operating system)

ตัวอย่างที่ 6.7 การตรวจสอบขนาดไฟล์

```
from glob import glob
fn = glob(r"d:\*.txt")
```

```
for i in fn:
    print("%s ขนาดไฟล์ %d ไบต์"%(i,os.path.getsize(i)))
```

ผลลัพธ์ :

```
d:\binary-mode.txt ขนาดไฟล์ 22 ไบต์
d:\text-mode.txt ขนาดไฟล์ 20 ไบต์
```

ตัวแปรที่ 6.7 ตัวแปร fn ใช้สำหรับเก็บผลลัพธ์รายชื่อไฟล์ที่ได้จากฟังก์ชัน glob() โดยในตัวอย่างกำหนดการค้นหาไฟล์ r"d:*.txt" ซึ่งเป็นไฟล์ในไดเรกทอรี d:\ นามสกุล *.txt ซึ่งมีอยู่ 2 ไฟล์สร้างขึ้นจากตัวอย่างที่ผ่านมา ผลลัพธ์ คือ ['d:\\binary-mode.txt', 'd:\\text-mode.txt']

6.10 การลบไฟล์และไดเรกทอรี

โมดูล os (operating system) เป็นไลบรารีสำหรับการจัดการไฟล์ ได้แก่ การลบไฟล์ remove() การลบไดเรกทอรี rmdir() และการตรวจสอบการมีอยู่ของไฟล์ os.path.exists()

หมายเหตุ: การลบด้วย rmdir() ต้องเป็นไดเรกทอรีเปล่าไม่มีไฟล์อยู่ภายในนั้น

ตัวอย่างที่ 6.8 การลบไฟล์

```
import os
if os.path.exists(r"d:\demofile.txt"):
    os.remove("demofile.txt")
else:
    print("ไม่มีไฟล์ d:\demofile.txt")
```

ตัวอย่างที่ 6.8 การลบไฟล์ควรตรวจสอบว่ามีไฟล์อยู่จริงหรือไม่ด้วยคำสั่ง os.path.exists() ถ้ามีไฟล์อยู่จริงจึงลบไฟล์ d:\demofile.txt ถ้าไม่มีไฟล์หรือไฟล์ถูกลบไปแล้วจะแสดงข้อความว่า "ไม่มีไฟล์ d:\demofile.txt"

ตัวอย่างที่ 6.9 การสร้างและลบโฟลเดอร์ด้วยโมดูล os

```
import os
os.mkdir(r"d:\myfolder")
os.rmdir(r"d:\myfolder")
```

ตัวอย่างที่ 6.9 การสร้างไดเรกทอรีใช้คำสั่ง os.mkdir() การลบไดเรกทอรีใช้คำสั่ง os.rmdir() โดยที่ภายในไดเรกทอรีต้องไม่มีไฟล์

ตัวอย่างที่ 6.10 การลบไฟล์และโฟลเดอร์ด้วยคำสั่ง rmtree()

```
import shutil, os
os.mkdir(r"d:\myfolder")
open(r"d:\myfolder\demofile.txt", "w").write("Hello")
shutil.rmtree(r"d:\myfolder")
```

ตัวอย่างที่ 6.10 การลบไดเรกทอรีและไฟล์ภายในไดเรกทอรีใช้คำสั่ง `rmtree()` ภายในโมดูล `shutil` ทดลองสร้างไดเรกทอรีด้วยคำสั่ง `mkdir()` จากนั้นสร้างไฟล์ `demofile.txt` เก็บไว้ในไดเรกทอรีดังกล่าว จากนั้นลบไดเรกทอรีด้วยคำสั่ง `shutil.rmtree(r"d:\myfolder")` เพื่อลบไฟล์และไดเรกทอรีพร้อมกัน

6.11 ฟังก์ชันภายในอ็อบเจกต์เพิ่มข้อมูล

คำสั่งภายในอ็อบเจกต์ไฟล์แสดงได้ดังนี้

ตัวอย่างที่ 6.11 ฟังก์ชันภายในอ็อบเจกต์ไฟล์

คำสั่ง	คำอธิบาย
<code>fp = open("test.txt","w")</code>	เปิดไฟล์เพื่อเขียน
<code>print(fp.closed)</code>	<code>closed</code> ปิดไฟล์หรือยัง: <code>False</code>
<code>print(fp.encoding)</code>	<code>encoding</code> เข้ารหัสตัวอักษรแบบไหน: <code>UTF-8</code>
<code>print(fp.isatty())</code>	<code>isatty</code> เป็น terminal (<code>tty</code>) ใช่มั้ย: <code>False</code>
<code>print(fp.mode)</code>	<code>mode</code> เปิดด้วยโหมดอะไร: <code>w</code>
<code>print(fp.name)</code>	<code>name</code> ไฟล์ที่เปิดชื่ออะไร: <code>demofile.txt</code>
<code>print(fp.seekable())</code>	เลื่อนหัวอ่านไปยังตำแหน่งต่าง ๆ (<code>seek</code>) ได้มั้ย: <code>True</code>
<code>print(fp.readable())</code>	<code>readable</code> ไฟล์นี้อ่านได้มั้ย: <code>False</code>
<code>print(fp.writable())</code>	<code>writable</code> ไฟล์นี้เขียนได้มั้ย: <code>True</code>
<code>fp.close()</code>	ปิดไฟล์

6.12 การอ่านไฟล์ด้วยการระบุตำแหน่งภายในไฟล์ `seek` และ `tell`

คำสั่ง `seek()` เป็นการเคลื่อนหัวอ่านไปยังตำแหน่งที่ต้องการ

คำสั่ง `read()` คือ การอ่านข้อมูลจากตำแหน่งปัจจุบันเป็นต้นไป

คำสั่ง `tell()` คือ การบอกตำแหน่งปัจจุบันของหัวอ่าน

ตัวอย่างที่ 6.12 การอ่านไฟล์ด้วยการระบุตำแหน่งภายในไฟล์ `seek` และ `tell`

```
open("demofile.txt","w").write("Information Sceince")
f = open("demofile.txt", "r")
f.seek(12)
s = f.read(7)
print(s, f.tell())
f.seek(0)
s = f.read(11)
print(s, f.tell())
f.close()
```

ผลลัพธ์ :

```
Sceince 19
Information 11
```

ตัวอย่างที่ 6.12 การเขียนข้อมูล "Information Sceince" เก็บไว้ในไฟล์ demofile.txt ด้วยคำสั่ง `open("demofile.txt","w").write("Information Sceince")` จากนั้นเปิดไฟล์ด้วยโหมดการอ่าน ดังนี้ `f = open("demofile.txt", "r")` คำสั่ง `seek(12)` คือ เลื่อนหัวอ่านไปตำแหน่งที่ 12 จากนั้น `read(7)` อ่านข้อมูลออกมา 7 ตัวอักษร ตอนนี้หัวอ่านเลื่อนจากตำแหน่ง 12 ไปอีก 7 ตัวอักษร ดังนั้นตำแหน่งปัจจุบันคือ $12+7 = 19$ พิมพ์ข้อมูลที่อยู่ในตัวแปร `s` และตำแหน่งหัวอ่านออกมา ด้วยคำสั่ง `print(s, f.tell())` ผลลัพธ์คือ "Science"

เคลื่อนหัวอ่านกลับไปตำแหน่งที่ 0 ด้วยคำสั่ง `seek(0)` จากนั้นอ่านข้อมูลออกมา 11 ตัวอักษร ด้วยคำสั่ง `read(11)` ตอนนี้หัวอ่านจะเคลื่อนจากตำแหน่ง $0 + 11$ ตัวอักษร แสดงผลลัพธ์ตัวแปร `s` และตำแหน่งหัวอ่านปัจจุบัน ด้วยคำสั่ง `print(s, f.tell())` ผลลัพธ์ คือ "Information 11"

6.13 การตัดข้อมูลท้ายไฟล์ทิ้งด้วย truncate()

คำสั่ง `truncate()` ทำหน้าที่ตัดข้อมูลท้ายไฟล์ทิ้ง

ไวยากรณ์การอ่านไฟล์

```
file.truncate(size)
```

ค่าพารามิเตอร์

พารามิเตอร์	คำอธิบาย
Size	คือ ขนาดข้อมูลที่ต้องการเก็บไว้ในไฟล์

ตัวอย่างที่ 6.13 การตัดข้อมูลท้ายไฟล์ทิ้งด้วย truncate()

```
open("demofile.txt","w").write("Information Sceince")
f = open("demofile.txt", "r+")
f.truncate(11)
f.close()
print(open("demofile.txt", "r").read())
```

ผลลัพธ์ :

```
Information
```

ตัวอย่างที่ 6.13 การเขียนข้อความ "Information Sceince" ลงในไฟล์ demofile.txt ด้วยคำสั่ง `open("demofile.txt","w").write("Information Sceince")` จากนั้นเปิดไฟล์เพื่อทำการตัดส่วนท้ายไฟล์ทิ้งไปด้วยการเปิดไฟล์เพื่ออ่าน `f = open("demofile.txt", "r+")` จากนั้นกำหนดขนาดที่ต้องการลงในฟังก์ชัน `truncate()` ในตัวอย่างนี้กำหนดให้เหลือข้อมูลในไฟล์ 11 ตัวอักษร เมื่อปิดไฟล์แล้วขนาดไฟล์จากเดิมจำนวน 19 ตัวอักษรจะเหลือ 11 โดยเหลือเพียงคำว่า "Information"

6.14 การอ่านไฟล์จากเว็บด้วย urllib

ไลบรารี urllib ใช้สำหรับการอ่านไฟล์จากลิงค์ URLs ด้วยโปรโตคอล http, ftp และ https ประกอบด้วยคำสั่ง urllib.request และ urllib.parse

คำสั่งภายในไลบรารี urllib

คำสั่ง	คำอธิบาย
urllib.request	เปิดและอ่านข้อมูลจากลิงค์ URLs
urllib.parse	การแปลงลิงค์ URLs

ตัวอย่างที่ 6.14 การอ่านข้อมูลจากเว็บด้วย urllib

```
import urllib
a = "https://th.wikipedia.org/"
b = "wiki/ประเทศไทย"
c = urllib.parse.quote(b)
url = a+c
s = urllib.request.urlopen(url).read().decode('utf-8', 'ignore')
print(s[0:300])
```

ผลลัพธ์ :

```
<!DOCTYPE html><html class="client-nojs" lang="th" dir="ltr"><head><meta charset="
UTF-8"/><title>ประเทศไทย - วิกีพีเดีย</title><script>document.documentElement.classN
ame="client-js";RLCONF={"wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],"wgDigi
tTransformTable":["",""],"wgDateFormat
```

ตัวอย่างที่ 6.14 การอ่านเนื้อหาจากลิงค์ "https://th.wikipedia.org/wiki/ประเทศไทย" ด้วยไลบรารี urllib ใช้คำสั่ง import urllib จากนั้นแบ่ง url เป็น 2 ส่วน คือ "https://th.wikipedia.org/" และ urllib.parse.quote("wiki/ประเทศไทย") เก็บไว้ในตัวแปร url มีค่า "https://th.wikipedia.org/wiki/%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B9%80%E0%B8%97%E0%B8%A8%E0%B9%84%E0%B8%97%E0%B8%A2"

คำสั่ง s = urllib.request.urlopen(url).read().decode('utf-8', 'ignore') อ่านข้อมูลจาก url เข้ามาเก็บไว้ในตัวแปร s โดยถอดรหัสตัวอักษรเป็น utf-8 และไม่ต้องสนใจอักขระที่ผิดพลาดด้วย "ignore" นำผลลัพธ์มาแสดงผล 300 ตัวอักษรด้วยคำสั่ง print(s[0:300])

แบบฝึกหัดท้ายบท

1. จงอธิบายการอ่านและเขียนแฟ้มข้อมูล
2. จงอธิบายความแตกต่างระหว่างเท็กซ์และไบนารีไฟล์
3. จงอธิบายการอ่านบรรทัดที่ต้องการภายในไฟล์

4. จงอธิบายการแสดงรายชื่อไฟล์และไดเรกทอรี
5. จงอธิบายการสร้างและลบไฟล์และไดเรกทอรี
6. จงอธิบายการอ่านไฟล์จากเว็บไซต์
7. จงเขียนโปรแกรมเพื่ออ่านข้อมูลจากเว็บ lyric เช่น <https://lyric.in.th/lyric.php?n=5102> โดยสกัด ชื่อเพลง ศิลปิน และเนื้อเพลง จากนั้น เก็บข้อมูลลงบนไฟล์

บทที่ 7

ตัวแปรดิกชันนารีและแม่พรีดิคส์

แนวคิด

ตัวแปรดิกชันนารี หรือตัวแปรพจนานุกรม (dictionary) เป็นการเก็บคีย์และค่าของคีย์ (key และ value) เป็นโครงสร้างข้อมูลที่มีประโยชน์และมีประสิทธิภาพ

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ การประกาศใช้งานตัวแปรดิกชันนารี การเข้าถึงข้อมูลภายในตัวแปรดิกชันนารี การตรวจสอบคีย์ภายในดิกชันนารี การเคลียร์ข้อมูลทั้งหมดในดิกชันนารี การแปลงตัวแปรลิสต์เป็นดิกชันนารี การดึงข้อมูลในตัวแปรดิกชันนารีด้วยคำสั่ง pop การอัปเดตค่าตัวแปรให้กับตัวแปรดิกชันนารี การรวมตัวแปรชนิดดิกชันนารีเข้าด้วยกัน และฟังก์ชันแม่พรีดิคส์

วัตถุประสงค์

1. อธิบายการประกาศใช้ตัวแปรดิกชันนารีได้
2. อธิบายการเข้าถึงสมาชิกภายในตัวแปรดิกชันนารีได้
3. อธิบายการตรวจสอบคีย์ภายในดิกชันนารีได้
4. อธิบายการ pop และอัปเดตค่าตัวแปรให้กับตัวแปรดิกชันนารีได้
5. อธิบายการใช้งานฟังก์ชันแม่พรีดิคส์และตัวแปรดิกชันนารีได้

7.1 การประกาศและเข้าถึงตัวแปรชนิดดิกชันนารี

ตัวแปรดิกชันนารี มีรูปแบบการประกาศตัวแปร ดังนี้

ไวยากรณ์การประกาศตัวแปรดิกชันนารี

```
dict_name = { key: value }
```

คำอธิบายการประกาศตัวแปรดิกชันนารี

ชื่อเรียก	คำอธิบาย
<code>dict_name</code>	คือ ชื่อตัวแปรที่ต้องการประกาศเป็นชนิดดิกชันนารี
<code>Key</code>	คือ คีย์เวิร์ด
<code>Value</code>	คือ ค่าของคีย์

ตัวอย่างที่ 7.1 การประกาศตัวแปรดิกชันนารี

```
d = { }  
d = {'Computer' : 'คอมพิวเตอร์', 'Programming': 'โปรแกรมมิ่ง' }  
print(d['Computer'])  
print(d['Programming'])  
print(len(d))
```

ผลลัพธ์ :

```
คอมพิวเตอร์
โปรแกรมมิ่ง
2
```

ตัวอย่างที่ 7.1 ตัวแปร d เป็นดิกชันนารีเปล่า ด้วยคำสั่ง d={} และเพิ่มข้อมูล 2 คำคือ "Computer" และ "Programming" แสดงค่าของตัวแปรดิกชันนารีด้วยการระบุด้วยค่าคีย์ เมื่อ d['Computer'] มีข้อมูลคำว่า "คอมพิวเตอร์" และ d['Programming'] มีข้อมูลคำว่า "โปรแกรมมิ่ง"

7.2 การเพิ่มและแก้ไขข้อมูลในตัวแปรดิกชันนารีด้วยคำสั่ง update()

คำสั่ง update() ให้เพิ่มและแก้ไขข้อมูลภายในดิกชันนารี

ไวยากรณ์การเพิ่มและแก้ไขข้อมูลในตัวแปรดิกชันนารี

```
dict_name.update({ key: value })
```

คำอธิบายการประกาศตัวแปรดิกชันนารี

ชื่อเรียก	คำอธิบาย
<i>dict_name</i>	คือ ตัวแปรดิกชันนารี
<i>Key</i>	คือ คีย์เวิร์ดที่ต้องการเพิ่มหรือแก้ไข
<i>Value</i>	คือ ค่าของคีย์

ตัวอย่างที่ 7.2 การอัปเดตค่าภายในตัวแปรดิกชันนารีด้วยคำสั่ง update()

```
d = { }
d.update({'Computer' : 'คอมพิวเตอร์' })
print(d.keys())
d.update({'Programming': 'โปรแกรมมิ่ง'})
print(d.keys())
print(d['Computer'], d['Programming'])
d.update({'Computer' : 'คณิตกรณ์' })
print(d.keys())
print(d['Computer'], d['Programming'])
```

ผลลัพธ์ :

```
dict_keys(['Computer'])
dict_keys(['Computer', 'Programming'])
คอมพิวเตอร์ โปรแกรมมิ่ง
dict_keys(['Computer', 'Programming'])
คณิตกรณ์ โปรแกรมมิ่ง
```

ตัวอย่างที่ 7.2 ตัวแปร d เป็นดิกชันนารี กำหนดให้ "Computer" มีค่า "คอมพิวเตอร์" จากนั้นทำการแก้ไขให้เป็นคำว่า "คณิตกรณ์" ด้วยคำสั่ง d['Computer'] = "คณิตกรณ์" จะพบว่าคียังมีค่าเท่าเดิม 2 คีย์ คือ Computer และ Programming ตามลำดับ

7.3 การสร้างดิกชันนารีใหม่จากตัวแปรลิสต์

ฟังก์ชัน dict() ใช้สำหรับสร้างตัวแปรดิกชันนารี โดยใช้ร่วมกับฟังก์ชัน zip() เพื่อรวมคีย์และค่าข้อมูลเข้าด้วยกัน พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 7.3 การสร้างตัวแปรดิกชันนารีจากตัวแปรลิสต์

```
k = ["one", "two", "three"]
v = ["หนึ่ง", "สอง", "สาม"]
d = dict(zip(k,v))
print(d)
```

ผลลัพธ์ :

```
{'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม'}
```

ตัวอย่างที่ 7.3 ตัวแปร k และ v เก็บข้อมูลคีย์และค่าของคีย์ จากนั้นฟังก์ชัน dict() ใช้สำหรับรวมคู่คีย์และข้อมูลเข้าด้วยกันด้วยฟังก์ชัน zip() ผลลัพธ์ที่ได้เป็นตัวแปรดิกชันนารีที่สร้างจากตัวแปรลิสต์มีค่า {'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม'}

7.4 การอ่านข้อมูลภายในตัวแปรดิกชันนารีด้วยคำสั่ง get()

ไพธอนอนุญาตให้เข้าถึงข้อมูลภายในตัวแปรดิกชันนารีด้วยคำสั่ง get() โดยมีรูปแบบดังนี้

ไวยากรณ์

```
dict_name.get(key)
```

คำอธิบาย

ชื่อเรียก	คำอธิบาย
dict_name	คือ ตัวแปรดิกชันนารี
Key	คือ คีย์ในตัวแปรดิกชันนารี

ตัวอย่างที่ 7.4 การสร้างตัวแปรดิกชันนารีจากตัวแปรลิสต์

```
d = {'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม'}
print(d.get('one') , d['one'])
```

ผลลัพธ์ :

```
หนึ่ง หนึ่ง
```

ตัวอย่างที่ 7.4 ตัวแปร d มีคีย์ 3 ตัว คือ ['one', 'two', 'three'] การใช้ฟังก์ชัน get() มีค่าเท่ากับการถึงด้วยเครื่องหมาย [] เช่น d.get('one') ให้ผลลัพธ์เช่นเดียวกับคำสั่ง d['one']

7.5 การ pop ข้อมูลภายในดิกชันนารีด้วยคำสั่ง pop()

การ pop คือการนำเอาข้อมูลภายในดิกชันนารีออกมาด้วยคำสั่ง pop()

ไวยากรณ์

```
dict_name.pop(key)
```

คำอธิบาย

ชื่อเรียก	คำอธิบาย
<i>dict_name</i>	คือ ตัวแปรดิกชันนารี
<i>Key</i>	คือ คีย์ที่ต้องการ pop ออกมาจากตัวแปรดิกชันนารี

ตัวอย่างที่ 7.5 การ pop ข้อมูลภายในดิกชันนารีด้วยคำสั่ง pop()

```
d = {'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม'}
s = d.pop('one')
print(d)
print(s)
```

ผลลัพธ์ :

```
{'two': 'สอง', 'three': 'สาม'}
หนึ่ง
```

ตัวอย่างที่ 7.5 ตัวแปร d เป็นชนิดดิกชันนารีมีคีย์ 3 ตัว คือ ['one', 'two', 'three'] เมื่อ pop('one') จะเป็นการนำค่าของคีย์ 'one' ออกมาใส่ในตัวแปร s ซึ่งมีค่าคือ "หนึ่ง" ตัวแปร d จะเหลือสมาชิกเพียง 2 ตัว คือ {'two': 'สอง', 'three': 'สาม'}

7.6 การรวมตัวแปรดิกชันนารีเข้าด้วยกันด้วยคำสั่ง update()

นอกจากจะทำการเพิ่มและแก้ไขข้อมูลภายในตัวแปรดิกชันนารีแล้วยังรวมตัวแปรดิกชันนารีสองตัวเข้าด้วยกันได้

ตัวอย่างที่ 7.6 การรวมดิกชันนารีสองตัวเข้าด้วยกัน

```
d1 = {'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม'}
d2 = {'four': 'สี่', 'five': 'ห้า'}
d1.update(d2)
print(d1)
```

ผลลัพธ์ :

```
{'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม', 'four': 'สี่', 'five': 'ห้า'}
```

ตัวอย่างที่ 7.6 ตัวแปรดิกชันนารี 2 ตัว คือ d1 และ d2 นำมารวมกันด้วยคำสั่ง update() จะเป็นการรวม ตัวแปรดิกชันนารีเข้าด้วยกัน ผลลัพธ์ d1 มีค่าเท่ากับ {'one': 'หนึ่ง', 'two': 'สอง', 'three': 'สาม', 'four': 'สี่', 'five': 'ห้า'}

7.7 คำสั่ง map และ lambda

คำสั่ง map() ทำหน้าที่แมปฟังก์ชันกับอาร์กิวเมนต์เข้าด้วยกัน

คำสั่ง lambda คือ การสร้างฟังก์ชันที่ไม่มีชื่อ (Anonymous Function)

ตัวอย่างที่ 7.7 การใช้คำสั่ง map และ lambda

```
x = [ 2 , 3 , 4, 2, 1 ]
list(map(lambda a:a**2, x))
```

ผลลัพธ์ :

```
[4, 9, 16, 4, 1]
```

ตัวอย่างที่ 7.7 การนำสมาชิกภายในตัวแปร x ซึ่งมีค่า [2 , 3 , 4, 2, 1] มายกกำลังสอง ผลลัพธ์จะมีค่า [2², 3², 4², 2², 1²] เมื่อคำสั่ง lambda a:a**2 หมายถึง รับค่ามาเข้ามาเก็บในตัวแปร a จากนั้น ยกกำลังสองด้วยคำสั่ง a**2 คำสั่ง map จะทำการแมปค่าตัวแปรให้กับฟังก์ชัน ในตัวอย่างนี้ ฟังก์ชัน คือ lambda a:a**2 และตัวแปร คือ x ดังนั้น ค่าตัวแปรที่อยู่ใน x แต่ละตัวจะถูกส่งเข้าไปให้ฟังก์ชัน lambda a:a**2 ผลลัพธ์จึงมีค่าเท่ากับ [4, 9, 16, 4, 1]

ตัวอย่างที่ 7.8 การใช้ map และ lambda กับตัวแปรลิสต์ 2 ชุด

```
x = [ 2 , 3 , 4, 2, 2 ]
y = [ 2, 1, 2, 4, 5 ]
list(map(lambda a,b:a+b , x, y))
```

ผลลัพธ์ :

```
[4, 4, 6, 6, 7]
```

ตัวอย่างที่ 7.8 สมาชิกในตัวแปร x และ y จะถูกนำมาบวกกันทีละคู่ ผลลัพธ์จะมีค่า [2+2, 3+1, 4+2, 2+4, 2+5] ฟังก์ชันที่สร้างขึ้น คือ lambda a,b : a+b หมายถึง ฟังก์ชันรับอาร์กิวเมนต์ 2 ตัว คือ a และ b และคืนค่าเป็นผลรวม a+b ฟังก์ชัน map() จะแมปฟังก์ชันและตัวแปรเข้าด้วยกัน ในตัวอย่างนี้มีตัวแปร 2 ชุด ดังนั้นเรียกใช้ด้วยคำสั่ง map(function, x,y) เมื่อ function คือ lambda a, b:a+b จากนั้น ผลลัพธ์ที่ได้แปลงเป็นตัวแปรลิสต์ด้วยคำสั่ง list() ผลลัพธ์มีค่า [4, 4, 6, 6, 7]

7.8 คำสั่ง filter

คำสั่ง filter() ทำหน้าที่กรองค่าออกมาจากตัวแปรลิสต์ โดยผลลัพธ์ที่ได้จะกรองตามเงื่อนไข ดังนี้

ตัวอย่างที่ 7.9 การใช้ filter และ lambda กรองข้อมูล

```
x = [0, 1, 2, 0, 3, 5]
z = list(filter(lambda a:a != 0, x))
print(z)
```

ผลลัพธ์ :

```
[1, 2, 3, 5]
```

ตัวอย่างที่ 7.9 ตัวแปร x มีค่า [0, 1, 2, 0, 3, 5] คำสั่ง filter() เป็นการกรองให้เหลือค่าที่ตรงตามเงื่อนไขที่กำหนด ในตัวอย่างนี้ฟังก์ชันกรองคือ `lambda a:a != 0` หมายถึง ค่าที่ส่งเข้ามาแต่ละรอบจะเก็บไว้ในตัวแปร a จากนั้นตรวจสอบว่าถ้า `a != 0` ให้คืนค่าออกมาจากฟังก์ชัน ถ้าค่า a ที่ส่งเข้าไปเท่ากับ 0 ให้ตัดทิ้งไป คำสั่ง `filter(function,x)` เมื่อ x เป็นอาร์กิวเมนต์ที่ส่งให้ฟังก์ชันโดยจะเก็บในตัวแปร a ผลลัพธ์หลังจากการกรองแล้วมีค่า [1, 2, 3, 5]

7.9 คำสั่ง reduce

คำสั่ง reduce() ทำหน้าที่ดำเนินการผลลัพธ์ให้เหลือเพียงค่าเดียวตามฟังก์ชันที่กำหนดขึ้น

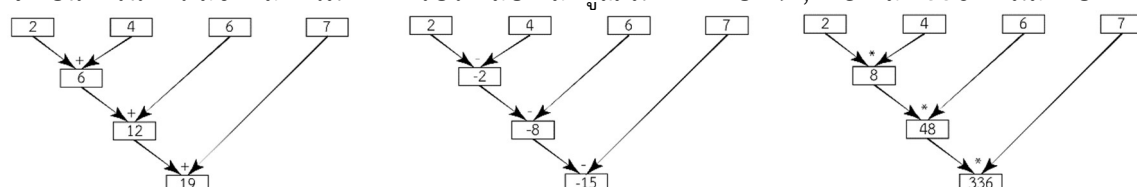
ตัวอย่างที่ 7.10 คำสั่ง reduce

```
from functools import reduce
x = [2, 4, 6, 7]
m = reduce(lambda a,b:a+b, x)
n = reduce(lambda a,b:a-b, x)
o = reduce(lambda a,b:a*b, x)
print(m, n, o)
```

ผลลัพธ์ :

```
19 -15 336
```

ตัวอย่างที่ 7.10 ตัวแปร x ประกอบด้วยตัวเลข [2, 4, 6, 7] เมื่อผ่านฟังก์ชัน `reduce()` จะทำการส่งตัวแปร x ให้กับฟังก์ชันที่สร้างขึ้นด้วยคำสั่ง `lambda` โดยมีฟังก์ชันเท่ากับ `a+b`, `a-b` และ `a*b` ผลลัพธ์จะเป็นการนำตัวเลขแต่ละตัวมาทำการบวก ลบ และคูณ มีค่าเท่ากับ 19, -15 และ 336 ตามลำดับ



ภาพประกอบที่ 7.1 แสดงการ reduce ด้วยฟังก์ชันการบวก ลบ และคูณ

7.10 การสืบค้นข้อมูลรายชื่อวิทยานิพนธ์ 10,000 รายการ

ในหัวข้อนี้แสดงตัวอย่างการโหลดไฟล์รายชื่อวิทยานิพนธ์จากเว็บไซต์และทำการแยกแต่ละรายการด้วยเครื่องหมายขึ้นต้นบรรทัดใหม่ ("\n") จากนั้นใช้คำสั่ง filter เพื่อกรองคำค้น และแสดงผลรายชื่อวิทยานิพนธ์ ที่ผ่านการสืบค้น

ตัวอย่างที่ 7.11 การสืบค้นข้อมูลรายชื่อวิทยานิพนธ์

```
import urllib, urllib.request
url = "http://dsdi.msu.ac.th/programming/thailis-10000.txt"
s = urllib.request.urlopen(url).read().decode('utf-8', 'ignore').split("\n")
q = "การสืบสาน"
result = list(filter(lambda a:q in a,s))
print("พบ '%s' ทั้งหมด %d เล่ม"%(q,len(result)))
for i in range(len(result)):
    print("%d) %s"%(i+1,result[i]))
```

ผลลัพธ์ :

```
พบ 'การสืบสาน' ทั้งหมด 8 เล่ม
1) การสืบสานการทอผ้าพื้นเมืองของชุมชน ตำบลหาดเสี้ยว อำเภอสรีษัณาลัย จังหวัดสุโขทัย
2) การสืบสานและอนุรักษ์วัฒนธรรมประเพณีของกลุ่มชาติพันธุ์ชาวอาข่าที่เื้อต่อเยาวชนอาข่า :
กรณีศึกษาเยาวชนอาข่า หมู่บ้านอาโยะใหม่ ตำบลแม่สลองใน อำเภอแม่ฟ้าหลวง จังหวัดเชียงราย
3) มหาวิทยาลัยราชภัฏ : การสืบสานพระราชปณิธาน
4) การสืบสานวัฒนธรรมท้องถิ่นของสภาวัฒนธรรมจังหวัดขอนแก่น
5) วิถีคิดและบทบาทการสืบสานวัฒนธรรมของปราชญ์ท้องถิ่น กรณีศึกษา นายสาร สารทัศนันทน์
6) บทบาทของวัดกับการสืบสานวัฒนธรรม กรณีศึกษาวัดพระบาทภูพานคำ อำเภออุบลรัตน์ จังหวัด
ขอนแก่น
7) เพลงพื้นบ้านราชบุรีกับการสืบสานมรดกทางวัฒนธรรม
8) การสืบสานภูมิปัญญากระบือข้าวลายช้างกระแบบดั้งเดิมไทยอีสานเชิงพิลึกส์
```

ตัวอย่างที่ 7.11 ไฟล์ thailis-10000.txt เก็บรายชื่อวิทยานิพนธ์จำนวน 10,000 เล่ม เก็บไว้ที่ลิงค์ <http://dsdi.msu.ac.th/programming/> อ่านเนื้อไฟล์เข้ามาเก็บในตัวแปร s โดยถอดรหัสเป็น utf-8 ถ้าพบข้อความที่โหลดไม่ได้ให้ ignore จากนั้นแยกด้วยเครื่องหมายขึ้นต้นบรรทัดใหม่ ("\n") จะได้ตัวแปร s เก็บรายชื่อวิทยานิพนธ์ 10,000 เล่ม

คำสั่ง `list(filter(lambda a:q in a,x))` เป็นการกรองให้เหลือเฉพาะรายการที่มีคำว่า "การสืบสาน" และเก็บผลลัพธ์ไว้ในตัวแปร result จากนั้นพิมพ์จำนวนผลลัพธ์จากการสืบค้นด้วย `len(result)` จากนั้นแสดงรายการผลลัพธ์ โดยวนรอบจาก 0 ถึง ความยาวของผลลัพธ์ `len(result)` ในแต่ละรอบพิมพ์ด้วยคำสั่ง `print("%d) %s"%(i+1,result[i]))` เมื่อ `i+1` คือ ลำดับของผลลัพธ์ที่พบเริ่มจากลำดับที่ 0 จึงให้เพิ่มค่าด้วย 1 เพื่อแสดงหมายเลขลำดับของผลการสืบค้น

แบบฝึกหัดท้ายบท

1. จงอธิบายการประกาศใช้ตัวแปรดิกชันนารี
2. จงอธิบายการเข้าถึงสมาชิกภายในตัวแปรดิกชันนารี
3. จงอธิบายการตรวจสอบคีย์ภายในดิกชันนารี
4. จงอธิบายการ pop และอัปเดตค่าตัวแปรให้กับตัวแปรดิกชันนารี
5. จงอธิบายการใช้งานฟังก์ชันแม่ปริตวิสและตัวแปรดิกชันนารี
6. จงเขียนโปรแกรมเพื่ออ่านข้อมูลวิทยานิพนธ์จากเว็บ <https://tdc.thailis.or.th> จำนวน 1 เรื่อง 5 เรื่อง เช่น https://tdc.thailis.or.th/tdc/browse.php?option=show&browse_type=title&titleid=10 จากนั้น สกัดข้อมูล title, title alternative และ description และเก็บไว้ในตัวแปรพจนานุกรม

บทที่ 8

การเขียนโปรแกรมภาษาไพธอนสำหรับสถิติเบื้องต้น

แนวคิด

ภาษาไพธอนมีไลบรารีด้านสถิติให้ใช้งานหลากหลายแต่สำหรับบทนี้จะเรียนรู้การเขียนคำสั่งเพื่อคำนวณหาค่าสถิติพื้นฐานเพื่อให้เข้าใจค่าสถิติแต่ละตัวในเชิงการเขียนโปรแกรม

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ ความรู้เบื้องต้นเกี่ยวกับสถิติ ตัวแปรต้นและตัวแปรตาม ค่าแนวโน้มสู่ศูนย์กลาง การเขียนโปรแกรมคำนวณค่าเฉลี่ย มัธยฐาน (Median) ฐานนิยม (Mode) ค่าเบี่ยงเบนมาตรฐาน ความแปรปรวน ระดับความเบ้ของข้อมูล ค่าความโด่งของข้อมูล เส้นโค้งปกติ การแปลงข้อมูลเป็นคะแนน Z การเขียนโปรแกรมคำนวณพื้นที่ใต้เส้นโค้งปกติด้วยการอินทิเกรต การคำนวณค่าเปอร์เซ็นต์ที่มากกว่าหรือน้อยกว่าและคะแนนที่อยู่ระหว่างช่วงที่กำหนดขึ้น

วัตถุประสงค์

1. อธิบายการเขียนโปรแกรมคำนวณค่าเฉลี่ย มัธยฐาน ฐานนิยมได้
2. อธิบายการเขียนโปรแกรมคำนวณค่าเบี่ยงเบนมาตรฐาน ความแปรปรวนของข้อมูลได้
3. อธิบายการเขียนโปรแกรมความเบ้และความโด่งของข้อมูลได้
4. อธิบายการเขียนโปรแกรมแปลงค่า z และคำนวณพื้นที่ใต้เส้นโค้งปกติด้วยการอินทิเกรตได้

8.1 การเขียนโปรแกรมคำนวณค่าเฉลี่ย

ค่าเฉลี่ย คือ ผลรวมของข้อมูลทั้งหมด / จำนวนข้อมูล เขียนแทนด้วยสูตร

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i$$

เมื่อ n คือจำนวนข้อมูล และ x_i คือ สมาชิกของตัวแปร x ลำดับที่ i

ตัวอย่างที่ 8.1 การเขียนโปรแกรมคำนวณค่าเฉลี่ย

```
def ค่าเฉลี่ย(data):  
    n = len(data)  
    mean = sum(data)/n  
    return mean  
  
x = [1,3,2]  
print("ค่าเฉลี่ยของ %s เท่ากับ %.2f"%(str(x), ค่าเฉลี่ย(x)))
```

ผลลัพธ์ :

ค่าเฉลี่ยของ [1, 3, 2] เท่ากับ 2.00

ตัวอย่างที่ 8.1 การเขียนฟังก์ชัน ค่าเฉลี่ย() มีอาร์กิวเมนต์คือตัวแปร data ในตอนเริ่มต้นกำหนดค่าตัวแปร $x = [1, 3, 2]$ จากนั้นส่งเข้าให้ฟังก์ชัน ค่าเฉลี่ย ภายในฟังก์ชันจะคำนวณหาค่า n คือจำนวนสมาชิก มีค่าเท่ากับ $\text{len}(\text{data})$ และคำนวณค่าเฉลี่ยเท่ากับ ผลรวมข้อมูลหารด้วยจำนวนข้อมูล จากนั้น return ผลลัพธ์ค่าเฉลี่ย เมื่อคำสั่ง `print("ค่าเฉลี่ยของ %s เท่ากับ %.2f"%(str(x), ค่าเฉลี่ย(x)))` มีอีกกระบวนการแสดงผล `%s` คือนำค่าจาก `str(x)` มาใส่ `%s` และ `%.2f` คือการแสดงผลทศนิยมสองตำแหน่งของฟังก์ชันค่าเฉลี่ย ผลลัพธ์ มีค่า "ค่าเฉลี่ยของ [1, 3, 2] เท่ากับ 2.00"

`print("ค่าเฉลี่ยของ %s เท่ากับ %.2f"%(str(x), ค่าเฉลี่ย(x)))`

ภาพประกอบที่ 8.1 แสดงอักขระควบคุม `%s` และ `%.2f` แสดงข้อความและทศนิยมสองตำแหน่ง

8.2 การเขียนโปรแกรมคำนวณค่ามัธยฐาน (Median)

ค่ามัธยฐาน หาได้จากนำข้อมูลมาเรียงลำดับจากน้อยไปมากหรือมากไปน้อย โดยมัธยฐานคือข้อมูลที่อยู่ตรงกลางพอดี

ตัวอย่างข้อมูล [4.3, 2.9, 2.7, 4.1, 40.8, 3.4, 4.7, 3.7, 3.1, 4.7]

1) เรียงลำดับข้อมูลจากน้อยไปมากหรือมากไปน้อย ได้ผลลัพธ์ [2.7, 2.9, 3.1, 3.4, 3.7, 4.1, 4.3, 4.7, 4.7, 40.8]

2) มัธยฐาน คือ ค่าข้อมูลที่อยู่ตำแหน่งกึ่งกลางข้อมูลที่เรียงลำดับแล้ว

ตัวอย่างที่ 8.2 การเขียนโปรแกรมคำนวณค่ามัธยฐาน (Median)

```
x = [3, 2, 3, 5, 7, 1]
x.sort()
if (len(x) % 2) == 0:
    a = x[int(len(x)/2)-1]
    b = x[int(len(x)/2)]
    c = (a+b)/2
    print("ค่ามัธยฐานของ %s คือ (%.2f + %.2f/2 มีค่าเท่ากับ %.2f"%(str(x),a,b,c))
else:
    c = x[int(len(x)/2)-1]
    print("ค่ามัธยฐานของ %s มีค่าเท่ากับ %.2f"%(str(x),c))
```

ผลลัพธ์ :

ค่ามัธยฐานของ [1, 2, 3, 3, 5, 7] คือ $(3.00 + 3.00/2)$ มีค่าเท่ากับ 3.00

ตัวอย่างที่ 8.2 ตัวแปร x มีค่า [3, 2, 3, 5, 7, 1] นำมาเรียงลำดับจากน้อยไปมากด้วยคำสั่ง `x.sort()` มีค่า [1, 2, 3, 3, 5, 7] ค่ามัธยฐาน คือ ข้อมูลที่ตำแหน่งตรงกลาง ในตัวอย่างนี้มีจำนวนสมาชิกเป็นเลขคู่ (6 ตัว) ดังนั้น คำนวณค่าเฉลี่ยของตำแหน่งที่อยู่ตรงกลาง คือ $3+3 / 2$ เท่ากับ 3

เครื่องหมายดำเนินการ % คือ การหารเอาเศษ เช่น $3\%2$ คือ การนำ $3/2$ เหลือเศษ 1 ดังนั้น $3\%2$ เท่ากับ 1 ในตัวอย่างนี้มีจำนวนสมาชิก 6 ดังนั้น $6\%2$ มีค่า 0 คือ หารลงตัว เมื่อหารลงตัวค่ามัธยฐานคือตำแหน่งตรงกลาง คือ $3+3/2$ มีค่าเท่ากับ 3 โดยคำนวณจาก $a+b/2$ เมื่อ a คือ ข้อมูลที่ตำแหน่ง $x[\text{int}(\text{len}(x)/2)-1]$ และ b คือข้อมูลที่ตำแหน่ง $x[\text{int}(\text{len}(x)/2)]$ ผลลัพธ์ที่ได้คือ ค่ามัธยฐานของ [1, 2, 3, 3, 5, 7] คือ $(3.00 + 3.00)/2$ มีค่าเท่ากับ 3.00

8.3 การเขียนโปรแกรมคำนวณค่าฐานนิยม (Mode)

ค่าฐานนิยม (Mode) คือค่าที่พบบ่อยที่สุด เช่น [3, 2, 3, 5, 7, 1] พบว่า เลข 3 พบบ่อยที่สุด ดังนั้น ค่าฐานนิยม คือ 3

ตัวอย่างที่ 8.3 การเขียนโปรแกรมคำนวณค่าฐานนิยม

```
x = [3, 2, 3, 5, 7, 2, 1]
u = list(set(x))
m = max(list(map(lambda i: x.count(i), u)))
print("ตัวแปร u มีค่า %s พบตัวเลขที่เกิดบ่อยสุดจำนวน %d ครั้ง ยังไม่รู้ว่าจะอะไร"%(u,m))
for i in u:
    if x.count(i) == m:
        print("» พบแล้วฐานนิยมคือ ค่าที่พบบ่อยสุด %d ครั้ง คือ เลข %d"%(m,i))
```

ผลลัพธ์ :

```
ตัวแปร u มีค่า [1, 2, 3, 5, 7] พบตัวเลขที่เกิดบ่อยสุดจำนวน 2 ครั้ง ยังไม่รู้ว่าจะอะไร
» พบแล้วฐานนิยมคือ ค่าที่พบบ่อยสุด 2 ครั้ง คือ เลข 2
» พบแล้วฐานนิยมคือ ค่าที่พบบ่อยสุด 2 ครั้ง คือ เลข 3
```

ตัวอย่างที่ 8.3 ตัวแปร x มีค่า [3, 2, 3, 5, 7, 2, 1] ตัวแปร u มีค่าที่ไม่ซ้ำกันเพราะคำสั่ง set() ใช้กำจัดค่าที่ซ้ำกันทิ้งไป ตัวแปร u จึงมีค่า [1, 2, 3, 5, 7] คำสั่ง `lambda i: x.count(i)` คือ การนับค่า i ที่อยู่ในตัวแปร u ว่ามีกี่ตัว ผลลัพธ์คือ [1, 2, 2, 1, 1, 1] หมายถึง พบเลข 1 จำนวน 1 ตัว , พบเลข 2 จำนวน 2 ตัว , พบเลข 3 จำนวน 2 ตัว พบเลข 5 จำนวน 1 ตัว และพบเลข 7 จำนวน 1 ตัว

วนเข้าไปในตัวแปร u คือ [1, 2, 3, 5, 7] และตรวจว่าค่า `x.count()` ที่มีค่าเท่ากับ m ซึ่งเป็นค่าที่มีความถี่สูงสุด ถ้าพบแล้วให้พิมพ์ค่าฐานนิยมด้วยคำสั่ง `print("» พบแล้วฐานนิยมคือ ค่าที่พบบ่อยสุด %d ครั้ง คือ เลข %d"%(m,i))` ดังนั้น ผลลัพธ์จะมีค่าฐานนิยมอยู่ 2 ค่า คือ เลข 2 และ เลข 3

8.4 การเขียนโปรแกรมคำนวณค่าเบี่ยงเบนมาตรฐาน

ค่าเบี่ยงเบนมาตรฐาน คำนวณจากสูตร

$$sd = \sqrt{\frac{\sum |x_i - \bar{x}|^2}{n}}$$

เมื่อ sd คือ ค่าเบี่ยงเบนมาตรฐาน
 $\bar{x}$ คือ ค่าเฉลี่ย
 x_i คือ ข้อมูลลำดับที่ i
 n คือ จำนวนข้อมูล

ตัวอย่างที่ 8.4 การเขียนโปรแกรมคำนวณค่าเบี่ยงเบนมาตรฐาน

```
from math import sqrt
x = [3, 2, 3, 5, 7, 2, 1]
n = len(x)
mean = sum(x)/n
print(list(map(lambda i: ((i-mean)**2)/(n), x)))
sd = sqrt(sum(list(map(lambda i: ((i-mean)**2)/(n), x))))
print("ค่าเบี่ยงเบนมาตรฐานของข้อมูล %s มีค่า %.2f"%(str(x), sd))
```

ผลลัพธ์ :

```
[0.01, 0.23, 0.01, 0.41, 1.97, 0.23, 0.74]
ค่าเบี่ยงเบนมาตรฐานของข้อมูล [8, 5, 2, 4, 10, 1, 7, 3, 6, 9] มีค่า 1.91
```

ตัวอย่างที่ 8.4 ตัวแปร x มีค่า `[3, 2, 3, 5, 7, 2, 1]` มีจำนวนสมาชิก 7 ตัว เก็บไว้ในตัวแปร n คำนวณจาก $n = \text{len}(x)$ ตัวแปร mean คือค่าเฉลี่ย มีค่า $\text{sum}(x) / n$ คำสั่ง $\text{map}(\text{lambda } i: ((i-\text{mean})^2)/n, x)$ คือ การนำค่า x แต่ละตัวไปใส่ลงในตัวแปร i จากนั้นแทนค่าและคำนวณด้วยคำสั่ง $((i - \text{mean})^2)/n$ คือ นำค่า x แต่ละตัวลบด้วยค่าเฉลี่ยและนำไปยกกำลังสองและหารด้วยจำนวนข้อมูล ผลลัพธ์ที่ได้จะมีค่า `[0.01, 0.23, 0.01, 0.41, 1.97, 0.23, 0.74]` นำมารวมกัน ด้วยคำสั่ง $\text{sum}()$ จากนั้นถอดรากที่สองด้วยคำสั่ง $\text{sqrt}()$ ผลลัพธ์มีค่า ค่าเบี่ยงเบนมาตรฐานของข้อมูล `[8, 5, 2, 4, 10, 1, 7, 3, 6, 9]` มีค่า 1.91

8.5 การเขียนโปรแกรมคำนวณค่าความแปรปรวน

ค่าความแปรปรวน คำนวณจากสูตร

$$s^2 = \frac{\sum |x_i - \bar{x}|^2}{n}$$

เมื่อ s^2 คือ ค่าความแปรปรวน
 $\bar{x}$ คือ ค่าเฉลี่ย
 x_i คือ ข้อมูลลำดับที่ i
 n คือ จำนวนข้อมูล

ตัวอย่างที่ 8.5 การเขียนโปรแกรมคำนวณค่าความแปรปรวน

```
x = [3, 2, 3, 5, 7, 2, 1]
n = len(x)
mean = sum(x)/n
s2 = sum(list(map(lambda i: ((i-mean)**2)/n, x)))
print("ค่าความแปรปรวนของข้อมูล %s มีค่า %.2f"%(str(x), s2))
```

ผลลัพธ์ :

ค่าความแปรปรวนของข้อมูล [3, 2, 3, 5, 7, 2, 1] มีค่า 3.63

ตัวอย่างที่ 8.5 ตัวแปร x มีค่า [3, 2, 3, 5, 7, 2, 1] ตัวแปร n มีค่าเท่ากับขนาดของ x คือ 7 ตัวแปร mean คือผลรวมของ xหารด้วย n ตัวแปร s2 จะนำตัวแปร x ทีละตัวส่งเข้าไปให้ฟังก์ชัน lambda ซึ่งจะแทนด้วยตัวแปร i นำค่า i ลบด้วยค่าเฉลี่ยยกกำลังสองหาร n ด้วยคำสั่ง lambda i: ((i - mean)**2)/n จากนั้นพิมพ์ผลลัพธ์ ด้วยคำสั่ง print() ผลลัพธ์ที่ได้ คือ "ค่าความแปรปรวนของข้อมูล [3, 2, 3, 5, 7, 2, 1] มีค่า 3.63"

8.6 การเขียนโปรแกรมคำนวณระดับความเบ้ของข้อมูล (Skewness)

ค่าความระดับความเบ้ของข้อมูล คำนวณจากสูตร

$$skewness = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{sd} \right)^3$$

เมื่อ *skewness* คือ ค่าความเบ้ของข้อมูล

$\bar{x}$ คือ ค่าเฉลี่ย

x_i คือ ข้อมูลลำดับที่ *i*

n คือ จำนวนข้อมูล

การวิเคราะห์ค่าความเบ้

ค่าความเบ้มีค่าเป็นลบ แสดงว่ารูปทรงเป็นแบบเบ้ซ้าย

ค่าความเบ้มีค่าเป็นบวก แสดงว่ารูปทรงเป็นแบบเบ้ขวา

ค่าความเบ้มีค่าเป็น 0 แสดงว่ารูปทรงเป็นแบบสมมาตร

ตัวอย่างที่ 8.6 การเขียนโปรแกรมคำนวณความเบ้ของข้อมูล

```
def ส่วนเบี่ยงเบนมาตรฐาน(data):
    n = len(data)
    mean = sum(data)/n
    sd = sqrt(sum(list(map(lambda i: ((i-mean)**2)/(n), data))))
    return sd
```

```
x = [10, 9, 10, 11, 12, 9, 11, 11, 8, 10]
n = len(x)
mean = sum(x)/n
sd = ส่วนเบี่ยงเบนมาตรฐาน(x)
skewness = (n/((n-1)*(n-2))) * (sum(list(map(lambda i: ((i-mean)/sd)**3, x))))
print("ความเบ้ของ %s มีค่า %.2f" % (str(x), skewness))
```

ผลลัพธ์ :

```
ความเบ้ของ [10, 9, 10, 11, 12, 9, 11, 11, 8, 10] มีค่า -0.27
```

ตัวอย่างที่ 8.6 ตัวแปร x มีข้อมูล $x = [10, 9, 10, 11, 12, 9, 11, 11, 8, 10]$ ตัวแปร n เก็บขนาดข้อมูลด้วยคำสั่ง `len(x)` ตัวแปร `mean` เก็บค่าเฉลี่ย ด้วยคำสั่ง `sum(x)/n` ตัวแปร `sd` เก็บค่าส่วนเบี่ยงเบนมาตรฐาน คำนวณจากฟังก์ชัน `ส่วนเบี่ยงเบนมาตรฐาน()` ตัวแปร `skewness` เก็บค่าความเบ้ คำนวณจากสูตรความเบ้ เมื่อ $(n/((n-1)*(n-2))) * \text{sum}(\text{list}(\text{map}(\text{lambda } i: ((i-\text{mean})/\text{sd})^3, x)))$ โดยคำสั่ง `lambda i: ((i-mean)/sd)**3` เป็นฟังก์ชัน ที่รับค่า i เป็นอาร์กิวเมนต์และนำค่า i ลบค่าเฉลี่ยหารด้วยส่วนเบี่ยงเบนมาตรฐานทั้งหมดยกกำลังสาม ผลลัพธ์มีค่า ความเบ้ของ $[10, 9, 10, 11, 12, 9, 11, 11, 8, 10]$ มีค่า -0.27

8.7 การเขียนโปรแกรมคำนวณค่าความโด่งของข้อมูล (kurtosis)

ค่าความโด่งของข้อมูล คำนวณจากสูตร

$$kurtosis = \left(\frac{n(n+1)}{(n-1)(n-2)(n-3)} \times \sum \left(\frac{x_i - \bar{x}}{sd} \right)^4 \right) - \left(\frac{3(n-1)^2}{(n-2)(n-3)} \right)$$

เมื่อ $kurtosis$ คือ ค่าความโด่งของข้อมูล

$\bar{x}$ คือ ค่าเฉลี่ยของข้อมูล

sd คือ ค่าเบี่ยงเบนมาตรฐาน

x_i คือ ข้อมูลลำดับที่ i

n คือ จำนวนข้อมูล

การวิเคราะห์ค่าโด่งของข้อมูล

ค่าความโด่งมีค่าน้อยกว่า 0.263 แสดงว่ารูปทรงเป็นแบบแบนราบ

ค่าความโด่งมีค่ามากกว่า 0.263 แสดงว่ารูปทรงเป็นแบบโด่งมาก

ค่าความโด่งมีค่าเท่ากับ 0.263 แสดงว่ารูปทรงเป็นแบบโด่งปกติ

ตัวอย่างที่ 8.7 การเขียนโปรแกรมคำนวณค่าความโด่งของข้อมูล

```
from math import sqrt
def ส่วนเบี่ยงเบนมาตรฐาน(data):
```

```

n = len(data)
mean = sum(data)/n
sd = sqrt(sum(list(map(lambda i: ((i-mean)**2)/(n), data))))
return sd

n = len(data)
sd = ส่วนเบี่ยงเบนมาตรฐาน(data)
mean = sum(x)/n
k1 = (n*(n-1))/((n-1)*(n-2)*(n-3))
k2 = sum(list(map(lambda i: ((i-mean)/sd)**4, x)))
k3 = ((3*(n-1)**2)/((n-2)*(n-3)))
x = [10, 9, 10, 11, 12, 9, 11, 11, 8, 10]
kurtosis = (k1 * k2) - k3
print("ความโด่งของ %s มีค่าเท่ากับ %.2f"%(str(x), kurtosis))

```

ผลลัพธ์ :

ความโด่งของ [10, 9, 10, 11, 12, 9, 11, 11, 8, 10] มีค่าเท่ากับ -0.33

ตัวอย่างที่ 8.7 ตัวแปร x มีค่า [10, 9, 10, 11, 12, 9, 11, 11, 8, 10] ตัวแปร n มีค่าเท่ากับ จำนวนข้อมูล จากคำสั่ง len(x) ตัวแปร sd คำนวณจากฟังก์ชัน ส่วนเบี่ยงเบนมาตรฐาน() ตัวแปร mean คำนวณค่าเฉลี่ยจากคำสั่ง sum(x)/n

ตัวแปร k1 คือพจน์แรกของสูตรคำนวณความโด่ง มีค่า $(n*(n-1))/((n-1)*(n-2)*(n-3))$

ตัวแปร k2 คือพจน์กลางของสูตรคำนวณความโด่งมีค่า $\text{lambda i: } ((i-\text{mean})/\text{sd})^{**4}$ ทำหน้าที่เป็นฟังก์ชันคำนวณค่าในตัวแปร x_i แต่ละตัวลบค่าเฉลี่ย หาค่าเบี่ยงเบนมาตรฐานทั้งหมดยกกำลัง 4 จากนั้นรวมค่าทั้งหมดด้วยคำสั่ง sum() และถอดรากที่สองด้วยคำสั่ง sqrt()

ตัวแปร k3 คือ พจน์ขาสุดของสูตรความโด่ง มีค่า $((3*(n-1)**2)/((n-2)*(n-3)))$

ตัวแปร kurtosis คือค่าความโด่งของข้อมูล มีค่าเท่ากับ $(k1*k2) - k3$ เมื่อนำมาพิมพ์ด้วยคำสั่ง `print("ความโด่งของ %s มีค่าเท่ากับ %.2f"%(str(x), kurtosis))` จะมีค่า ความโด่งของ [10, 9, 10, 11, 12, 9, 11, 11, 8, 10] มีค่าเท่ากับ -0.33

8.8 การเขียนโปรแกรมคำนวณพื้นที่ใต้โค้งปกติด้วยการอินทิเกรต

การแจกแจงปกติมาตรฐานมีลักษณะเป็นเส้นโค้งปกติ หรือรูปประฆังคว่ำ มีค่าเฉลี่ย มัธยฐานและฐานนิยมมีค่าเท่ากัน คำนวณจากสูตร

$$f(z) = \frac{1}{\sqrt{2 * \pi}} e^{\frac{-z^2}{2}}$$

เมื่อ

z คือ คะแนน z คำนวณจากสูตร $z = \frac{x-\mu}{\sigma}$

e คือ ค่าคงที่มีค่า 2.718

π คือ ค่าคงที่มีค่า 3.14

x คือ ข้อมูลที่ต้องการแปลงเป็นคะแนน z

σ คือ ค่าเบี่ยงเบนมาตรฐานของประชากร

μ คือ ค่าเฉลี่ยของประชากร

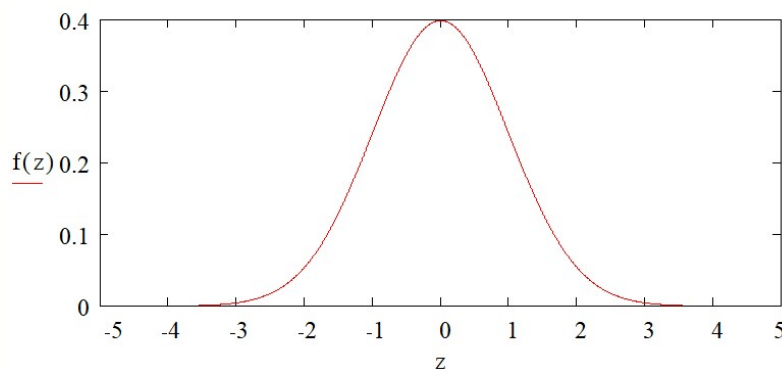
ตัวอย่างที่ 8.8 การเขียนโปรแกรมคำนวณพื้นที่ใต้โค้งปกติด้วยการอินทิเกรตด้วย scipy

```
from math import e,sqrt,pi
import scipy.integrate
f= lambda z: (1/sqrt(2*pi))*e**((-z**2)/2)
integrate = scipy.integrate.quad(f, -5, 5)
p = integrate[0]
print("พื้นที่ใต้โค้งปกติหรือค่าความน่าจะเป็นในช่วง %d ถึง %d มีค่า %.2f"%(-5,5,p))
```

ผลลัพธ์ :

ค่าความน่าจะเป็นในช่วง -5 ถึง 5 มีค่า 1.00

ตัวอย่างที่ 8.8 เป็นฟังก์ชันของเส้นโค้งปกติ มีค่า $f = \text{lambda } z: (1/\sqrt{2\pi}) * e^{((-z^2)/2)}$ เมื่อต้องการหาพื้นที่ใต้เส้นโค้งปกติจาก $z = -5$ ถึง $z=5$ ของฟังก์ชัน f ใช้คำสั่ง `scipy.integrate.quad(f, -5, 5)` ผลรวมพื้นที่ใต้โค้งปกติจะอยู่ในตัวแปร `integrate[0]` นำค่าที่ได้มาพิมพ์จะได้ผลลัพธ์ คือ พื้นที่ใต้โค้งปกติหรือค่าความน่าจะเป็นในช่วง -5 ถึง 5 มีค่า 1.00



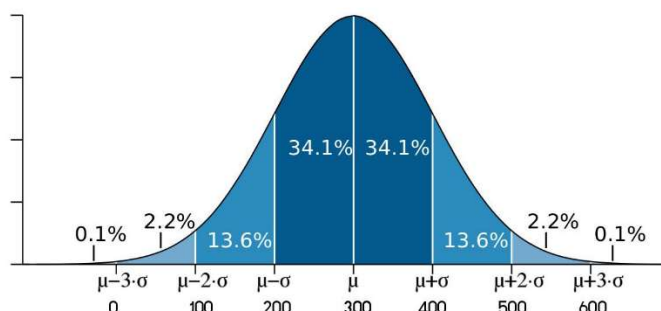
ภาพประกอบที่ 8.2 แสดงเส้นโค้งปกติในช่วงค่า z มีค่า -5 ถึง 5

8.9 การคำนวณพื้นที่ใต้เส้นโค้งปกติ

สมมติว่าข้อมูลมีการแจกแจงปกติ มีค่าเฉลี่ย 300 และส่วนเบี่ยงเบนมาตรฐาน 100 จงคำนวณค่าต่อไปนี้

1. คะแนนมากกว่า 400 คะแนนมีกี่เปอร์เซ็นต์ ?
2. คะแนนน้อยกว่า 500 คะแนนมีกี่เปอร์เซ็นต์ ?
3. คะแนนระหว่าง 200 ถึง 400 มีกี่เปอร์เซ็นต์ ?
4. คะแนนระหว่าง 100 ถึง 500 มีกี่เปอร์เซ็นต์ ?

พิจารณาการเขียนเส้นโค้งปกติเมื่อค่าเฉลี่ยเท่ากับ 300 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 100



ภาพประกอบที่ 8.3 แสดงเส้นโค้งปกติที่มีค่าเฉลี่ย 300 และส่วนเบี่ยงเบนมาตรฐาน 100

พื้นที่ใต้เส้นโค้งปกติแบ่งออกเป็น 8 ส่วน คือ 1) 0.1% 2) 2.2% 3) 13.6% 4) 34.1% 5) 34.1% 6) 13.6% 7) 2.2% และ 8) 0.1%

พิจารณาคำถามต่อไปนี้

1. คะแนนมากกว่า 400 คะแนนมีกี่เปอร์เซ็นต์ ? ตอบ = $13.6 + 2.2 + 0.1 = 15.9\%$
2. คะแนนน้อยกว่า 500 คะแนนมีกี่เปอร์เซ็นต์ ? ตอบ = $13.6 + 34.1 + 34.1 + 13.6 + 2.2 + 0.1 = 97.7\%$
3. คะแนนระหว่าง 200 ถึง 400 มีกี่เปอร์เซ็นต์ ? ตอบ = $34.1 + 34.1 = 68.2\%$
4. คะแนนระหว่าง 100 ถึง 500 มีกี่เปอร์เซ็นต์ ? ตอบ = $13.6 + 34.1 + 34.1 + 13.6 = 95.4\%$

ตัวอย่างที่ 8.9 การเขียนโปรแกรมคำนวณคะแนนที่มากกว่าค่าที่กำหนด

```
from math import e,sqrt,pi
import scipy.integrate

def การคำนวณค่าเปอร์เซ็นต์ที่มากกว่าค่าที่กำหนดขึ้น(mean,sigma,score):
    f = lambda z: (1/sqrt(2*pi))*e**((-z**2)/2)
    z = (score - mean)/sigma
    integrate = scipy.integrate.quad(f, z, 5)
    p = integrate[0]*100
    return p

test = การคำนวณค่าเปอร์เซ็นต์ที่มากกว่าค่าที่กำหนดขึ้น(300, 100, 400)
print("คนที่มีคะแนนมากกว่า 400 คิดเป็น %.1f เปอร์เซ็นต์"%test)
```

ผลลัพธ์ :

คนที่มีความดันมากกว่า 400 คิดเป็น 15.9 เปอร์เซนต์

ตัวอย่างที่ 8.9 การคำนวณคะแนนที่มากกว่า 400 คิดเป็นกี่เปอร์เซนต์ ในขั้นแรกนำค่าคะแนน 400 มาแปลงเป็นคะแนน z ด้วยสูตร $(400 - \text{ค่าเฉลี่ย})/\text{ส่วนเบี่ยงเบนมาตรฐาน}$ จะได้ $(400-300)/100 = 1$ ตอนนี้ค่า z มีค่าเท่ากับ 1 จากนั้นทำการอินทิเกรตจาก 1 ถึง 5 ทำเป็นเปอร์เซนต์ด้วยการคูณด้วย 100 จะได้ผลลัพธ์ คือ คนที่มีความดันมากกว่า 400 คิดเป็น 15.9 เปอร์เซนต์

ตัวอย่างที่ 8.10 การเขียนโปรแกรมคำนวณคะแนนที่น้อยกว่าค่าที่กำหนด

```
from math import e,sqrt,pi
import scipy.integrate

def การคำนวณค่าเปอร์เซนต์ที่น้อยกว่าค่าที่กำหนดขึ้น(mean,sigma,score):
    f= lambda z: (1/sqrt(2*pi))*e**((-z**2)/2)
    z = (score - mean)/sigma
    integrate = scipy.integrate.quad(f, -5, z)
    p = integrate[0]*100
    return p
test = การคำนวณค่าเปอร์เซนต์ที่น้อยกว่าค่าที่กำหนดขึ้น(300, 100, 500)
print("คนที่มีความดันมากกว่า 500 คิดเป็น %.1f เปอร์เซนต์"%test)
```

ผลลัพธ์ :

คนที่มีความดันมากกว่า 500 คิดเป็น 97.7 เปอร์เซนต์

ตัวอย่างที่ 8.10 การคำนวณคะแนนที่น้อยกว่า 500 คิดเป็นกี่เปอร์เซนต์ ในขั้นแรกนำค่าคะแนน 500 มาแปลงเป็นคะแนน z ด้วยสูตร $(500 - \text{ค่าเฉลี่ย})/\text{ส่วนเบี่ยงเบนมาตรฐาน}$ จะได้ $(500-300)/100 = 2$ ตอนนี้ค่า z มีค่าเท่ากับ 2 จากนั้นทำการอินทิเกรตจาก -5 ถึง 2 แปลงเป็นเปอร์เซนต์ด้วยการคูณด้วย 100 จะได้ผลลัพธ์ คือ คนที่มีความดันมากกว่า 500 คิดเป็น 97.7 เปอร์เซนต์

ตัวอย่างที่ 8.11 การเขียนโปรแกรมคำนวณคะแนนที่อยู่ช่วงของค่าที่กำหนด

```
from math import e,sqrt,pi
import scipy.integrate

def คำนวณเปอร์เซนต์ของคะแนนที่อยู่ในช่วงที่กำหนด(mean, sigma, score1, score2):
    f= lambda z: (1/sqrt(2*pi))*e**((-z**2)/2)
    z1 = (score1 - mean)/sigma
    z2 = (score2 - mean)/sigma
    integrate = scipy.integrate.quad(f, z1, z2)
    p = integrate[0]*100
    return p
```

```
test = คำนวณเปอร์เซ็นต์ของคะแนนที่อยู่ในช่วงที่กำหนด(300, 100, 200, 400)
print("คะแนนที่อยู่ในช่วง %d ถึง %d มีอยู่ %.1f เปอร์เซ็นต์"%(200,400,test))

test = คำนวณเปอร์เซ็นต์ของคะแนนที่อยู่ในช่วงที่กำหนด(300, 100, 100, 500)
print("คะแนนที่อยู่ในช่วง %d ถึง %d มีอยู่ %.1f เปอร์เซ็นต์"%(100,500,test))
```

ผลลัพธ์ :

```
คะแนนที่อยู่ในช่วง 200 ถึง 400 มีอยู่ 68.3 เปอร์เซ็นต์
คะแนนที่อยู่ในช่วง 100 ถึง 500 มีอยู่ 95.4 เปอร์เซ็นต์
```

ตัวอย่างที่ 8.11 การคำนวณคะแนนที่อยู่ในช่วง 100 ถึง 500 คิดเป็นกี่เปอร์เซ็นต์ ในขั้นแรกนำค่าคะแนน 100 และ 500 มาแปลงเป็นคะแนน z ด้วยสูตร $z1 = (score1 - mean)/sigma$ และ $z2 = (score2 - mean)/sigma$ จะได้ค่า $z1 = -2.0$ และ $z2 = 2.0$ จากนั้นทำการอินทิเกรตจาก -2 ถึง 2 แปลงเป็นเปอร์เซ็นต์ด้วยการคูณด้วย 100 จะได้ผลลัพธ์ คือ คะแนนที่อยู่ในช่วง 100 ถึง 500 มีอยู่ 95.4 เปอร์เซ็นต์

คะแนนในช่วง 200 ถึง 400 แปลงเป็น $z1 = (200-300)/100$ เท่ากับ -1 และ $z2=(400-300)/100$ เท่ากับ 1.0 ทำการอินทิเกรตในช่วง -1 ถึง 1 จะได้ผลลัพธ์ คะแนนที่อยู่ในช่วง 200 ถึง 400 มีอยู่ 68.3 เปอร์เซ็นต์

แบบฝึกหัดท้ายบท

1. จงอธิบายการเขียนโปรแกรมคำนวณค่าเฉลี่ย มัธยฐาน ฐานนิยม
2. จงอธิบายการเขียนโปรแกรมค่าเบี่ยงเบนมาตรฐาน ความแปรปรวนของข้อมูล
3. จงอธิบายการเขียนโปรแกรมความเบ้และความโด่งของข้อมูล
4. จงการเขียนโปรแกรมแปลงค่า z และคำนวณพื้นที่ใต้เส้นโค้งปกติด้วยการอินทิเกรต
5. จงเขียนโปรแกรมรับค่าตัวเลขจำนวน 10 ชุด จากนั้น เขียนโปรแกรมคำนวณหาค่า เฉลี่ย เบี่ยงเบนมาตรฐาน ค่าความเบ้ และค่าความโด่งของข้อมูล

บทที่ 9

การเขียนโปรแกรมภาษาไพธอนติดต่อบรรณข้อมูล MySQL

แนวคิด

ระบบฐานข้อมูล คือ การจัดเก็บข้อมูลอย่างเป็นระเบียบมีหมวดหมู่ชัดเจนและสามารถเพิ่ม ลบ แก้ไข และสืบค้นภายในฐานข้อมูลได้ ในปัจจุบันมีระบบจัดการฐานข้อมูลหลายตัว ได้แก่ Oracle , PostgreSQL, MySQL, MS SQL Server, SQLite เป็นต้น

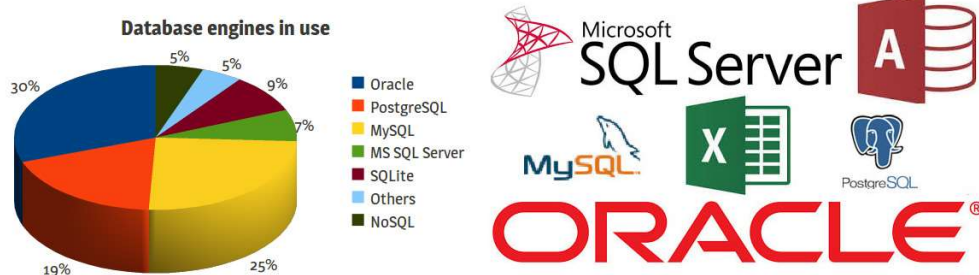
ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ ระบบฐานข้อมูล การติดตั้ง mysql-connector การเชื่อมต่อฐานข้อมูล MySQL เข้ากับ Python การสร้างตาราง การเขียนโปรแกรมเพิ่ม ลบ แก้ไข และสืบค้นด้วยภาษาไพธอน

วัตถุประสงค์

1. อธิบายการสร้างตารางด้วย phpmyadmin ได้
2. อธิบายการเพิ่มข้อมูลในตารางด้วยภาษาไพธอนได้
3. อธิบายการลบข้อมูลในตารางด้วยเงื่อนไขที่กำหนดได้
4. อธิบายการสืบค้นข้อมูลภายในตารางและแสดงผลด้วยภาษาไพธอนได้
5. อธิบายการแก้ไขข้อมูลภายในตารางด้วย SQL และภาษาไพธอนได้

9.1 ระบบฐานข้อมูล

ซอฟต์แวร์ที่ใช้จัดการฐานข้อมูล ได้แก่ 1) ออราเคิล (Oracle) 2) MySQL 3) PostgreSQL 4) Microsoft Excel 5) Microsoft Access 6) Microsoft SQL Server 7) SQLite ฯลฯ



ภาพประกอบที่ 9.1 สัดส่วนการใช้งานฐานข้อมูลแบบต่าง ๆ

ที่มา: <https://dsdi.msu.ac.th>

9.2 การสร้างตารางด้วย phpmyadmin

โปรแกรม phpmyadmin เป็นตัวจัดการฐานข้อมูลผ่านเว็บไซต์ ใช้งานได้ง่ายและได้รับความนิยมในปัจจุบัน ให้นิสิตสร้างตารางชื่อ friend มีโครงสร้างดังต่อไปนี้

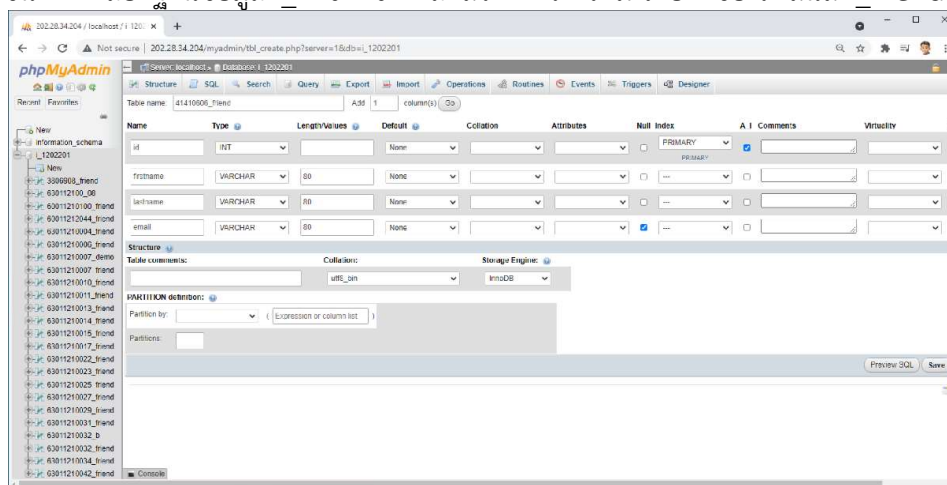
โครงสร้างตาราง

ชื่อคอลัมน์	กำหนด
<i>Id</i>	กำหนดเป็น primary key และ auto_increment คือ ตัวเลขเพิ่มอัตโนมัติ
<i>Firstname</i>	สำหรับเก็บชื่อ มีชนิด varchar ขนาด 80 ตัวอักษรเข้ารหัส UTF8
<i>Lastname</i>	สำหรับเก็บนามสกุล มีชนิด varchar ขนาด 80 ตัวอักษรเข้ารหัส UTF8
<i>Email</i>	สำหรับเก็บอีเมล มีชนิด varchar 80 ตัวอักษรเข้ารหัส UTF8

ขั้นตอนการดำเนินการ

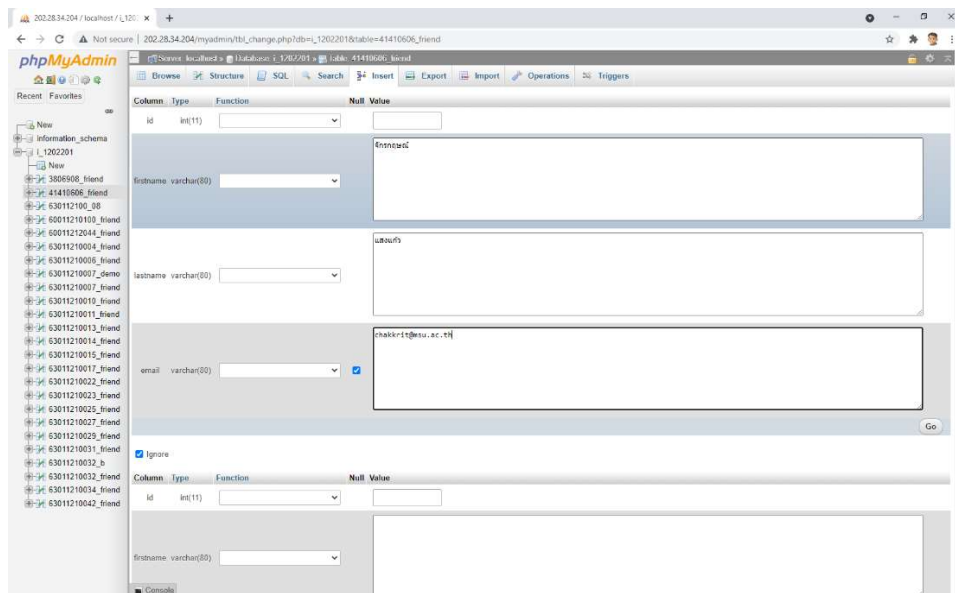
ขั้นที่ 1. เปิดบราวเซอร์และป้อน URL: <http://202.28.34.204/myadmin> จากนั้นป้อน username และ password ด้วย i_user/123456

ขั้นที่ 2. เลือกฐานข้อมูล i_1202201 และสร้างตารางใหม่ โดยตั้งชื่อ รหัสனிสิต_friend ดังนี้



ภาพประกอบที่ 9.2 การสร้างตารางด้วย phpmyadmin

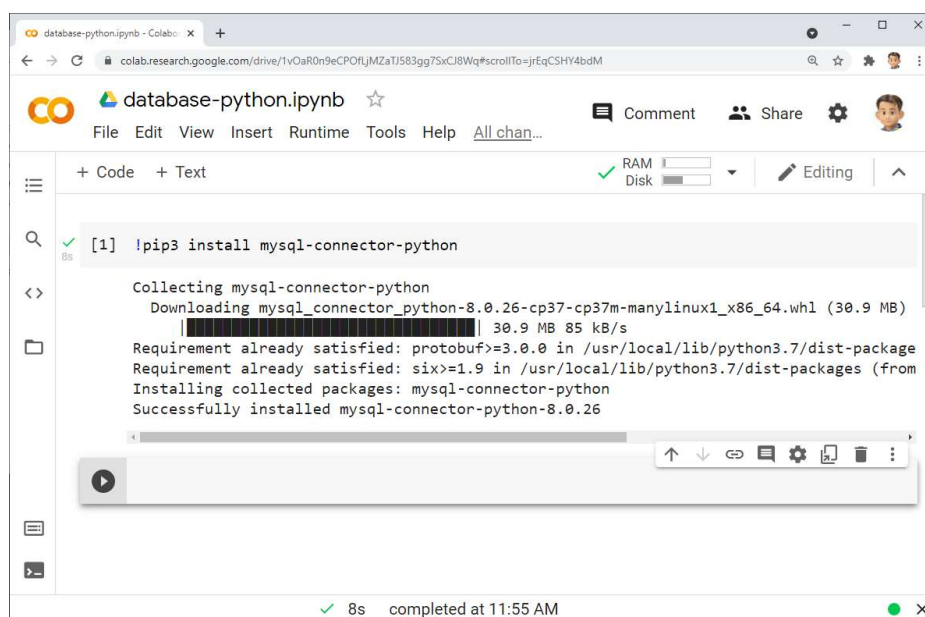
ขั้นที่ 3. ให้นิสิตเพิ่มข้อมูลรายชื่อเพื่อนอย่างน้อย 2 รายการ โดยกดที่แท็บ Insert จากนั้นป้อนข้อมูลในช่อง firstname, lastname และ email และกดปุ่ม Go ดังภาพต่อไปนี้



ภาพประกอบที่ 9.3 แสดงการเพิ่มข้อมูลด้วยโปรแกรม phpmyadmin

9.3 การติดตั้ง mysql-connector

ในการเชื่อมต่อระหว่างภาษาไพธอนและฐานข้อมูล MySQL จะใช้ไลบรารีเพื่ออนุญาตให้ไพธอนติดต่อกับระบบฐานข้อมูล โดยใช้ไลบรารี mysql-connector ในตัวอย่างต่อไปนี้จะใช้ภาษาไพธอนบน <https://colab.research.google.com> ให้คลิกที่ล็อกอินด้วยรหัสและสร้างโปรเจกต์ใหม่และติดตั้ง mysql-connector ด้วยคำสั่ง `!pip3 install mysql-connector-python` จากนั้นกดปุ่ม ▶ เพื่อสั่งให้โปรแกรมเริ่มต้นทำงาน หรือกดแป้น Shift + Enter ได้เช่นเดียวกัน ถ้าติดตั้งเสร็จสมบูรณ์จะแสดงดังภาพต่อไปนี้



ภาพประกอบที่ 9.4 แสดงการติดตั้งโปรแกรม mysql-connector-python

9.4 การเชื่อมต่อฐานข้อมูล MySQL เข้ากับ Python

การเชื่อมต่อฐานข้อมูล MySQL ต้องระบุ หมายเลข IP ของ MySQL Server, ชื่อผู้ใช้, รหัสผ่าน, ชื่อฐานข้อมูล โดยเขียนโค้ดดังนี้

ตัวอย่างที่ 9.1 การเชื่อมต่อฐานข้อมูล

```
import mysql.connector as MySQL
config = {'user': "i_user", 'password': "123456", 'host': "202.28.34.204",
'database': "i_1202201", 'raise_on_warnings': True}
cnx = MySQL.connect(**config)
cnx.close()
```

ตัวอย่างที่ 9.1 คำสั่ง `import mysql.connector as MySQL` เป็นการขอใช้โมดูล `mysql.connector` และตั้งชื่อว่า `MySQL` ตัวแปร `config` มีชนิดดิกชันนารี เก็บคีย์และค่าข้อมูลของ `user`, `password`, `host`, `database` และ `raise_on_warnings` ตัวแปร `cnx` เป็นอ็อบเจกต์การเชื่อมต่อโดยส่งตัวแปร `config` เข้าไปเป็นอาร์กิวเมนต์ของฟังก์ชัน `connect()` เมื่อเครื่องหมาย `**` คือ การป้อนอาร์กิวเมนต์ให้ฟังก์ชันด้วยตัวแปรดิกชันนารี หลังจากเชื่อมต่อสำเร็จจะทำการปิดการเชื่อมต่อด้วยคำสั่ง `cnx.close()`

9.5 การเพิ่มข้อมูลด้วยคำสั่ง Insert

ตัวอย่างที่ 9.2 การเพิ่มข้อมูลในตาราง 41410606_friend

```
import mysql.connector as MySQL
config = {'user': "i_user", 'password': "123456", 'host': "202.28.34.204",
'database': "i_1202201", 'raise_on_warnings': True}
cnx = MySQL.connect(**config)
cursor = cnx.cursor()
Ds = {'firstname': 'จักรกฤษณ์',
'lastname': 'แสงแก้ว',
'email': "chakkrit@msu.ac.th" }
placeholders = ', '.join(['%s'] * len(Ds))
columns = ', '.join("'" + str(x).replace('/', '_') + "'" for x in Ds.keys())
values = ', '.join("'" + str(x).replace('/', '_') + "'" for x in Ds.values())
sql = "INSERT INTO %s ( %s ) VALUES ( %s );" % ('41410606_friend', columns, values)
cursor.execute(sql)
cnx.commit()
cnx.close()
```

ตัวอย่างที่ 9.2 การเพิ่มข้อมูลจะเตรียมข้อมูลในตัวแปร `Ds` เป็นชนิดดิกชันนารี และจะมีคีย์ประกอบด้วย `firstname`, `lastname` และ `email` ซึ่งตรงกับคอลัมน์ของตารางที่สร้างขึ้น ตัวแปร `placeholders` จะทำหน้าที่เชื่อมคีย์และค่าข้อมูลและสังเคราะห์คำสั่ง SQL จากตัวแปรดิกชันนารี

จากนั้น คำสั่ง `cursor.execute(sq)` จะนำคำสั่ง SQL ที่อยู่ภายในตัวแปร `sql` ไปประมวลผลบน MySQL Server หลังจากนั้นใช้คำสั่ง `commit()` เพื่อยืนยันการเปลี่ยนแปลงและปิดการเชื่อมต่อด้วยคำสั่ง `close()`

9.6 การสืบค้นด้วยคำสั่ง Select

การสืบค้นข้อมูลภายในตารางใช้คำสั่ง SQL ด้วยประโยค SELECT ดังนี้

ตัวอย่างที่ 9.3 การสืบค้นข้อมูลในตาราง 41410606_friend

```
import mysql.connector as MySQL
config = {'user': "i_user", 'password': "123456", 'host': "202.28.34.204",
'database': "i_1202201", 'raise_on_warnings': True}
cnx = mysql.connector.connect(**config)
cursor = cnx.cursor()
sql = "SELECT * FROM 41410606_friend"
cursor.execute(sql)
for i in cursor:
    print(i)
cursor.close()
cnx.close()
```

ผลลัพธ์ :

```
(1, 'จักรกฤษณ์', 'แสงแก้ว', 'chakkrit@msu.ac.th')
```

ตัวอย่างที่ 9.3 การสืบค้นข้อมูลภายในตาราง 41410606_friend ด้วยคำสั่ง "SELECT * FROM 41410606_friend" โดยคำสั่ง `sql` จะถูกประมวลผลบน MySQL Server และผลลัพธ์ที่ได้เก็บไว้ในตัวแปร `cursor` เมื่อวนเข้าไปใน `cursor` จะหมายถึงจำนวนแถวแต่ละแถวที่เป็นผลลัพธ์ ในตัวอย่างนี้มีเพียง 1 เรคคอร์ด หลังจากนั้นทำการปิดการสื่อสารด้วยคำสั่ง `close()`

9.7 การแก้ไขข้อมูล

การแก้ไขข้อมูลใช้คำสั่ง UPDATE ดังนี้

ตัวอย่างที่ 9.4 การแก้ไขข้อมูลด้วยคำสั่ง Update

```
import mysql.connector as MySQL
config = {'user': "i_user", 'password': "123456", 'host': "202.28.34.204",
'database': "i_1202201", 'raise_on_warnings': True}
cnx = mysql.connector.connect(**config)
cursor = cnx.cursor()
sql = "UPDATE 41410606_friend SET email = 'electoday@gmail.com' WHERE firstname =
'จักรกฤษณ์' AND lastname = 'แสงแก้ว'"

```

```

cursor.execute(sql)
for i in cursor:
    print(i)
cursor.close()
cnx.close()

```

การอัปเดตข้อมูลใช้คำสั่ง SQL ด้วยรูปแบบคำสั่ง "UPDATE 41410606_friend SET email = 'electoday@gmail.com' WHERE firstname = 'จักรกฤษณ์' AND lastname = 'แสงแก้ว' " คือการแก้ไข email เป็นคำว่า "chakkrit@gmail.com" เมื่อ firstname ตรงกับคำว่า "จักรกฤษณ์" และ lastname ตรงกับคำว่า "แสงแก้ว"

9.8 การลบข้อมูล

การลบข้อมูลใช้คำสั่ง DELETE ดังนี้

ตัวอย่างที่ 9.5 การลบข้อมูลด้วยคำสั่ง Delete

```

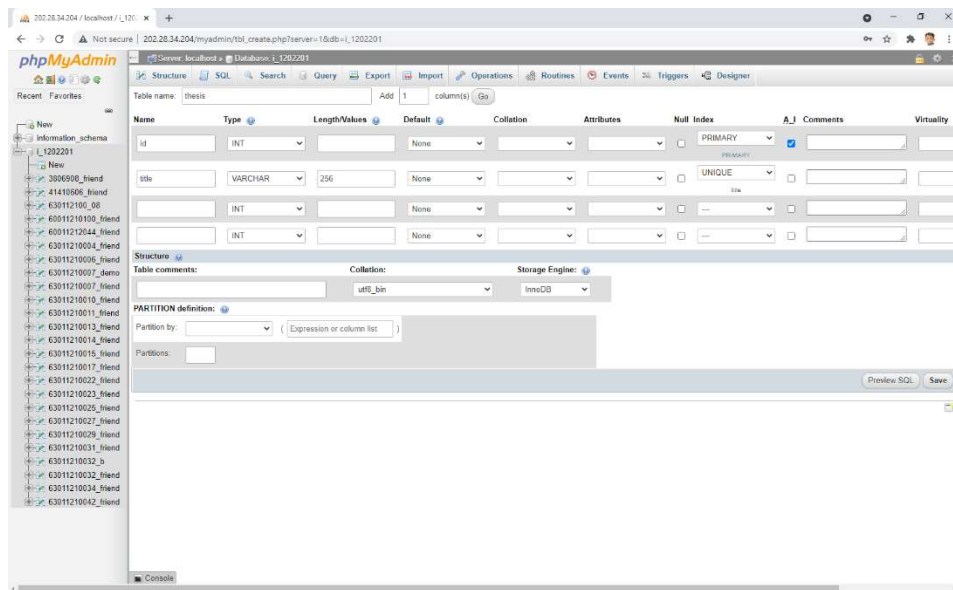
import mysql.connector as MySQL
config = {'user': "i_user", 'password': "123456", 'host': "202.28.34.204",
'database': "i_1202201", 'raise_on_warnings': True}
cnx = mysql.connector.connect(**config)
cursor = cnx.cursor()
sql = "DELETE FROM friend WHERE email = 'chakkrit@gmail.com' "
cursor.execute(sql)
cursor.close()
cnx.close()

```

การลบข้อมูลภายในตารางใช้คำสั่ง "DELETE FROM friend WHERE email = 'chakkrit@gmail.com' " เป็นการลบแถวที่มี email ตรงกับคำว่า "chakkrit@gmail.com" ทิ้งไป

9.9 การนำข้อมูลวิทยานิพนธ์ 10,000 รายการใส่ลงในฐานข้อมูล

ขั้นที่ 1: สร้างตารางชื่อ thesis มีโครงสร้างข้อมูล id เป็น number auto increment และ title เป็น varchar ขนาด 200 ตัวอักษร ดังต่อไปนี้



ภาพประกอบที่ 9.5 การสร้างตารางวิทยานิพนธ์ thesis

9.10 การนำข้อมูลวิทยานิพนธ์ใส่เข้าไปในตาราง thesis

ตัวอย่างที่ 9.6 การนำข้อมูลวิทยานิพนธ์และใส่ในตาราง thesis

```
import urllib, urllib.request
url = "http://dsdi.msu.ac.th/programming/thailis-10000.txt"
s = urllib.request.urlopen(url).read().decode('utf-8', 'ignore').split("\n")
import mysql.connector as MySQL
config = {'user': 'i_user', 'password': '123456', 'host': '202.28.34.204',
'database': 'i_1202201', 'raise_on_warnings': True}
cnx = MySQL.connect(**config)
cursor = cnx.cursor()
for i in s:
    sql = "INSERT INTO thesis (title) VALUES(\"%s\")"%i
    cursor.execute(sql)
    cnx.commit()
cnx.close()
```

ตัวอย่างที่ 9.6 การอ่านไฟล์ thailis-10000.txt จากเว็บ dsdi.msu.ac.th จากนั้นทำการสกัดรายชื่อวิทยานิพนธ์ด้วยคำสั่ง `split("\n")` ซึ่งเป็นการแยกด้วยเครื่องหมายขึ้นต้นบรรทัดใหม่ หลังจากเชื่อมต่อเข้ากับ Mysql Server ได้แล้วจะสร้างคำสั่ง SQL ด้วยข้อความ `"INSERT INTO thesis (title) VALUES (\"%s\")"%i` เมื่อตัวแปร `i` คือ รายชื่อวิทยานิพนธ์ในแต่ละเรื่อง หลังจากนั้นทำการ `execute()` และทำการ `commit()` เมื่อดำเนินการเสร็จสิ้นแล้วทำการปิดการสื่อสารด้วยคำสั่ง `close()`

แบบฝึกหัดท้ายบท

1. จงอธิบายการสร้างตารางด้วย phpmyadmin
2. จงอธิบายการเพิ่มข้อมูลในตารางด้วยภาษาไพธอน
3. จงอธิบายการลบข้อมูลในตารางด้วยเงื่อนไขที่กำหนด
4. จงอธิบายการสืบค้นข้อมูลภายในตารางและแสดงผลด้วยภาษาไพธอน
5. จงอธิบายการแก้ไขข้อมูลภายในตารางด้วย SQL และภาษาไพธอน
6. จงเขียนโปรแกรมเพื่อเพิ่ม ลบ แก้ไข และสืบค้น โดยตั้งชื่อตารางรหัสนิสิต และประกอบด้วย
firstname lastname phone และ nickname

บทที่ 10

การเขียนโปรแกรมภาษาไพธอนในงานภูมิสารสนเทศ

แนวคิด

ภูมิสารสนเทศ (Geoinformatics) เป็นสาขาที่เกี่ยวข้องกับการทำแผนที่และการวิเคราะห์ข้อมูลทางพื้นที่ โดยใช้ข้อมูลจากเทคโนโลยีขั้นสูง อาทิ การรับรู้จากระยะไกล (Remote Sensing) เช่น ภาพถ่ายดาวเทียม ภาพเรดาร์ เป็นต้น ข้อมูลที่ได้จากการรับรู้ระยะไกล (Remote Sensing) สามารถนำไปใช้ติดตามสภาพแวดล้อม เหตุการณ์ภัยพิบัติ อุทกภัย ไฟป่าและอื่น ๆ ในตอนท้ายของบทนี้จะนำเสนอการสกัดข้อมูลจากเอกสารประกาศราชกิจจานุเบกษาซึ่งเป็นเอกสารที่ไม่มีโครงสร้างจากนั้นนำมาจัดเก็บในไฟล์แผนที่และแสดงผลตามระดับการจัดสรรงบประมาณประจำปี 2564

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ ระบบพิกัดภูมิศาสตร์ (Geographic Coordinate System - GCS) ระบบพิกัดกริด (Grid Coordinate) หรือ UTM ซอฟต์แวร์ในงานภูมิสารสนเทศ ความรู้ทั่วไปเกี่ยวกับโปรแกรม Quantum GIS การเพิ่มแผนที่จังหวัดและอำเภอด้วย QGIS การเขียนโปรแกรมภาษาไพธอนสกัดรายชื่อจังหวัดที่อยู่ภายในพื้นที่รัศมีเรดาร์ดาวเทียมและอ้อมก้อย การประยุกต์ใช้ Folium และ GeoPandas

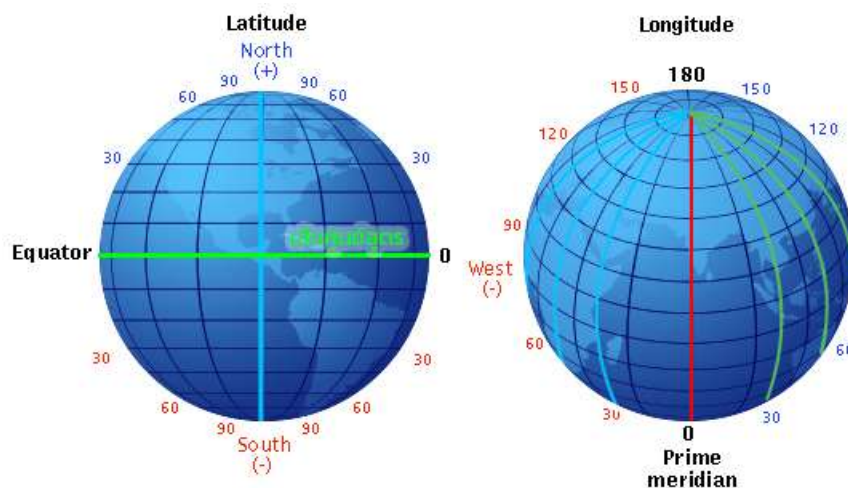
วัตถุประสงค์

1. อธิบายการบอกพิกัดตำแหน่งด้วย UTM ได้
2. อธิบายการเพิ่มและลบข้อมูลเวกเตอร์ลงบนแผนที่ได้
3. อธิบายการตรึงพิกัดของภาพด้วย Georeferencing ได้
4. อธิบายการนำเสนอด้วย Folium ได้
5. อธิบายการจัดการข้อมูลแผนที่ด้วย GeoPandas ได้
6. อธิบายการสกัดข้อมูลงบประมาณประจำปี 2564 ใส่ลงใน Shape ของจังหวัดได้

10.1 ระบบพิกัดในแผนที่ (Coordinate System)

ระบบพิกัดใช้สำหรับอ้างอิงถึงตำแหน่งวัตถุ มีอยู่หลายระบบ ได้แก่

1. ระบบพิกัดภูมิสารสนเทศ (Geographic Coordinate System - GCS) บอกตำแหน่งด้วยค่าองศา ลิปดา และฟิลิปดา โดยมีจุดกำเนิดของที่ผ่านเมืองกรีนวิช (Greenwich) ประเทศอังกฤษ เรียกเส้นเมริเดียนแรก (Prime meridian) การบอกตำแหน่งเหมือนกับ ละติจูดต่างกันที่ต้องบอกซีกโลกด้วย เช่น ลองจิจูดที่ 90 องศา 20 ลิปดา 45 ฟิลิปดาตะวันตก เป็นต้น
2. ระบบพิกัด UTM (Universal Transverse Mercator) พัฒนาโดยกองทัพสหรัฐอเมริกาและนำมาใช้ในปี ค.ศ. 1946 เพื่อให้ได้แผนที่ที่มีความถูกต้องยิ่งขึ้น บอกตำแหน่งบนพื้นผิวโลกด้วย 2 ค่า คือ 1) ลองจิจูด (longitude) 2) แลติจูด (latitude) โดยอ้างอิงกับเส้นเมริเดียนหรือเส้นแวนดิง (meridian) และเส้นศูนย์สูตรหรือเส้นแนวราบ (Equator)
3. ระบบ GLO (General Land Office grid system) สำหรับงานด้านธรณีวิทยา



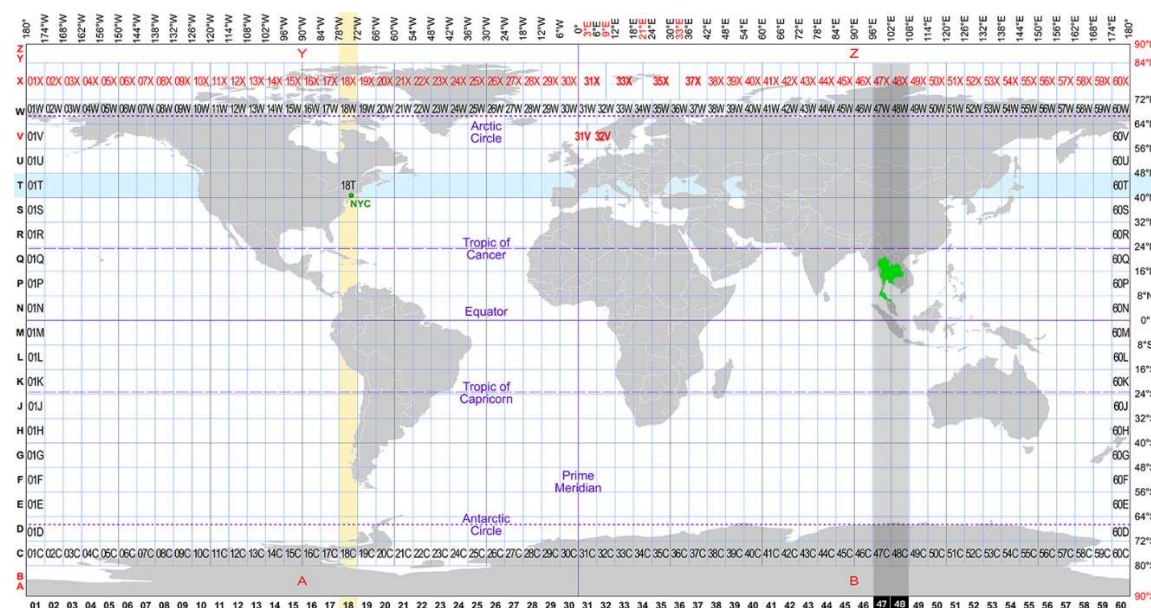
ภาพประกอบที่ 10.1 แสดงแนวตั้งเมริเดียน (Meridian) และเส้นศูนย์สูตร (Equator)

10.2 ระบบพิกัดกริด (Grid Coordinate) หรือ UTM (Universal Transverse Mercator)

ระบบพิกัดภูมิศาสตร์ที่ใช้ในประเทศไทยแสดงได้ดังตารางต่อไปนี้

ตารางที่ 10.1 ระบบพิกัดภูมิศาสตร์ที่ใช้ในประเทศไทย

ระบบอ้างอิง	โซน (Zone)	พื้นหลักฐาน (datum)	รหัส EPSG
Geographic	-	Indian 1975	4240
Geographic	-	WGS 1984	4326
UTM	47	Indian 1975	24047
UTM	48	Indian 1975	24048
UTM	47	WGS 1984	32647
UTM	48	WGS 1984	32648



ภาพประกอบที่ 10.2 พิกัดกริด (Grid Coordinate) หรือ UTM(Universal Transverse Mercator)

10.3 โปรแกรม Quantum GIS (QGIS)

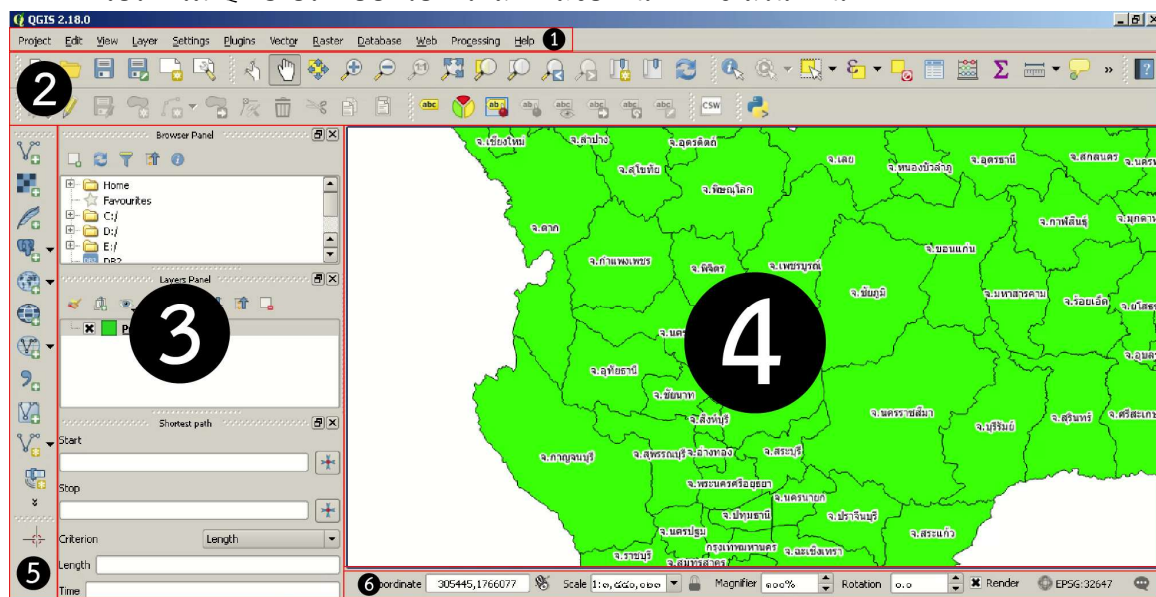
ในปี พ.ศ. 2545 โปรแกรม QGIS พัฒนาขึ้นโดยนักพัฒนาชาวเยอรมันทำงานบนระบบปฏิบัติการ Windows, Linux และ Mac OS สามารถใช้งานได้ไม่เสียค่าใช้จ่าย ดาวน์โหลดได้ที่ <https://qgis.org>

ส่วนประกอบของ QGIS ได้แก่ 1) QGIS Desktop ใช้สำหรับเพิ่ม ลบ แก้ไข วิเคราะห์ นำเสนอ ข้อมูลภูมิสารสนเทศ 2) QGIS Browser ใช้ตรวจสอบข้อมูลและเมตาดาต้าของข้อมูลภูมิสารสนเทศ 3) QGIS Server ใช้จัดการ WMS (Web Map Service) และ WFS (Web Feature Service) เพื่อควบคุม ชั้นข้อมูลและเลย์เอาต์สำหรับแสดงผล และ 4) QGIS Web Client ใช้แสดงผลผ่านเว็บ

ข้อมูลที่ใช้ภายในโปรแกรมภูมิศาสตร์ ได้แก่ 1) เวกเตอร์ (Vector) เก็บในรูปแบบพิกัดและความสัมพันธ์ระหว่างพิกัด แบ่งเป็น ก) จุด (point) ข) เส้น (line) ค) รูปหลายเหลี่ยม (polygon) 2) ข้อมูลราสเตอร์ (Raster) ใช้เก็บข้อมูลภาพ เช่น ภาพถ่ายทางอากาศ ภาพถ่ายดาวเทียม เป็นต้น 3) ข้อความที่มีเครื่องหมายคั่น (Delimited Text)

10.4 การติดต่อผู้ใช้ด้วยภาพกราฟิก (GUI : Graphic User Interface)

โปรแกรม QGIS ประกอบด้วยการแสดงผลของหน้าต่าง 6 ส่วน ดังนี้



ภาพประกอบที่ 10.3 ส่วนประกอบของโปรแกรม Qgis

- 1) แถบเมนู (Menu bar) คือ แถบคำสั่งทั้งหมดของโปรแกรม
- 2) แถบเครื่องมือ (Tool bar) คือ คำสั่งที่ใช้งานบ่อยมีลักษณะเป็นกลุ่มและหมวดหมู่เดียวกัน
- 3) หน้าต่างแสดงชั้นข้อมูล (Map legend) คือ คำอธิบายแผนที่และเปิด-ปิดเลเยอร์
- 4) แผนที่ (Map view) คือ การแสดงผลแผนที่ทั้งหมด
- 5) แถบเพิ่มลบเลเยอร์ (Manage Layer) คือ การเพิ่มลบชั้นแผนที่
- 6) แถบแสดงสถานะ (Status bar) คือ การแสดงข้อมูลต่าง ๆ อาทิ พิกัดที่เลื่อนเมาส์บนแผนที่ มาตราส่วน และระบบพิกัด เป็นต้น

หมายเหตุ: กำหนด WGS84 Zone 47N ทุกครั้งเมื่อเปิดโปรแกรม QGIS กำหนดได้โดย เลือก settings -> Options -> CRS Tab -> Use a default CRS : เลือก EPSG:32647-WGS 84/UTM zone 47N

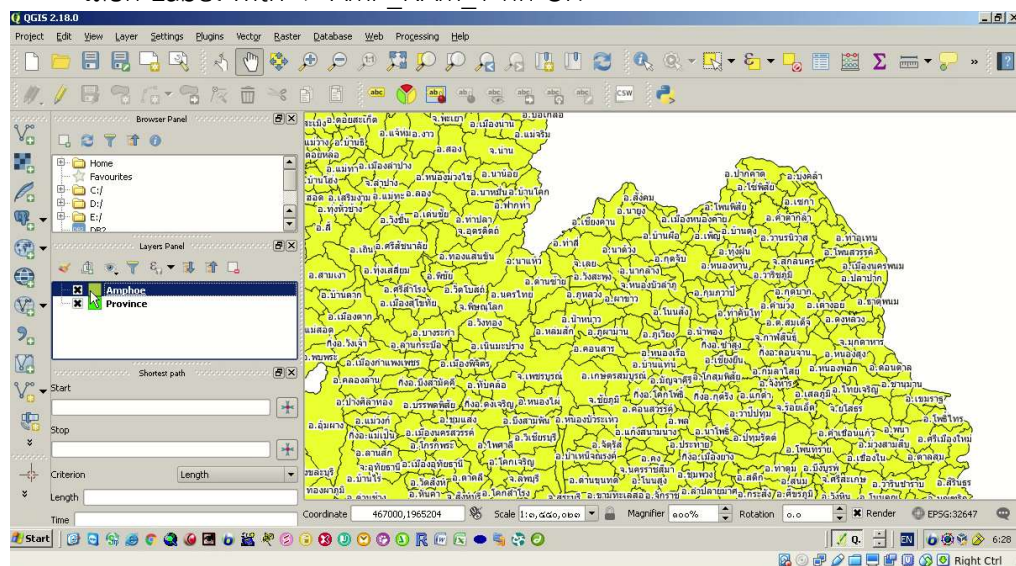
10.5 การเพิ่มแผนที่จังหวัดและแสดงรายชื่อด้วย QGIS

การเพิ่มแผนที่จังหวัดและแสดงรายชื่อด้วย QGIS มีขั้นตอนดังนี้

1. ดาวน์โหลดแผนที่จังหวัดที่ลิงค์ data/articles/programming/gis/shape/Province.zip จากนั้นแยกไฟล์ไว้ในโฟลเดอร์
2. เลือกเมนู Layer -> Add Layer -> Add Vector Layer -> เลือกไฟล์ Province.shp ที่ Layer panel จะเห็นว่ามีเลเยอร์เพิ่มขึ้นมา ชื่อ Province ดังภาพต่อไปนี้
3. เลือกที่เลเยอร์ Province ที่แท็บ Labels ให้เลือก Label with -> PROV_NAM_T กด OK ผลลัพธ์แสดงดังภาพที่ 10.3

10.6 เพิ่มข้อมูลอำเภอลงบนแผนที่ด้วย QGIS

1. ดาวน์โหลดแผนที่อำเภอที่ลิงค์ data/articles/programming/gis/shape/ Amphoe.zip จากนั้นแยกไฟล์ไว้ในโฟลเดอร์
2. เลือกเมนู Layer -> Add Layer -> Add Vector Layer -> เลือกไฟล์ Amphoe.shp ที่ Layer panel จะเห็นว่ามีเลเยอร์เพิ่มขึ้นมา ชื่อ Amphoe ดังภาพต่อไปนี้
3. แสดงข้อความรายชื่ออำเภอบนแผนที่โดยการเลือกที่เลเยอร์ Amphoe ที่แท็บ Labels ให้เลือก Label with -> AMP_NAM_T กด OK



ภาพประกอบที่ 10.4 แสดงการเพิ่มข้อมูลแผนที่อำเภอ

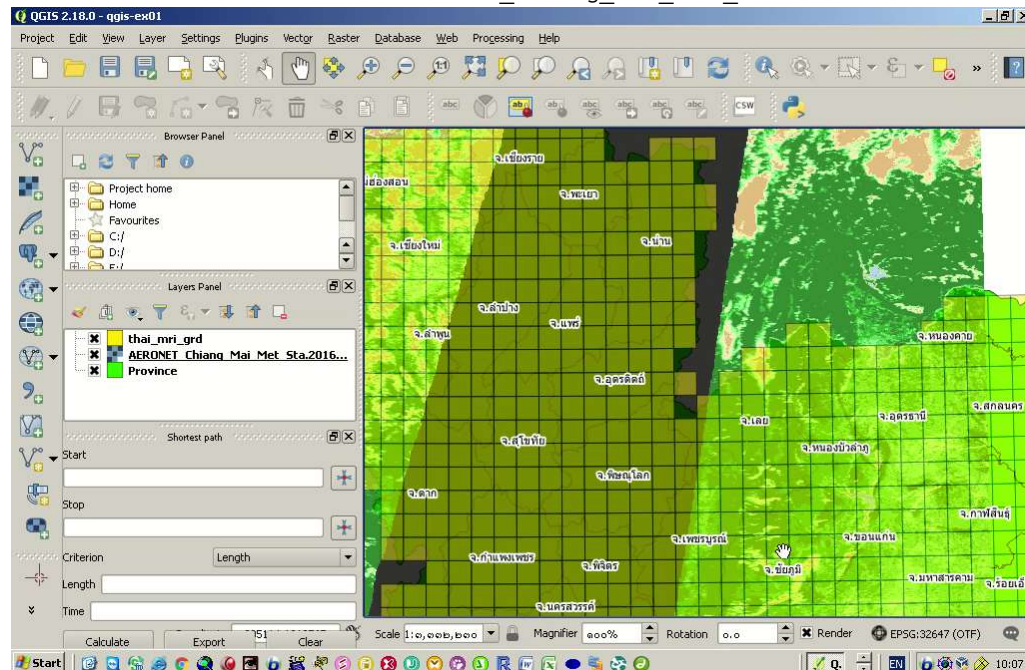
10.7 การนำเข้าข้อมูลภาพถ่ายดาวเทียมพื้นที่ประเทศไทยจากดาวเทียมโมดิส (MODIS)

ภาพถ่ายดาวเทียม MODIS มีแถบการถ่ายภาพที่กว้างครอบคลุมพื้นที่ประเทศไทยด้วยการถ่ายภาพเพียงครั้งเดียว มีความละเอียด 250- 1,000 เมตร สามารถใช้งานภาพถ่ายได้ที่เว็บองค์การนาซ่า

https://lance.modaps.eosdis.nasa.gov/imagery/subsets/?subset=AERONET_Chiang_Mai_Met_Sta

» ตัวอย่างภาพถ่ายดาวเทียม : AERONET_Chiang_Mai_Met_Sta.2016318.aqua.721.250m

» ตัวอย่างภาพถ่ายดาวเทียม : AERONET_Chiang_Mai_Met_Sta.2016318.terra.ndvi.250m



ภาพประกอบที่ 10.5 การเพิ่มชั้นภาพถ่ายดาวเทียม MODIS

10.8 การสกัดรายชื่อจังหวัดที่อยู่ภายในพื้นที่รัศมีเรดาร์ตาคลีและอมก๋อย

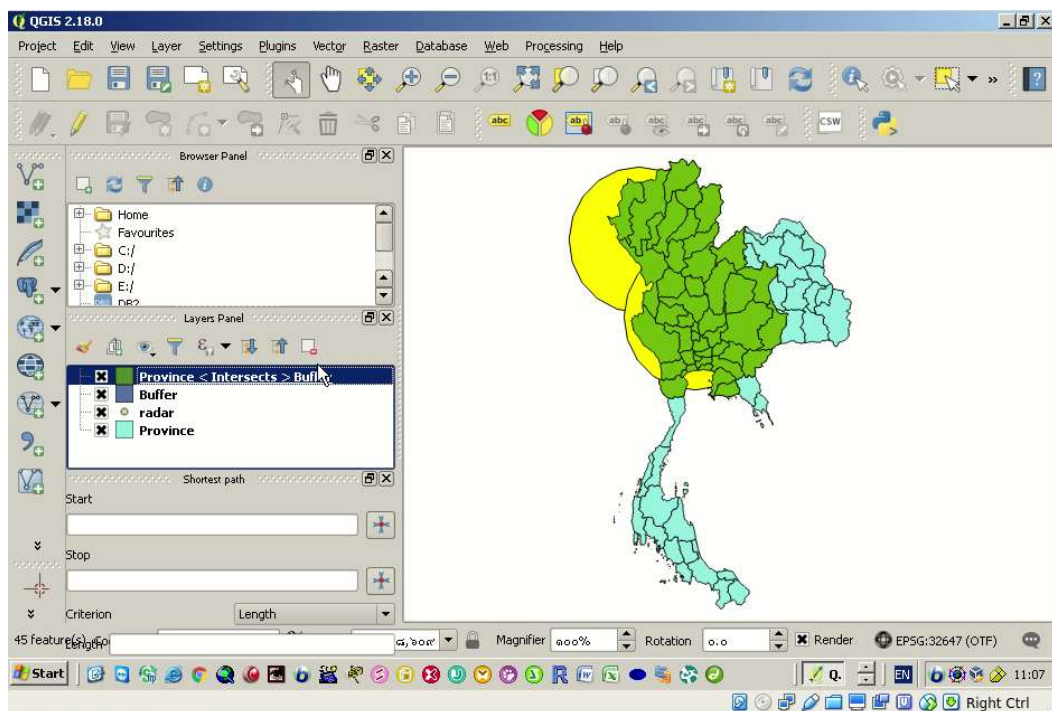
ในหัวข้อนี้จะเขียนโปรแกรมภาษาไพธอนเพื่อกรองรายชื่อจังหวัดที่อยู่ภายในพื้นที่รัศมีของเรดาร์ตาคลีและอมก๋อย ซึ่งมีระยะทางครอบคลุมรัศมี 240 กิโลเมตร จังหวัดนครสวรรค์และเชียงใหม่ โดยจะทำการรวมพื้นที่รัศมี 240 กิโลเมตรจากสถานีเรดาร์ทั้งสองสถานี

1. สร้างไฟล์ radar.txt ประกอบด้วยข้อมูลต่อไปนี้

STATION	LAT	LON
OMKOI	17.798229	98.432477
TAKLI	15.250701	100.336763

2. เลือก Layer -> Add Layer -> Add Delimited Text Layer -> เลือกไฟล์ radar.txt จากขั้นตอนที่ 1 -> ที่ File format ให้เลือก CSV (Comma separated values) เลือก Custom delimiters -> กด OK

3. เมนู Vector -> Geoprocessing Tools -> Fixed distance buffer -> เลือก Layer : Radar [EPSG:4326] จากที่ตั้ง Distance เท่ากับ 2.4 ซึ่งมีค่า 240 กิโลเมตร กำหนด segment เท่ากับ 5 -> กด OK ผลลัพธ์จะแสดงขอบเขตของเรดาร์ ดังภาพต่อไปนี้



ภาพประกอบที่ 10.6 แสดงขอบเขตรัศมีเรดาร์ดาวเทียมและอ้อมกอด

4. เลือกเลเยอร์ Province คลิกขวา -> Open Attribute Table กดที่มุมบนด้านซ้ายของตาราง ดังที่ลูกศรชี้ในภาพด้านล่าง

	AREA	PROV_CODE	PROV_NAM_T	PROV_NAM_E
1	1566623475.366	10	กรุงเทพมหานคร	Bangkok
2	967116164.666	11	จ.สมุทรปราการ	Changwat Samut...
3	636400456.390	12	จ.นนทบุรี	Changwat Nonth...
4	1520800567.615	13	จ.ปทุมธานี	Changwat Pathu...
5	2547347739.987	14	จ.พระนครศรีอยุธยา	Changwat Phra ...
6	950496874.168	15	จ.อ่างทอง	Changwat Ang T...
7	6502715463.332	16	จ.ลพบุรี	Changwat Lopburi
8	817006565.890	17	จ.สิงห์บุรี	Changwat Singburi
9	2505262595.493	18	จ.ชัยนาท	Changwat Chainat
10	3488177074.439	19	จ.สระบุรี	Changwat Saraburi
11	4507781009.190	20	จ.ชลบุรี	Changwat Chonb...
12	3665288460.886	21	จ.ระยอง	Changwat Rayong
13	6373260853.424	22	จ.ฉะเชิงเทรา	Changwat Chant...
14	2867269656.336	23	จ.ตราด	Changwat Trat
15	5169824096.890	24	จ.ฉะเชิงเทรา	Changwat Chach...

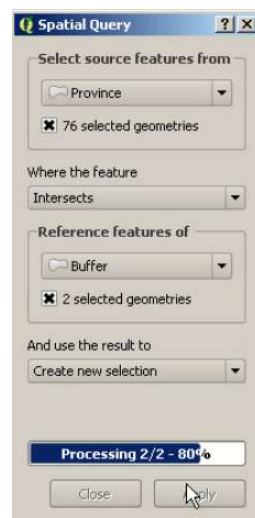
ภาพประกอบที่ 10.7 แสดงการเลือกรายการแอตทริบิวต์ของชั้น province

5. เลือกเลเยอร์ Buffer -> คลิกขวา -> Open Attribute Table จากนั้นกดเลือกที่มุมบนด้านซ้ายของตาราง ดังภาพต่อไปนี้

	STATION	LAT	LON
1	OMKOI	17.79823	98.43248
2	TAKLI	15.25070	100.33676

ภาพประกอบที่ 10.8 แสดงการเลือกรายการแอตทริบิวต์ของชั้น Buffer

6. เลือกเมนู Vector -> Spatial Query -> Spatial Query เลือก Source Feature From: Province และเลือก และเลือก Reference features of: Buffer จากนั้นกดปุ่ม Apply



ภาพประกอบที่ 10.9 แสดงจังหวัดที่อยู่ในพื้นที่ intersec ระหว่าง province และ buffer

7. เลือกที่เลเยอร์ Province <intersects> Buffer จากนั้นเลือกเมนู Plugins -> Python Console เขียนโค้ดดังนี้

ตัวอย่างที่ 10.1 การแสดงรายชื่อจังหวัดในเรดาร์ตาลีและอ้อมก้อย

```
Layer=iface.activeLayer()
for i in Layer.getFeatures():
    print i['PROV_NAM_T']
```

ผลลัพธ์

```
จ.เลย
จ.เชียงใหม่
จ.ลำพูน
...
จ.สมุทรสงคราม
จ.เพชรบุรี
```

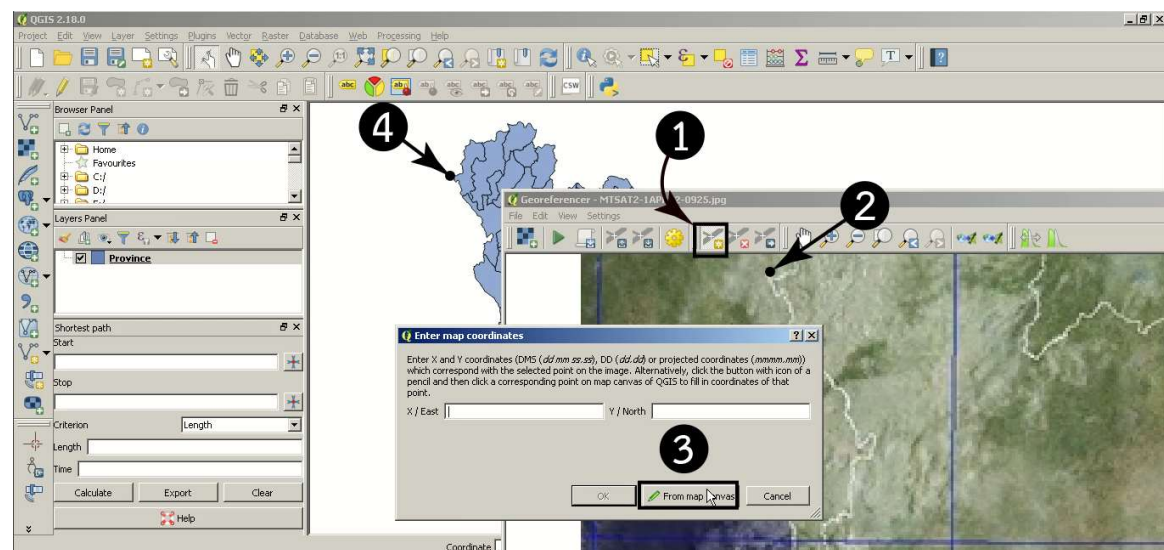
ตัวอย่างที่ 10.1 ตัวแปร Layer เป็นชั้น layer ปัจจุบันที่ถูกเลือกให้ทำงานซึ่งคือชั้น Province <intersects> Buffer จากนั้นวนเข้าไปใน Features และแสดงค่าตัวแปร PROV_NAM_T ออกมา ผลลัพธ์จะแสดงรายชื่อจังหวัดที่อยู่ภายในพื้นที่รัศมีเรดาร์ตามคี่และอ้อมก้อยในรัศมี 240 กิโลเมตร

10.9 การตรึงภาพถ่ายและแผนที่ด้วย Georeference

การตรึงภาพ หรือ Georeference คือ การกำหนดพิกัดให้กับภาพ อาจจะเป็นภาพถ่ายมุมสูงหรือภาพถ่ายจากดาวเทียมเพื่ออ้างอิงตำแหน่งภายในภาพให้ตรงกับพิกัดบนแผนที่ได้อย่างถูกต้อง โดยปกติภาพถ่ายดาวเทียมที่ให้บริการมักผ่านกระบวนการทำ Georeferencing มาแล้ว

ให้นิสิตดาวน์โหลดภาพถ่ายดาวเทียมจากลิงค์ <http://dsdi.msu.ac.th/programming/gis/MTSAT2-1APR12-0925.jpg> และทำ Georeference

1. เปิดโปรแกรม QGIS เลือก Setting -> Option -> CRS -> เลือก WGS84/UTM Zone 47N
2. ดาวน์โหลดภาพถ่ายดาวเทียม MTSAT2 (นามสกุล .jpg)
3. เปิด QGIS เลือก Plugins -> Georeferencer -> Georeferencer
4. เลือกเมนู Raster -> Georeferencer -> เลือก Open Raster -> เลือกภาพที่ต้องการตรึงในตัวอย่างนี้คือไฟล์ MTSAT2-1APR12-0925.jpg
5. เมนู Setting -> Transformation setting -> เลือก Target SRS เป็น WGS 84/UTM zone 47N จากนั้นคลิกที่จุดที่ต้องการตรึงในภาพ
6. เลือกที่ปุ่มเพิ่มจุดอ้างอิงที่ 1 จากนั้นเลือกลงไปจุดบนภาพ จะให้ป้อนพิกัด ให้เลือกปุ่ม From map canvas จากนั้นคลิกที่จุดอ้างอิงบนแผนที่ ทำซ้ำจนครอบคลุมพื้นที่ภาพถ่าย
7. บันทึกไฟล์ผลลัพธ์เป็นนามสกุล .TIF



ภาพประกอบที่ 10.10 การตรึงภาพถ่ายดาวเทียม MSAT2 ด้วย Georeferencing

10.10 การแสดงแผนที่ด้วย folium

โมดูล Folium สำหรับแสดงแผนที่ในรูปแบบไฟล์ Html สามารถใส่ marker และกำหนดข้อมูล การแสดงแผนที่ รองรับแผนที่จาก OpenStreetMap, Mapbox และ Stamen รองรับ GeoJSON และ TopoJSON ติดตั้งด้วยคำสั่ง pip install folium หรือจากเว็บ <https://pypi.org/project/folium>

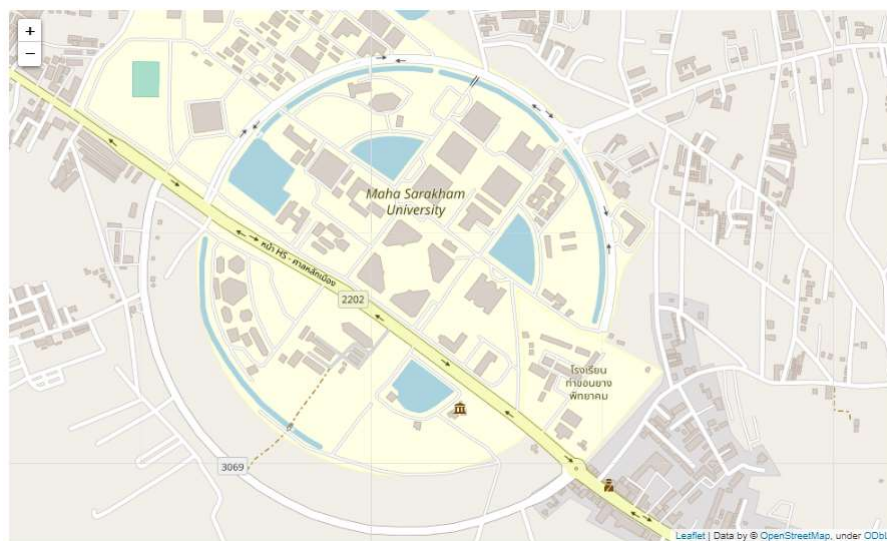
Folium ทำงานบน leaflet.js ซึ่งเป็น javascript เกี่ยวกับแผนที่บนสมาร์ตโฟนและคอมพิวเตอร์ ซึ่งมีลักษณะคล้าย Openlayers แต่มีจุดเด่นคือแสดงผลบนสมาร์ตโฟนและมีขนาดเพียง 33 KB การระบุตัวแปร tiles มีค่าดังนี้ ถ้าไม่ระบุจะหมายถึง OpenStreetMap

OpenStreetMap: OSM เป็นฐานข้อมูลแผนที่เปิด ซึ่งอนุญาตให้ผู้ใช้แก้ไขข้อมูลได้ นักพัฒนาสามารถนำข้อมูลไปใช้ได้โดยไม่เสียค่าใช้จ่าย ข้อเสียของ OpenStreetMap มีรายละเอียดน้อยกว่าแผนที่อื่น ๆ และไม่มีภาพถ่ายดาวเทียม

ตัวอย่างที่ 10.2 การพล็อตตำแหน่งพิกัด 16.2453, 103.2514 และอัตราการขยายภาพที่ 15

```
import folium
lat = 16.245327039142097
lng = 103.25143414503279
folium.Map(location=[lat,lng], zoom_start=15)
```

ผลลัพธ์



ภาพประกอบที่ 10.11 แสดงแผนที่ที่พิกัดตำแหน่งมหาวิทยาลัยมหาสารคาม

ตัวอย่างที่ 10.2 ขอใช้ไลบรารี folium ด้วยคำสั่ง `import folium` ประกาศตำแหน่งพิกัดในตัวแปร `lat` และ `lng` จากนั้นใช้คำสั่ง `Map()` ในการพล็อตพิกัดลงบนแผนที่ โดยมีพารามิเตอร์ `location` และ `zoom_start` เท่ากับ 15

ตารางที่ 10. 2 ตารางพารามิเตอร์ Folium

Folium	อธิบาย
geo_data	พาร์ทไปยังไฟล์ geoJSON ซึ่งมีพิกัดของพื้นที่
Data	คือข้อมูลใช้งาน
columns	คอลัมน์ที่ใช้สร้างแผนที่ choropleth

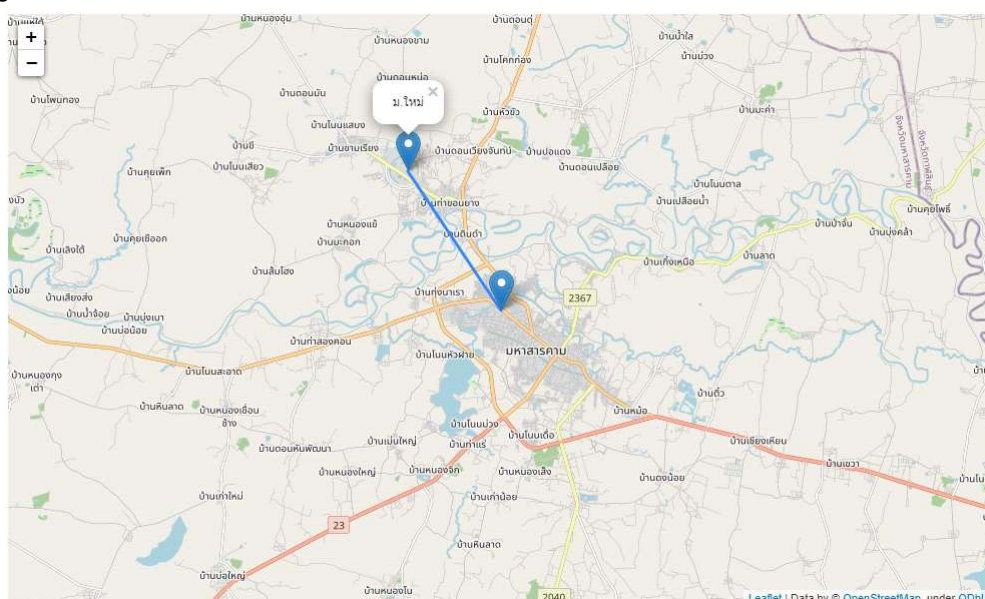
Folium	อธิบาย
key_on	ชื่อไฟล์ json ที่ต้องการแสดงบนแผนที่
fill_color	สีที่ระบายลงบนแผนที่ เช่น BuPu (Blue Purple)
fill_opacity	ความโปร่งของพื้นที่ มีค่าระหว่าง 0-1 ค่าเริ่มต้นมีค่า 0.6
line_opacity	ความโปร่งของเส้น มีค่าระหว่าง 0-1 ค่าเริ่มต้นมีค่า 1
legend_name	คำอธิบายแผนภูมิแสดงผลบนกรอบสี่เหลี่ยมด้านบนขวาของแผนที่
location	ตำแหน่งละติจูดและลองจิจูดของแผนที่
zoom_start	ระดับการขยายแผนที่ขณะเริ่มต้นการแสดงผล
smooth_factor	ค่าความต่อเนื่องในแต่ละระดับการย่อ/ขยาย

ตัวอย่างที่ 10.3 การเพิ่มเส้นลงในแผนที่

```
import folium
msu1 = [16.199662313316438, 103.28357298658125]
msu2 = [16.245327039142097, 103.25143414503279]
mp = folium.Map(location = [msu1[0], msu1[1]], zoom_start = 12)
folium.Marker([msu1[0], msu1[1]], popup = 'ม.เก่า').add_to(mp)
folium.Marker([msu2[0], msu2[1]], popup = 'ม.ใหม่').add_to(mp)

folium.PolyLine(locations = [(msu1[0], msu1[1]), (msu2[0], msu2[1])],
line_opacity = 0.5).add_to(mp)
mp
```

ผลลัพธ์



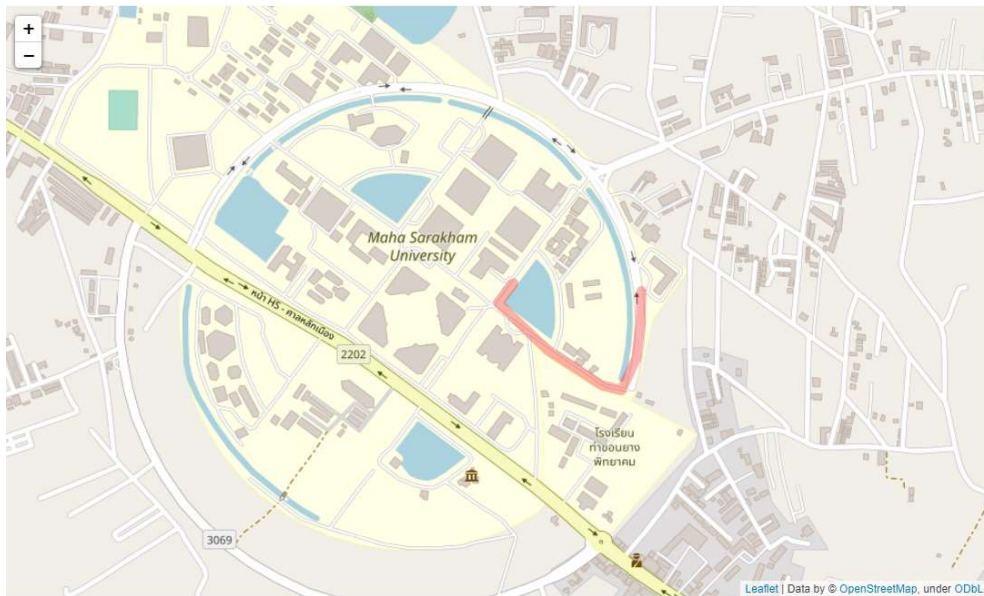
ภาพประกอบที่ 10.12 แสดงการวาดเส้นตรงระหว่าง มมส.เก่าและ มมส.ใหม่

ตัวอย่างที่ 10.3 การวาดเส้นตรงระหว่างพิกัดจุดสองจุดและสร้าง popup เมื่อคลิกที่จุด marker จะแสดงรายละเอียดของตำแหน่งนั้น โดยตัวแปร msu1 และ msu2 คือ พิกัดละติจูดและลองจิจูด ม.เก่า และ ม.ใหม่ จากนั้นสร้างจุดมาร์คเกอร์ด้วยคำสั่ง folium.Marker() โดยป้อนพิกัดและข้อความที่ต้องการแสดงผล การวาดเส้นตรงใช้คำสั่ง folium.PolyLine() โดยป้อนพิกัด x,y ของแต่ละจุด line_opacity คือ ความโปร่งของเส้น จากนั้นแสดงแผนที่ด้วยคำสั่ง mp

ตัวอย่างที่ 10.4 การเขียนเส้นด้วย polyline หลายจุดด้วย Folium

```
import folium
d = ["16.246096013122653, 103.25525358895801",
     "16.245426484563318, 103.25524286012269",
     "16.244715751751986, 103.25514630060484",
     "16.24404621849295, 103.25483516438064",
     "16.244159524281656, 103.25440601096794",
     "16.244880559589184, 103.25324729675368",
     "16.24583850242487, 103.25218514200752",
     "16.246250791217086, 103.2525176576928"].split("\n")
points = []
for i in d:
    a = i.split(",")
    points.append([float(a[0]), float(a[1])])
cp = [16.2458799759769, 103.25218506399402]
mp = folium.Map(location = [cp[0],cp[1]],zoom_start = 16)
folium.PolyLine(points, color="red", weight=10, opacity=0.3).add_to(mp)
mp
```

ผลลัพธ์



ภาพประกอบที่ 10.13 แสดงการวาดเส้นด้วยคำสั่ง polyline

ตัวอย่างที่ 10.4 นำพิกัดใส่ลงในตัวแปร `d` เป็นชนิดสตริงและทำการแยกด้วยเครื่องหมายขึ้นต้นบรรทัดใหม่ด้วยคำสั่ง `split('\n')` จากนั้น ประกาศตัวแปร `points` เป็นตัวแปรลิสต์มีค่าว่างเปล่า วนเข้าไปในตัวแปร `d` กำหนด `a` แยกพิกัดแต่ละบรรทัดด้วยเครื่องหมายคอมม่า (,) ละนำไปต่อย้ายตัวแปร `points` ด้วยคำสั่ง `append()` โดย `a[0]` คือละติจูด และ `a[1]` คือ ลองจิจูด ให้ทำการแปลงสตริงเป็นเลขทศนิยมด้วยคำสั่ง `float()` แล้วนำใส่ลงในตัวแปร `points`

ตัวแปร `cp` คือพิกัดเริ่มต้นที่ต้องการแสดงภาพแผนที่ นำพิกัดแต่ละจุดนำ มาใส่ตัวแปร `points` ตัวแปร `mp` จะแสดงแผนที่ตำแหน่งกึ่งกลางแผนที่ตำแหน่ง `cp[0]` และ `cp[1]` ด้วยอัตราขยายแผนที่ 16 และเขียนเส้นด้วยคำสั่ง `PolyLine()` โดยใช้ตัวแปร `points` เป็นจุดที่เขียนเส้นและวาดด้วยสีแดง หนา 10 และความโปร่งแสง 0.3

ตัวอย่างที่ 10.5 การวาดตราบมหาวิทยาลัยลงบนแผนที่และแสดงรายละเอียด

```
import folium
import pandas as pd
data = pd.DataFrame({
    'lat':[16.199662313316438,16.245327039142097],
    'lng':[103.28357298658125,103.25143414503279],
    'name':['มหาวิทยาลัยมหาสารคาม (ม.เก่า)', 'มหาวิทยาลัยมหาสารคาม (ม.ใหม่)']}, dtype=str)
m = folium.Map(location=[data['lat'][0], data['lng'][0]], zoom_start=13)
for i in range(0,len(data)):
    html=f"""
    <link href="https://fonts.googleapis.com/css?family=Athiti" rel="stylesheet">
```

```

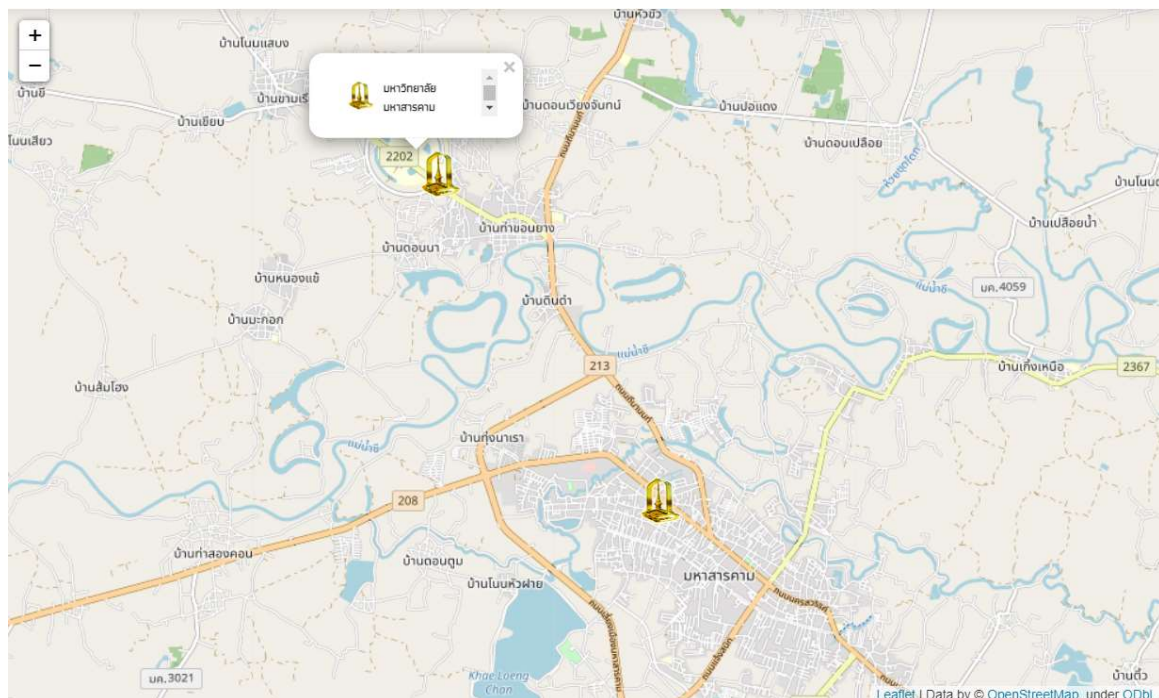

<h1 style="font-family:'Athiti';font-size:10px"> {data.iloc[i]['name']}</h1>
""

iframe = folium.IFrame(html=html, width=140, height=40)
popup = folium.Popup(iframe, max_width=2650)
folium.Marker(
    location=[data.iloc[i]['lat'], data.iloc[i]['lng']],
    popup=popup,
    icon=folium.DivIcon(html=f""<div><svg xmlns="http://www.w3.org/2000/svg">
<image href="https://dsdi.msu.ac.th/images/msu-i.png" height="50" width="50"/>
</svg></div>""")
).add_to(m)

```

M

ผลลัพธ์



ภาพประกอบที่ 10.14 แสดงการวาดตรามหาวิทยาลัยลงบนแผนที่และแสดงรายละเอียด

ตัวอย่างที่ 10.5 เป็นการวาดภาพ msu-i.png ลงบนแผนที่และเมื่อคลิกที่ตรามหาวิทยาลัยจะแสดงรายละเอียด ตัวแปร data จะเป็นชนิด DataFrame ประกอบด้วย lat, lng และ name ตัวแปร m เป็นแผนที่และแสดงข้อมูลที่พิกัด data['lat'][0] , data['lng'][0] โดยขยายแผนที่ zoom_start=13

การวนเข้าไปในตัวแปร data แต่ละรอบเพื่อเข้าไปกำหนดคุณลักษณะของมาร์คเกอร์ เช่น แสดงผลด้วยฟอนต์ Athiti กำหนดด้วยแท็ก `<link href="https://fonts.googleapis.com/css?family=Athiti" rel="stylesheet">` จากนั้นแสดงตรามหาวิทยาลัยมหาสารคาม ด้วยคำสั่ง `` กำหนดความกว้างและสูงของภาพ 30px และกำหนด float:left คือ ตัวอักษรที่พิมพ์ต่อจากภาพจะลอยชิดด้านบนซ้ายของภาพ จากนั้นแสดงข้อความด้วยคำสั่ง `<h1 style="font-family:'Athiti';font-size:10px">{data.iloc[i]['name']}` เมื่อ H1 คือแท็ก Heading 1 กำหนดการแสดงผลโดยใช้ฟอนต์ Athiti ด้วย font-family:Athiti และขนาดฟอนต์ด้วย font-size:10px และแสดงข้อมูลในตัวแปร data ด้วยคำสั่ง `{data.iloc[i]['name']}` คือข้อมูลภายในตัวแปร data ที่คอลัมน์ name ตัวแปร iframe จะสร้างเฟรมเอกสาร HTML ที่มีความกว้าง 140 และสูง 40 ซึ่งจะใช้กำหนดขนาด Popup จากนั้นทำการสร้าง Marker โดยกำหนดตัวแปร `location=[data.iloc[i]['lat'], data.iloc[i]['lon']]` ซึ่งจะนำพิกัดในตัวแปร data มาสร้าง Marker และกำหนด icon คือภาพที่ต้องการแสดงด้วยคำสั่ง `DivIcon()` โดยใช้แท็ก `<svg>` เพื่อสร้างภาพเวกเตอร์ และภายใน svg ใส่ภาพด้วยแท็ก `<image` โดยมี href="ลิงค์ของภาพ" โดยแอตทริบิวต์ height และ width = 50 พิกเซล จากนั้นนำ marker ใส่ลงในตัวแปร m ด้วยคำสั่ง `add_to()`

10.11 การใช้งาน GeoPandas

GeoPandas เป็นไลบรารีด้านการประมวลผลข้อมูลเชิงพื้นที่ (Geospatial data) ทำหน้าที่จัดการข้อมูลประเภท geometry data type การติดตั้งทำได้โดยเรียกใช้คำสั่ง `!pip install geopandas` หรือติดตั้งจากเว็บ <https://pypi.org/project/geopandas>

ตัวอย่างที่ 10.6 การแสดงผลข้อมูลภายใน province.shp

```
%pylab inline
import pandas as pd
import geopandas as gpd
province = gpd.read_file("Province.shp")
province.head(3)
```

ผลลัพธ์

	AREA	PROV_CODE	PROV_NAME_T	PROV_NAME_E	geometry
0	1.566623e+09	10	กรุงเทพมหานคร	Bangkok	POLYGON ((702982.043 1542027.044, 702982.980 1...
1	9.671162e+08	11	จังหวัดสมุทรปราการ	Changwat Samut Prakarn	POLYGON ((656510.646 1497394.325, 656509.269 1...
2	6.364005e+08	12	จังหวัดนนทบุรี	Changwat Nonthaburi	POLYGON ((636577.088 1539351.489, 636587.152 1...

ภาพประกอบที่ 10.15 แสดงรายละเอียดของไฟล์ province.shp

รายละเอียดของไฟล์ Province.shp ประกอบด้วย 5 คอลัมน์ โดยมีคอลัมน์ PROV_NAME_T เป็นชื่อจังหวัดที่มีการเข้ารหัสเป็น Windows 874 ทำให้แสดงผลไม่ถูกต้อง จะอธิบายวิธีการแก้ไขในหัวข้อถัดไป

ที่คอลัมน์ geometry จะเป็นข้อมูลเป็นพิกัดต่าง ๆ ในรูปของ POLYGON() ซึ่งภายในประกอบด้วยพิกัดจุดที่เป็นขอบเขตของจังหวัด

10.12 การสร้างไฟล์ .shp ด้วย GeoPandas

ในหัวข้อนี้จะสร้างไฟล์ shape โดยกำหนดข้อมูล name, lat, lon และ post

ตัวอย่างที่ 10.7 การสร้างไฟล์ shape ด้วยข้อมูล Points

```
%pylab inline
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point
from geopandas import GeoDataFrame

data = {
    'name': [ 'ร้อยเอ็ด','ขอนแก่น','สาคาม'],
    'lat':[ 15.919586563602, 16.40871542099, 15.999675121040],
    'lng':[ 103.8132632550, 102.5785438879, 103.165512463],
    'code':['45000','40000','44000' ]
}

df = pd.DataFrame(data)

from geopandas import GeoDataFrame
g = [Point(i) for i in zip(df['lng'], df['lat'])]
gdf = GeoDataFrame(df, geometry= g )
gdf.to_file("roi-kan-san.shp",encoding='utf-8')
```

ตัวอย่างที่ 10.7 ขอใช้ไลบรารี shapely และ geopandas โดยตัวแปร data จะประกอบด้วยชื่อจังหวัด พิกัด และรหัสไปรษณีย์ของ 3 จังหวัด จากนั้นแปลงเป็น df ด้วยคำสั่ง DataFrame() ให้ตัวแปร g เป็นพิกัดของจุด โดยรวมตัวแปรคู่ด้วยคำสั่ง zip(พิกัด) จากนั้นสร้าง GeoDataFrame() ด้วยตัวแปร df และ g และนำไปเขียนลงไฟล์ตั้งชื่อ roi-kan-san.shp เข้ารหัสไฟล์แบบ UTF-8 ผลลัพธ์ที่ได้เป็น shape ที่สามารถนำไปใช้งานในโปรแกรมแผนที่ทั่วไปได้

10.13 การเข้ารหัส Province.shp เป็น UTF8

ไฟล์ Province.shp เข้ารหัสตัวอักษรด้วยมาตรฐาน windows 874 หรือ TIS-620 เมื่อนำไปใช้งานอาจเกิดปัญหาการแสดงผลควรหลีกเลี่ยงด้วยการเข้ารหัสอักษรเป็น Utf-8 เขียนฟังก์ชัน tis620_utf8() ดังนี้

ตัวอย่างที่ 10.8 การสร้างฟังก์ชัน tis620_utf8()

```
def tis620_utf8(text):
    b = text.encode('cp1252')
    th = b.decode('tis-620')
    if th[0:2] == "จ.":
        th = th[2:]
    return th

m = gpd.read_file("Province/Province.shp")
c = list(map(lambda a:tis620_utf8(a), m['PROV_NAM_T']))
m['PROV_NAM_T'] = c
m.to_file("Province_utf8.shp",encoding='utf-8')
```

ตัวอย่างที่ 10.8 ฟังก์ชัน tis620_utf8() ใช้ถอดรหัสภาษาไทยด้วยมาตรฐาน TIS-620 หรือ Windows 874 ให้เป็นข้อความภาษาไทย ตัวแปร m ใช้อ่านไฟล์ Province.shp ตัวแปร c จะแปลงข้อมูลภายในคอลัมน์ "PROV_NAM_T" ให้เป็นภาษาไทย ภายในฟังก์ชัน tis620_utf8() จะตรวจสอบว่าอักขระ 2 ตัวแรกเป็นคำว่า "จ." หรือไม่ถ้าใช่จะลบทิ้งไปให้เหลือเฉพาะชื่อจังหวัดเท่านั้น ผลลัพธ์ที่ได้นำมาแทนที่ในตัวแปร m จากนั้นบันทึกไฟล์ ด้วยคำสั่ง to_file() โดยตั้งชื่อไฟล์ "Province_utf8.shp" และเข้ารหัสแบบ utf-8 ผลลัพธ์แสดงได้ดังนี้

	AREA	PROV_CODE	PROV_NAM_T	PROV_NAM_E	geometry
0	1.566623e+09	10	กรุงเทพมหานคร	Bangkok	POLYGON ((702982.043 1542027.044, 702982.980 1...
1	9.671162e+08	11	สมุทรปราการ	Changwat Samut Prakarn	POLYGON ((656510.646 1497394.325, 656509.269 1...
2	6.364005e+08	12	นนทบุรี	Changwat Nonthaburi	POLYGON ((636577.088 1539351.489, 636587.152 1...
3	1.520801e+09	13	ปทุมธานี	Changwat Pathum Thani	POLYGON ((644830.648 1559494.020, 644831.525 1...
4	2.547348e+09	14	พระนครศรีอยุธยา	Changwat Phra Nakhon Si Ayudhya	POLYGON ((633326.005 1570597.773, 633327.005 1...
...	...	...	...	...	...
71	4.730100e+09	92	ตรัง	Changwat Trang	MULTIPOLYGON (((531219.697 842203.686, 531219....
72	3.859920e+09	93	พัทลุง	Changwat Phatthalung	POLYGON ((583534.064 863245.678, 583534.253 86...
73	1.976771e+09	94	ปัตตานี	Changwat Pattani	POLYGON ((724297.064 749739.546, 724299.124 74...
74	4.475016e+09	95	ยะลา	Changwat Yala	POLYGON ((712530.050 710610.940, 712583.237 71...
75	4.491204e+09	96	นราธิวาส	Changwat Narathiwat	POLYGON ((765049.300 697375.877, 765052.302 69...

76 rows × 5 columns

ภาพประกอบที่ 10.16 การแสดงผลรายชื่อจังหวัดด้วยการเข้ารหัส UTF-8

10.14 การสกัดข้อมูลราคากิจงานเบกษาขงบประมาณประจำปี 2564 ลงในแผนที่จังหวัด

ประเทศไทยมีการใช้เงินภาษีเพื่อจัดสรรให้กับหน่วยงานต่าง ๆ ของภาครัฐ ข้อมูลเหล่านั้นอยู่ในรูปแบบไฟล์ .pdf ในหัวข้อนี้จะสกัดข้อมูลที่ได้รับการสนับสนุนขององค์การบริหารส่วนจังหวัด ใส่ลงในแผนที่ประเทศไทย มีขั้นตอนดังนี้

1. ขั้นตอนการเตรียมข้อมูล ดาวน์โหลดราชกิจจานุเบกษาว่าด้วยเรื่องงบประมาณรายจ่ายประจำปีของประเทศไทย ได้ที่ลิงค์
http://dsdi.msu.ac.th/programming/th_budget/2564.pdf
2. แปลง pdf เป็น text ผ่านเว็บ เช่น <https://avepdf.com/pdf-to-text> นำไฟล์งบประมาณรายจ่ายประจำปีเข้าเว็บดังกล่าว จากนั้นแปลงเป็นเท็กซ์ไฟล์และนำเท็กซ์ไฟล์ที่ได้มาประมวลผลเพื่อสกัดงบประมาณของแต่ละจังหวัดออกมา
3. เขียนโปรแกรมสกัดข้อมูลการตั้งงบประมาณประจำปี 2564 ดังนี้

ตัวอย่างที่ 10.9 การเขียนโปรแกรมสกัดข้อมูลหมวดงบประมาณรายจ่ายประจำปี 2564

```
t= open("2564.txt","r",encoding='utf-8').read()
nt = "๑๒๓๔๕๖๗๘๙๐"
ne = "1234567890"
for i in range(len(nt)):
    t = t.replace(nt[i],ne[i])

for i in range(100):
    t = t.replace(" ", " ")

t = t.replace(" ๑","๑"); t = t.replace(" คือ ", "")
tmp = t.split("\n")
z = []

for i in tmp:
    if ("ราชกิจจานุเบกษา" not in i) and ("หน้า" not in i) and (len(i.strip()) > 4):
        z.append(i.strip())
tmp = "".join(z).replace("\n","")
tmp = tmp.split(" บาท")
for i in tmp:
    if "." in i:
        print("\n%s บาท"%(i))
    else:
        print("%s บาท"%(i))
```

ผลลัพธ์

- (ก) องค์การบริหารส่วนจังหวัดกาฬสินธุ์ 724,488,600 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 724,488,600 บาท
 (ข) องค์การบริหารส่วนจังหวัดขอนแก่น 1,044,598,900 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 1,044,598,900 บาท
 (ค) องค์การบริหารส่วนจังหวัดมหาสารคาม 782,110,300 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 782,110,300 บาท
 (ง) องค์การบริหารส่วนจังหวัดร้อยเอ็ด 672,937,000 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 672,937,000 บาท
 (13) องค์การบริหารส่วนจังหวัดในพื้นที่ภาคตะวันออกเฉียงเหนือตอนล่าง 1 รวม 4,302,157,500 บาท
 (ก) องค์การบริหารส่วนจังหวัดชัยภูมิ 995,566,300 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 995,566,300 บาท
 (ข) องค์การบริหารส่วนจังหวัดนครราชสีมา 2,303,398,700 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 2,303,398,700 บาท
 (ค) องค์การบริหารส่วนจังหวัดบุรีรัมย์ 548,335,700 บาท
 1) แผนงานยุทธศาสตร์ส่งเสริมการกระจายอำนาจให้แก่องค์กรปกครองส่วนท้องถิ่น 548,335,700 บาท
 (ง) องค์การบริหารส่วนจังหวัดสุรินทร์ 454,856,800 บาท

ภาพประกอบที่ 10.17 ผลลัพธ์จากการแปลงเอกสาร PDF เป็นข้อความและทำความสะอาดข้อมูล

ตัวอย่างที่ 10.9 ตัวแปร t เก็บข้อมูลของไฟล์ 2564.txt ด้วยโหมดการอ่านเข้ารหัสแบบ utf-8 จากนั้น ตัวแปร nt เป็นเลขไทย ตัวแปร ne เป็นเลขอารบิก วนเข้าไปในตัวแปร nt และแทนที่เลขไทยด้วยตัวเลขอารบิก ในการทำความสะอาดข้อมูลจะต้องอาศัยการสังเกตข้อมูลและพยายามทำให้ข้อมูลอยู่ในรูปแบบจะสกัดข้อมูลออกมาได้ ด้วยประเทศไทยไม่มีการเปิดเผยข้อมูลภาครัฐให้อยู่ในรูปแบบที่เครื่องจักรสามารถอ่านได้ (Machine Readable) ทำให้การนำข้อมูลภาครัฐไปใช้งานทำได้อย่างยากลำบาก

4. สกัดงบประมาณองค์การบริหารส่วนจังหวัด

ตัวอย่างที่ 10.10 การสกัดงบประมาณองค์การบริหารส่วนจังหวัด

```
d = {}
tmp = z.split(" องค์การบริหารส่วนจังหวัด")
for i in tmp[1:]:
    a = i.split("\n")[0]
    if " รวม " not in a:
        b = a.replace(" บาท","").replace(",","")
        p,v = b.split(" ")
        d.update({'%s'%p: '%s'%v})

print(d)
```

ผลลัพธ์

```
{'ชั๊ยนาท': '202122100',
 'พระนครศรีอยุธยา': '287952300',
 'ลพบุรี': '232428600',
 'สระบุรี': '211362000',
 'สิงห์บุรี': '118798800',
 ...}
```

```
'พิจิตร': '274943100',  
'อุทัยธานี': '176171300'}
```

ตัวอย่างที่ 10.10 แยกคำว่า ") องค์การบริหารส่วนจังหวัด" ออกจากกันจากนั้นทำการลบเครื่องหมายคอมม่า (,) ทิ้งไปและแยกให้ได้ส่วนของชื่อจังหวัดและงบประมาณจากนั้นอัปเดตตัวแปร d

5. เพิ่มข้อมูลงบประมาณจัดสรรขององค์การบริหารส่วนจังหวัดลงในแผนที่ ดังนี้

ตัวอย่างที่ 10.11 การสร้างตัวแปรงบประมาณ budget64

```
m = gpd.read_file("Province_utf8.shp")  
budget2564 = list(map(lambda a: int(d[a]), m['PROV_NAM_T']))  
m["งบ_อบจ_2564"] = budget2564  
m
```

6. อัปเดต Shape file ใหม่ดังนี้

ตัวอย่างที่ 10.12 การอัปเดต shapefile

```
m.to_file("Province_utf8.shp", encoding='utf-8')
```

ตัวอย่างที่ 10.12 การเพิ่มคอลัมน์ "งบ_อบจ_2564" ลงในตัวแปร m ซึ่งอ่านข้อมูลจากไฟล์ Province.shp โดยตัวแปร budget2564 = list(map(lambda a: d[a], m['PROV_NAM_T'])) เมื่อ m['PROV_NAM_T'] เป็นตัวแปรรายการส่งเข้าไปใน lambda แทนด้วยตัว a ซึ่งเป็นชื่อจังหวัด จากนั้นคืนค่าออกมาเป็นงบประมาณจัดสรรประจำปีภายในตัวแปร d['จังหวัด'] และคืนค่ากลับมา โดยคำสั่ง int() เป็นการแปลงจากสตริงเป็นเลขจำนวนเต็ม

10.5 การปรับโครงสร้างแผนที่จังหวัดและงบประมาณรายจ่ายประจำปี 2564

เนื่องจากแผนที่เก่ามี 76 จังหวัด ในปัจจุบันเพิ่มขึ้นอีก 1 จังหวัดคือ จังหวัดบึงกาฬ แยกออกมาจากจังหวัดหนองคาย มีรหัสจังหวัด คือ 38 ผู้เขียนปรับโครงสร้าง shapefile ใหม่ให้มีข้อมูลอัปเดตล่าสุด และเปลี่ยนชื่อคอลัมน์เพื่อให้สั้นและกระชับ ตั้งชื่อไฟล์ Province2564 มีรายละเอียดคอลัมน์ ดังนี้

	post	th	en	lat	lng	budget64	geometry
0	10	กรุงเทพมหานคร	Bangkok	13.771252	100.622135	22089883700	POLYGON (((673587.43200 1543227.55870, 673772.9...
1	11	สมุทรปราการ	Samut Prakarn	13.595962	100.708793	170387000	POLYGON (((687489.38870 1517117.08020, 687672.5...
2	12	นนทบุรี	Nonthaburi	13.924425	100.392413	673095800	POLYGON (((642612.16140 1562465.24460, 642761.4...
3	13	ปทุมธานี	Pathum Thani	14.064956	100.681723	275592100	POLYGON (((704054.90790 1576344.97530, 704169.6...
4	14	พระนครศรีอยุธยา	Phra Nakhon Si Ayudhya	14.344919	100.527038	287952300	POLYGON (((662539.32780 1622688.88150, 662550.0...
...	...	...	...	...	...	...	...
72	92	ตรัง	Trang	7.543923	99.594848	236165600	MULTIPOLYGON (((552493.61910 770736.07500, 552...
73	93	พัทลุง	Phatthalung	7.516274	100.069903	338926200	POLYGON (((605913.18960 872177.79630, 606158.94...
74	94	ปัตตานี	Pattani	6.730596	101.349710	242877400	POLYGON (((753341.66710 768818.74030, 753695.10...
75	95	ยะลา	Yala	6.190659	101.228051	187694100	POLYGON (((753445.83770 739010.61360, 753573.58...
76	96	นราธิวาส	Narathiwat	6.177805	101.718655	195853400	POLYGON (((791093.97870 733078.18650, 791112.79...

77 rows × 7 columns

ภาพประกอบที่ 10.18 แสดงการปรับโครงสร้าง province2564.shp

10.16 การแสดงแผนที่ตามระดับสีจากงบประมาณรายจ่ายประจำปี 2564

การพล็อตด้วยระดับสีตามระดับงบประมาณรายจ่ายประจำปี 2564 แสดงดังนี้

ตัวอย่างที่ 10.13 การแสดงแผนที่ตามระดับสีจากงบประมาณรายจ่ายประจำปี 2564

```
import pandas as pd
import geopandas as gpd
from geopandas import GeoDataFrame
m = gpd.read_file("province2564.shp")
m.crs="EPSG:32647"
gdf = GeoDataFrame(m)
gdf = gdf.to_crs("epsg:4326")
gdf["budget64"] = list(map(lambda k:int(k), gdf["budget64"]))
p1 = gdf[(gdf.budget64 < 30000000000)]
p1.plot(cmap='rainbow',figsize=(20,20),edgecolor='k', lw=0.7)

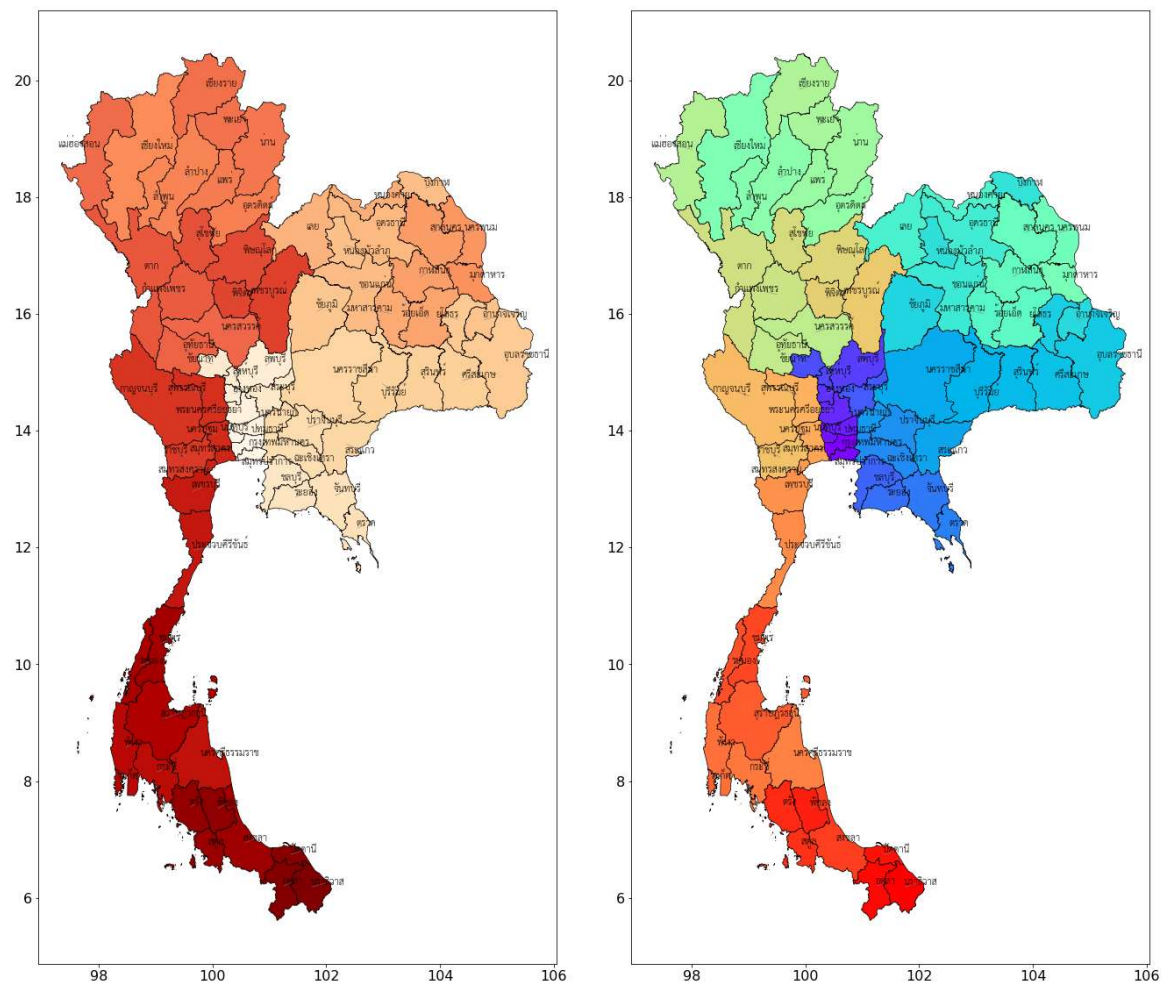
import matplotlib.font_manager as fm
import matplotlib.pyplot as plt
THSarabun = fm.FontProperties(fname='fonts/THSarabun.ttf')
```

```

texts = []
for x, y, label in zip(gdf.lng, gdf.lat, gdf.th):
    texts.append(plt.text(x, y, label, fontproperties=THSarabun, fontsize = 16))
import adjustText as aT
aT.adjust_text(texts, force_points=0.3, force_text=0.8, expand_points=(1,1), \
expand_text=(1,1), arrowprops=dict(arrowstyle="", color='grey', lw=0.5))

```

ผลลัพธ์



ภาพประกอบที่ 10.19 แสดงแผนที่ระบายสีตามระดับกระจายประจำปี 2564

ตัวอย่างที่ 10.13 ภายในไฟล์ Province2564.shp ประกอบด้วยคอลัมน์ budget64 เป็นงบประมาณรายจ่ายประจำปี 2564 หลังจากโหลดไฟล์ shape แล้วให้แปลงเป็นระบบพิกัด EPSG:4326 ตัวแปร p1 จะเป็นการกรองเอาจังหวัดที่ใช้งบประมาณที่น้อยกว่า 3 หมื่นล้านบาทออกมา เนื่องจากกรุงเทพมหานครได้รับงบประมาณที่สูงกว่าจังหวัดอื่นทำให้ช่วงสีมีความห่างจากกลุ่มสีที่ใช้ในการระบายลงบนแผนที่ โดยทั่วไปจังหวัดอื่นใช้งบประมาณประมาณ 100 ถึง 1 พันล้านบาท แต่ กทม ใช้สูงถึง 2 หมื่นสองพันล้านบาท

10.17 การเรียงลำดับข้อมูล GeoDataFrame ด้วย sort_values()

เนื่องจาก GeoDataFrame ทำงานอยู่บน pandas ดังนั้น สามารถเรียงลำดับด้วยคำสั่ง sort_values() โดยมีพารามิเตอร์ by คือคอลัมน์ที่ต้องการเรียง และ ascending คือ เรียงจากน้อยไปมาก หรือมากไปน้อย

ตัวอย่างที่ 10.14 การเรียงลำดับข้อมูล GeoDataFrame ด้วย sort_values()

```
import pandas as pd
import geopandas as gpd
from geopandas import GeoDataFrame
m = gpd.read_file("province2564.shp")
m.sort_values(by='budget64', ascending=False)
```

ผลลัพธ์

	post	th	en	lat	lng	budget64	geometry
0	10	กรุงเทพมหานคร	Bangkok	13.771252	100.622135	22089883700	POLYGON ((100.60692 13.95401, 100.60864 13.953...
18	30	นครราชสีมา	Nakhon Ratchasima	14.956593	102.109733	2303398700	POLYGON ((102.31857 15.80756, 102.31843 15.806...
21	33	ศรีสะเกษ	Sisaket	14.856449	104.368465	1342135900	POLYGON ((104.10559 15.56880, 104.10647 15.568...
28	40	ขอนแก่น	Khon Kaen	16.408721	102.578555	1044598900	POLYGON ((102.70693 17.08704, 102.70754 17.086...
22	34	อุบลราชธานี	Ubon Ratchathani	15.183975	105.110870	1016626700	POLYGON ((105.06703 16.09446, 105.06830 16.093...
...	...	...	...	...	...	...	...
16	26	นครนายก	Nakhon Nayok	14.217119	101.171100	143050400	POLYGON ((101.28252 14.49773, 101.28270 14.496...
13	23	ตราด	Trat	12.317309	102.520763	137896200	MULTIPOLYGON (((102.51403 11.72591, 102.51422 ...
60	75	สมุทรสงคราม	Samut Songkhram	13.396770	99.953244	120500600	POLYGON ((100.02546 13.50499, 100.02563 13.504...
7	17	สิงห์บุรี	Singburi	14.911987	100.347796	118798800	POLYGON ((100.36497 15.09354, 100.39081 15.075...
68	85	ระนอง	Ranong	9.970233	98.699031	103434700	MULTIPOLYGON (((98.39265 9.39969, 98.39259 9.3...

77 rows × 7 columns

ภาพประกอบที่ 10.20 ผลลัพธ์การเรียงลำดับจากคอลัมน์ budget64

ตัวอย่าง 10.14 เป็นการเรียงข้อมูลด้วยคำสั่ง sort_values() โดยใช้ข้อมูลในคอลัมน์ budget64 และเป็นการเรียงจากมากไปน้อยด้วยพารามิเตอร์ sort_values(by='budget64', ascending=False) พบว่ากรุงเทพมหานครได้รับงบประมาณสูงสุดที่สุดเป็นเงิน 2.4 หมื่นล้านบาท รองลงมาคือนครราชสีมา 2.3 พันล้านบาท และจังหวัดที่ได้รับการจัดสรรงบประมาณน้อยที่สุดคือ จังหวัดระนองเป็นเงิน 98 ล้านบาท
หมายเหตุ: ในแผนที่ยังไม่รวมเขตปกครองพิเศษ เมืองพัทยาที่ได้รับเงินงบประมาณ 1.9 พันล้านบาท

10.17 การติดตั้งและใช้งาน GeoServer ผ่าน Docker

Docker ทำหน้าที่คล้ายคอมพิวเตอร์เสมือน (Virtual Machine) คือ สามารถติดตั้งซอฟต์แวร์ต่าง ๆ ในคอมพิวเตอร์เสมือนโดยไม่ยุ่งเกี่ยวกับระบบปฏิบัติการหลัก (Host OS) โดยที่ docker จะทำงานได้เร็วกว่า vm เพราะมีการใช้ทรัพยากรร่วมกับ Host OS เช่น หน่วยความจำ หน่วยประมวลผลกลาง เป็นต้น ศึกษารายละเอียดเกี่ยวกับ docker เพิ่มเติมได้ที่ <https://www.docker.com>

1. ในหัวข้อนี้เป็นการเปิดบริการ GeoServer ผ่าน docker เรียกใช้งานด้วยคำสั่งต่อไปนี้

ตัวอย่างที่ 10.15 การเปิดบริการ GeoServer ด้วย docker

```
docker pull quay.io/geodocker/base
docker run --name geoserver -e -d -p 9090:9090 quay.io/geodocker/geoserver
```

เมื่อ docker คือ คำสั่งในการรัน docker

pull คือ การโหลดคอนเทนเนอร์ quay.io/geodocker/base

run คือ การรัน container ในตัวอย่างนี้ชื่อ quay.io/geodocker/geoserver

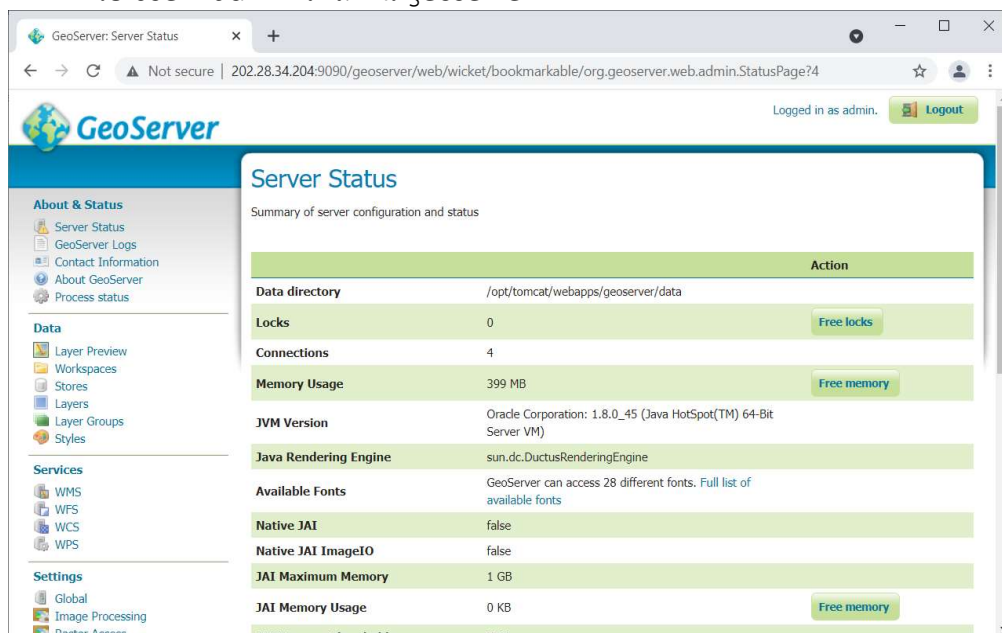
-- name คือ ชื่อ service ในตัวอย่างนี้ตั้งชื่อ geoserver

-d คือ ทำงานในโหมดเบื้องหลัง (daemon)

-p คือ การแมปพอร์ต เช่น 9090:9090 คือตัวเลขด้านซ้ายคือพอร์ตของโฮสและตัวเลขด้านขวาคือพอร์ตของคอนเทนเนอร์

-v คือ การแมป host path กับ container path

2. เปิดบราวเซอร์ด้วย url : <http://202.28.34.204:9090/geoserver> และ Login เข้าสู่ระบบด้วย user : admin รหัสผ่าน geoserver



ภาพประกอบที่ 10.21 แสดงสถานะของ GeoServer

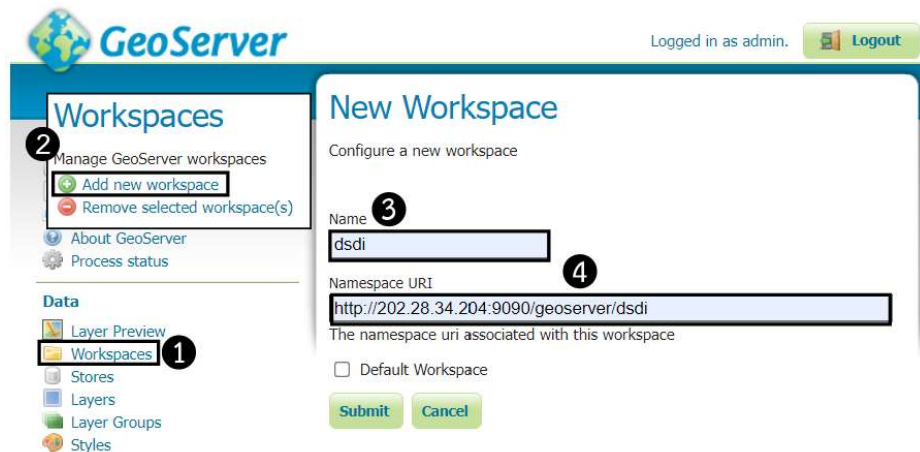
หมายเหตุ: การแมปไดเร็กทอรี

```
docker -v /host/data:/opt/tomcat/webapps/geoserver/data
```

ใน geoserver จะบันทึกข้อมูลไว้ในโฟลเดอร์ /opt/tomcat/webapps/geoserver/data ดังนั้น
ต้องทำการแก้ไขไฟล์ไดเรกทอรีกับไดเรกทอรีภายใน geoserver ด้วยพารามิเตอร์ -v

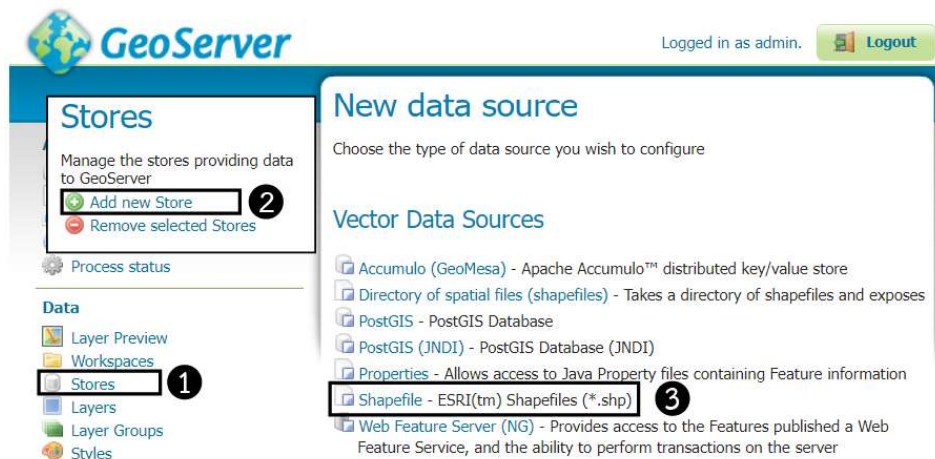
10.18 การเผยแพร่ข้อมูลบน GeoServer

1. เลือก Workspaces -> Add new workspace จากนั้นกำหนดชื่อ Name = ชื่อโครงการ
และ URI = http://เลขไอพีของแม่ข่าย:พอร์ต/geoserver/ชื่อโครงการ ดังนี้



ภาพประกอบที่ 10.22 แสดงการสร้าง workspace

2. เลือก Stores -> Add new Store -> Shapefile



ภาพประกอบที่ 10.23 แสดงการเพิ่ม Data Source ชนิด Shapefile

3. เลือกไฟล์ Workspace และตั้งชื่อ Data Source Name เลือกไฟล์และกำหนดการเข้ารหัส
ตัวอักษร ดังนี้

New Vector Data Source

Shapefile
ESRI(tm) Shapefiles (*.shp)

Basic Store Info

Workspace *
dsdi 1

Data Source Name *
province 2

Description

☒ Enabled

Connection Parameters

Shapefile location *
file:data/province/Province/Province.shp Browse... 3

DBF charset
x-windows-874 4

☒ Create spatial index if missing/outdated
☒ Use memory mapped buffers (Disable on Windows)
☒ Cache and reuse memory maps (Requires 'Use Memory mapped buffers' to be enabled)

ภาพประกอบที่ 10.24 แสดงการกำหนดการเข้ารหัสตัวอักษร windows-874

เนื่องจากในตัวอย่างนี้ใช้ shapfile ตัวเก่ามีชื่อไฟล์ Province.shp ดังนั้น ยังคงเข้ารหัสแบบ windows-874 หากใช้ shapefile ตัวใหม่ Province2564.shp จะต้องเลือกการเข้ารหัสแบบ UTF-8

4. เลือก Layers -> Add a new layer -> เลือก workspace และ store ที่สร้างขึ้น เช่น dsdi:province กดที่ Publish ดังนี้

New Layer

Add a new layer

Add layer from dsdi:province 1

You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#)
 Here is a list of resources contained in the store 'province'. Click on the layer you wish to configure

<< < 1 > >> Results 0 to 0 (out of 0 items) Search

Published	Layer name	Action
	Province	Publish 2

ภาพประกอบที่ 10.25 การเผยแพร่ shapefile ของ dsdi workspace

5. เลือก Find -> พิมพ์ 32648 -> เลือกระบบพิกัด WGS 84/UTM zone 48N -> กดคำนวณขอบเขตข้อมูลที่ปุ่ม Compute from data และกด Compute from native bounds ดังภาพต่อไปนี้

Coordinate Reference System

Native SRS
UNKNOWN WGS_1984_UT

Declared SRS
EPSG:32648 **1** Find... EPS

SRS handling
Force declared

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
325,217.21783880	620,830.06892958	1,213,677.586773	2,263,222.697306

Compute from data **4**

Compute from SRS bounds

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
103.32438561864	5.5813139570170	111.82475342644	20.467369848690

Compute from native bounds **5**

Select a coordinate system. Use the search box to narrow the list

2 3264

Code	Description
32645	WGS 84 / UTM zone 45N
32646	WGS 84 / UTM zone 46N
32647	WGS 84 / UTM zone 47N
32648	WGS 84 / UTM zone 48N
32649	WGS 84 / UTM zone 49N
61326405	FD58 (deg)
62326405	Fahud (deg)

3

ภาพประกอบที่ 10.26 แสดงการกำหนดระบบพิกัด CRS แบบ EPSG:32648

6. แสดงแผนที่จังหวัดด้วย OpenLayer

GeoServer

Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

Results 1 to 23 (out of 23 items)

Type	Title	Name	Common Formats	All Formats
Point	Spearfish bug locations	sf:bugsites	OpenLayers KML GML	Select
Polygon	Spearfish restricted areas	sf:restricted	OpenLayers KML GML	Select
Point	Spearfish archeological sites	sf:archsites	OpenLayers KML GML	Select
Line	Spearfish streams	sf:streams	OpenLayers KML GML	Select
Line	Spearfish roads	sf:roads	OpenLayers KML GML	Select
Polygon	Province	dsdi:Province	OpenLayers KML GML	Select
Polygon	TIGER New York	tiger-ny	OpenLayers KML	Select
Polygon	Spearfish	spearfish	OpenLayers KML	Select
Polygon	Tasmania	tasmania	OpenLayers KML	Select

1

2

3

Scale = 1 : 9M
Click on the map to get feature info

ภาพประกอบที่ 10.27 การแสดงผลแผนที่บน OpenLayer และทำงานผ่านเว็บ

แบบฝึกหัดท้ายบท

1. จงอธิบายการบอกพิกัดตำแหน่งด้วย UTM
2. จงอธิบายการเพิ่มและลบข้อมูลเวกเตอร์ลงบนแผนที่
3. จงอธิบายการตรึงพิกัดของภาพด้วย Georeferencing
4. จงอธิบายการนำเสนอข้อมูลบนแผนที่ด้วย Folium
5. จงอธิบายการจัดการข้อมูลแผนที่ด้วย GeoPandas
6. จงอธิบายการสกัดข้อมูลงบประมาณประจำปี 2564 ใส่ลงใน Shape ของจังหวัด
7. จงเขียนโปรแกรมสกัดงบประมาณประจำปี พ.ศ. 2564 ของแต่ละจังหวัดลงใน shape file ประเทศไทย

บทที่ 11

การเขียนโปรแกรมภาษาไพธอนในงานประมวลผลภาพ

แนวคิด

คอมพิวเตอร์กราฟิก (Computer Graphic) คือ การสร้างภาพจากการไม่มีสิ่งของอยู่จริงเกิดจากการคำนวณด้วยคุณสมบัติของผิวและการตกกระทบแสงด้วยการคำนวณ ได้แก่ โปรแกรม Maya, Sketchup, 3dmax ฯลฯ อิมเมจโปรเซสซิง (Image Processing) ทำงานตรงข้ามกับคอมพิวเตอร์กราฟิก คือ นำภาพที่มีอยู่แล้วมาผ่านการประมวลผลบางอย่างเพื่อให้เกิดเป็นภาพใหม่ เช่น ภาพที่มีความชัดเพิ่มขึ้น ภาพเบลอ หรือการคำนวณหาเส้นขอบภายในภาพ ฯลฯ ได้แก่ โปรแกรม Photoshop เป็นต้น

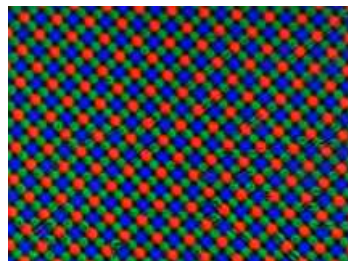
ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับ พิกเซลบนจอภาพคอมพิวเตอร์ (Pixel) โมเดลภาพแบบ RGB (Red Green Blue) การเขียนจุด เส้นตรงและวงกลม (Output primitive) การบวกและลบค่าสี การแยกและรวมสี RGB การกลับสีภาพ (Invert Color) การเขียนโปรแกรมเพื่อปรับค่าแกมมาของภาพ การระบายพื้นที่ภายในภาพ การกลับภาพซ้ายขวา/บนล่าง คอนโวลูชันและเคอร์เนล (Convolution และ Kernel) การตรวจจับใบหน้าด้วย Haar Cascade การตรวจจับใบหน้าด้วย MediaPipe การตรวจจับใบหน้าขนาดเล็กด้วย Tinyface การตรวจจับวัตถุด้วย Yolo3 การตรวจจับวัตถุด้วย MediaPipe การตรวจจับอิริยาบถและการตรวจจับแลนด์มาร์คของใบหน้า

วัตถุประสงค์

1. อธิบายการวาดจุดเส้นตรงและวงกลมด้วย OpenCV ได้
2. อธิบายการดำเนินการเกี่ยวกับบิต การรวมสี แยกสี ปรับแกมมาและกลับภาพด้วย OpenCV ได้
3. อธิบายการทำงานของคอนโวลูชันและเคอร์เนลได้
4. อธิบายการตรวจจับใบหน้าด้วย Haar Cascade และ Tinyface ได้
5. อธิบายการตรวจจับวัตถุด้วย Yolo3 ได้

11.1 พิกเซลบนจอภาพคอมพิวเตอร์ (Pixel)

บนหน้าจอคอมพิวเตอร์ประกอบด้วยสีหลัก 3 สี คือ RGB (Red, Geen, Blue) เมื่อนำกล้องขยายมาทำการขยายหน้าจอคอมพิวเตอร์ 1,000 เท่าและส่งลงไปบริเวณสีขาบบนหน้าจอคอมพิวเตอร์แสดงผลดังนี้

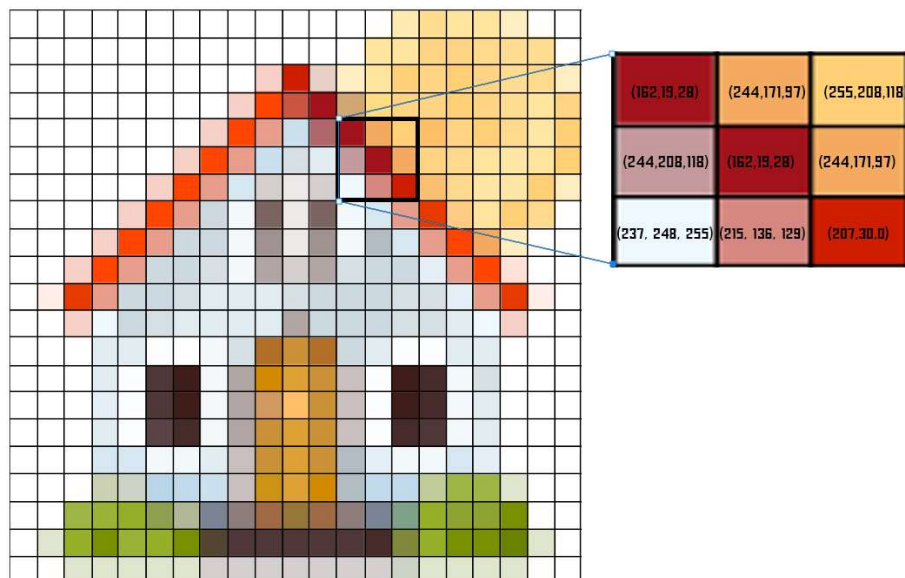


ภาพประกอบที่ 11.1 แสดงการเรียงตัวของพิกเซลบนจอภาพคอมพิวเตอร์

11.2 โมเดลภาพแบบ RGB (Red Green Blue)

จุดภาพ (Pixel ย่อจาก Picture Element) แต่ละจุดจะประกอบด้วย 3 สี คือ RGB (Red , Green , Blue) (แดง , เขียว , น้ำเงิน) โดยแต่ละสีจะมีค่าระหว่าง 0-255 ค่าของสี (255,255,255) จะหมายถึง Red , Green , Blue มีค่าสูงสุดคือ 255 เม็ดสีที่แสดงออกมาบนจอภาพจะเป็นสีขาว เมื่อค่าสีมีค่า (0,0,0) จะหมายถึง Red , Green , Blue มีค่าต่ำสุดคือ 0 เม็ดสีที่แสดงออกมาบนจอภาพจะเป็นสีดำ ส่วนเม็ดสี (255,0,0) จะหมายถึง Red มีค่าสูงสุดคือ 255 สีเขียวและสีน้ำเงินมีค่า 0 ดังนั้น เม็ดสีที่แสดงออกมาบนจอภาพจะเป็นสีแดง เมื่อจุดสีมีค่า (0,255,0) จะหมายถึง Green มีค่าสูงสุดคือ 255 ส่วนสีแดงและสีน้ำเงินจะมีค่า 0 ดังนั้น เม็ดสีที่แสดงออกมาบนจอภาพจะเป็นสีเขียว และเมื่อจุดสีมีค่า (0,0,255) จะหมายถึง Blue มีค่าสูงสุดคือ 255 ส่วนสีแดงและสีเขียวจะมีค่า 0 ดังนั้น เม็ดสีที่แสดงออกมาบนจอภาพจะเป็นสีน้ำเงิน

11.3 ค่าตัวเลขในแต่ละพิกเซลของภาพ RGB

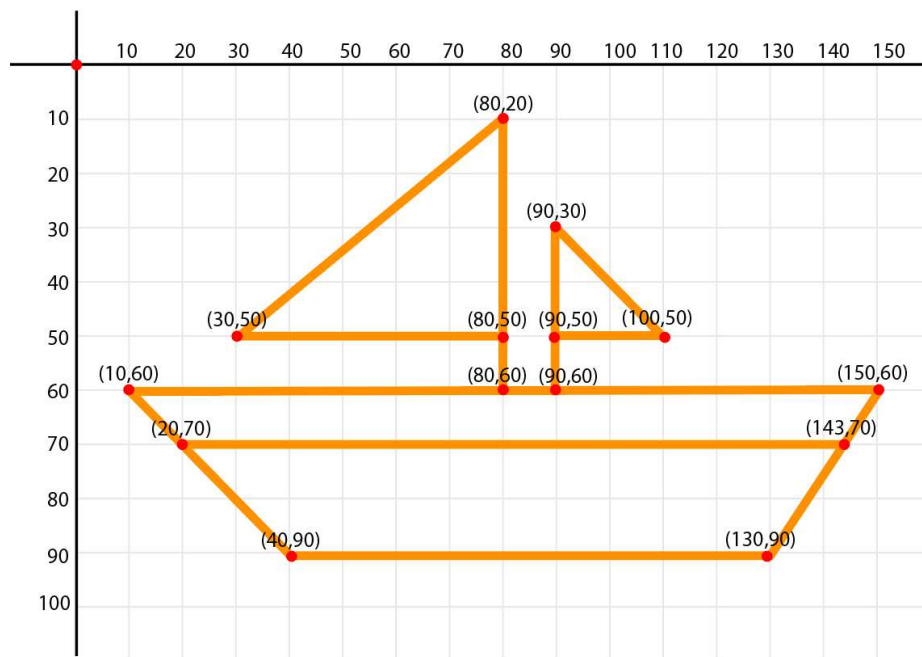


ภาพประกอบที่ 11.2 แสดงค่าสีภายในภาพ RGB

จากภาพด้านบนจุดสีที่ขยายขึ้นมามีจำนวน 9 จุดสี โดยมีค่าเรียงลำดับดังต่อไปนี้ (162,19,28), (244,171,97), (255,208,118), (244,208,118), (162,19,28), (244,171,97) , (237,248,255) (215,136,129) และ (207,30,0) ตามลำดับ

11.4 การเขียนจุด เส้นตรงและวงกลม (Output primitive)

องค์ประกอบพื้นฐานของภาพ (output primitive) ได้แก่ จุด เส้นตรง และวงกลม ในการวาดวัตถุใด ๆ บนจอภาพคอมพิวเตอร์จะเกิดขึ้นจากการนำองค์ประกอบพื้นฐานมาทำการวาดและรวมเข้าด้วยกัน โดยพล็อตลงไปบน Coordinate System ซึ่งเป็นระบบการบอกตำแหน่ง ซึ่งมีอยู่หลายระบบ ในหัวข้อนี้จะบอกด้วยระบบ Cartesian Coordinate System (พิกัด x,y)



ภาพประกอบที่ 11.3 การแสดงตำแหน่งพิกัดของโครงสร้างภาพสองมิติ

ตัวอย่างที่ 11.1 การเขียนโปรแกรมวาดรูปทรงสองพื้นฐานสองมิติ

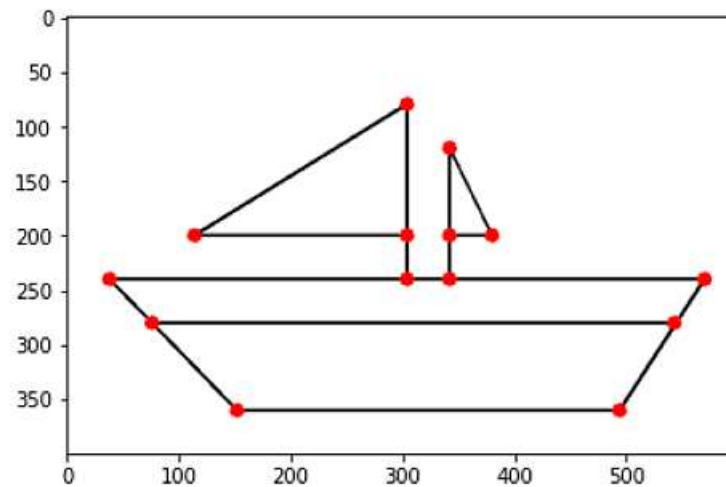
```
%pylab inline
import numpy
import cv2

img = numpy.ones((400,600,3),numpy.uint8)*255
x = [10,20,40,130,143,150, 80,80,80,30,90,90,90,100]
y = [60,70,90,90,70,60,60,50,20,50, 60,50,30,50]
sx = 3.8 ; sy = 4
X = list(map(lambda a:a*sx, x))
Y = list(map(lambda a:a*sy, y))
def line(p1,p2):
    cv2.line(img, (int(X[p1]),int(Y[p1])), (int(X[p2]),int(Y[p2])), (0,0,0), 2)

L = [[0,2],[2,3],[3,5],[0,5], [6,8], [8,9], [9,7], [10,12], [12,13], [13,11], [1,4]]
for i in L:
    line(i[0],i[1])

r = 3 ; thick = 5 ; c = (255,0,0)
for i in range(len(x)):
    img = cv2.circle(img, (int(X[i]),int(Y[i])), r, c, thick)
plt.imshow(img);plt.show()
```

ผลลัพธ์ :



ภาพประกอบที่ 11.4 แสดงผลลัพธ์การวาดรูปทรงพื้นฐานสองมิติ

ตัวอย่างที่ 11.1 คำสั่ง `%pylab inline` เป็นการขอใช้ไลบรารี `numpy` และ `matplotlib` เป็นการเขียนให้สั้นโดยไม่ต้องประกาศ `import numpy as np` และ `import matplotlib` ส่วนคำสั่ง `img = numpy.ones((400,600,3),numpy.uint8)*255` คือ การสร้างภาพขนาด 400x600 แบบ RGB (เลข 3 คือ สีสามสี) ส่วนการคูณด้วย 255 คือ ทำให้มีค่าทั้งหมดเท่ากับ 255 ซึ่งหมายถึงภาพสีขาว หากคูณด้วย 0 จะเป็นภาพสีดำ การประกาศตัวแปร `x` และ `y` จะใช้เก็บพิกัดของจุดยอดของเส้น การประกาศ `sx` และ `sy` คือ อัตราการขยายตำแหน่งจุดในแนวนอนและแนวตั้ง ตัวแปร `X` และ `Y` จะเก็บรายการพิกัดที่ถูกขยายด้วยสัดส่วน `sx` และ `sy` ฟังก์ชัน `line()` ใช้เขียนเส้นตรงระหว่างจุดสองจุด ตัวแปร `L` สำหรับเก็บลำดับของเส้นตรงที่ประกอบขึ้นเป็นภาพเรือ ตัวแปร `r c` และ `thick` คือรัศมี สี และความหนาของวงกลมที่เขียนลงบนจุดยอดของเส้นตรงแต่ละเส้น และคำสั่ง `imshow()` ใช้แสดงภาพตัวแปร `img`

11.5 การบวกและลบค่าสี

คำสั่ง `add()` และ `subtract()` ใช้สำหรับบวกและลบค่าสี

ตัวอย่างที่ 11.2 การเขียนโปรแกรมบวกและลบค่าสี

```
%pylab inline
import cv2
!wget "http://dsdi.msu.ac.th/articles/programming/lena.jpg"
lena = cv2.imread("lena.jpg")

matrix = np.ones(lena.shape, dtype = "uint8") * 100
added = cv2.add(lena, matrix)
matrix = np.ones(lena.shape, dtype = "uint8") * 50
```

```

subtracted = cv2.subtract(lena, matrix)
lena = cv2.cvtColor(lena, cv2.COLOR_RGB2BGR)
added = cv2.cvtColor(added, cv2.COLOR_RGB2BGR)
subtracted = cv2.cvtColor(subtracted, cv2.COLOR_RGB2BGR)
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,3,1)
ax.imshow(lena)
ax.set_title('Original')

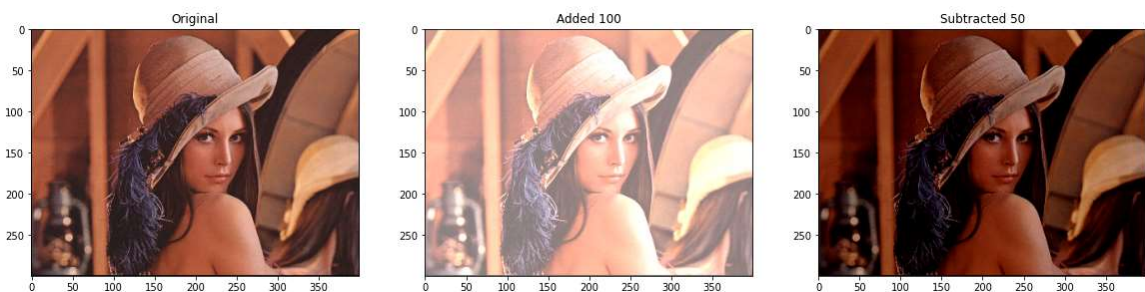
ax1 = fig.add_subplot(1,3,2)
ax1.imshow(added)
ax1.set_title('Added 100')

ax2 = fig.add_subplot(1,3,3)
ax2.imshow(subtracted)
ax2.set_title('Subtracted 50')

plt.show()

```

ผลลัพธ์ :



ภาพประกอบที่ 11.5 แสดงผลลัพธ์การบวกและลบค่าสี

ตัวอย่างที่ 11.2 คำสั่ง `wget` เป็นการดาวน์โหลดภาพ `lena.jpg` จากเว็บ `dsdi.msu.ac.th` ตัวแปร `lena` เป็นการโหลดภาพ `lena.jpg` ด้วยคำสั่ง `imread()` ตัวแปร `matrix` ใช้สร้างเมตริกซ์ขนาดเท่ากับภาพ `lena` ให้สมาชิกทุกตัวมีค่าเท่ากับ 100 ตัวแปร `added` จะเป็นการนำจุดสีในภาพ `lena` มาบวกด้วย 100 ทุก ๆ จุดทำให้ภาพมีความสว่างเพิ่มขึ้น ต่อมาตัวแปร `matrix` กำหนดให้มีค่าเท่ากับ 50 และมาลบกับภาพ `lena` และเก็บในตัวแปร `subtracted` ทำการแปลงสีภาพจาก RGB เป็น BGR และพล็อตด้วย `add_subplot(1,3,1)` คือ แกวที่ 1 ในจำนวน 3 ภาพและเป็นการแสดงภาพที่ 1 ทำไปจนครบทั้ง 3 ภาพ จากนั้นใช้คำสั่ง `plt.show()`

11.6 การดำเนินการเกี่ยวกับบิต (Bit)

การดำเนินการเกี่ยวกับบิต ประกอบด้วยคำสั่ง `bitwise_and()` , `bitwise_or()`, `bitwise_xor()` และ `bitwise_not()` โดยที่โอเปอเรเตอร์ AND OR และ Exclusive OR เขียนแทนด้วย `&` | และ `^` ตามลำดับ

ตัวอย่างที่ 11.3 การดำเนินการเกี่ยวกับบิต

```
%pylab inline
import cv2

rectangle = np.zeros((300, 300), dtype = "uint8")
cv2.rectangle(rectangle, (25, 25), (275, 275), 255, -1)
circle = np.zeros((300, 300), dtype = "uint8")
cv2.circle(circle, (150, 150), 150, 255, -1)

rectangle = cv2.cvtColor(rectangle, cv2.COLOR_BGR2RGB)
circle = cv2.cvtColor(circle, cv2.COLOR_BGR2RGB)

bitwise_and = cv2.bitwise_and(rectangle, circle)
bitwise_or = cv2.bitwise_or(rectangle, circle)
bitwise_xor = cv2.bitwise_xor(rectangle, circle)
bitwise_not = cv2.bitwise_not(rectangle)

fig = plt.figure(figsize=(20,20))
ax = fig.add_subplot(1,6,1)
ax.imshow(rectangle)
ax.set_title('rectangle')

ax1 = fig.add_subplot(1,6,2)
ax1.imshow(circle)
ax1.set_title('circle')

ax2 = fig.add_subplot(1,6,3)
ax2.imshow(bitwise_and)
ax2.set_title('and')

ax3 = fig.add_subplot(1,6,4)
ax3.imshow(bitwise_or)
```

```

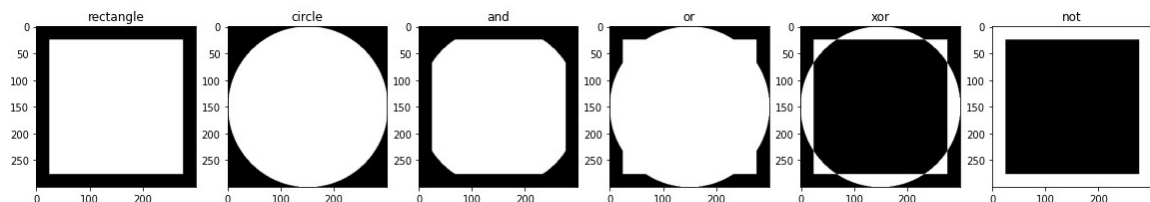
ax3.set_title('or')
ax4 = fig.add_subplot(1,6,5)
ax4.imshow(bitwise_xor)
ax4.set_title('xor')

ax5 = fig.add_subplot(1,6,6)
ax5.imshow(bitwise_not)
ax5.set_title('not')

plt.show()

```

ผลลัพธ์ :



ภาพประกอบที่ 11.6 แสดงการดำเนินการเกี่ยวกับบิต

ตัวอย่างที่ 11.3 ตัวแปร `rectangle` เป็นภาพสี่เหลี่ยมมีขนาด 300x300 พิกเซล และเขียนสี่เหลี่ยมที่พิกัด (25,25) ไปยัง (275,275) ด้วยสีขาว ตัวแปร `circle` มีขนาด 300x300 พิกเซลและวาดวงกลมที่จุดศูนย์กลางที่พิกัด (150,150) ความยาวรัศมี 150 พิกเซลระบายวงกลมด้วยสีขาว ตัวแปร `bitwise_and` จะทำการ `and` ระหว่างภาพ `rectangle` และ `circle` ตัวแปร `bitwise_or` จะทำการ `Or` ระหว่าง `rectangle` และ `circle` ส่วน `bitwise_xor` จะทำการ `exclusive or` ระหว่าง `rectangle` กับ `circle` ตัวแปร `bitwise_not` เป็นการกลับสีของ `rectangle` เมื่อพล็อตกราฟแสดงภาพได้ดังตัวอย่างด้านบน

11.7 การแยกและรวมสี RGB

คำสั่ง `split()` ใช้สำหรับแยกสีออกจากกัน และคำสั่ง `merge()` ใช้สำหรับรวมสีเข้าด้วยกัน

ตัวอย่างที่ 11.4 การแยกและรวมสีภาพด้วย `split()` และ `merge()`

```

%pylab inline
!wget "http://dsdi.msu.ac.th/articles/programming/lena.jpg"

import cv2
img = cv2.imread("lena.jpg")
(blue, green, red) = cv2.split(img)
black = np.zeros((img.shape[0], img.shape[1]), dtype = "uint8")

```

```

Blue = cv2.merge([blue, black, black])
Green = cv2.merge([black, green, black])
Red = cv2.merge([black, black, red])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
Red = cv2.cvtColor(Red, cv2.COLOR_BGR2RGB)
Green = cv2.cvtColor(Green, cv2.COLOR_BGR2RGB)
Blue = cv2.cvtColor(Blue, cv2.COLOR_BGR2RGB)

fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,4,1)
ax.imshow(img)
ax.set_title('Original')

ax1 = fig.add_subplot(1,4,2)
ax1.imshow(Red)
ax1.set_title('Red')

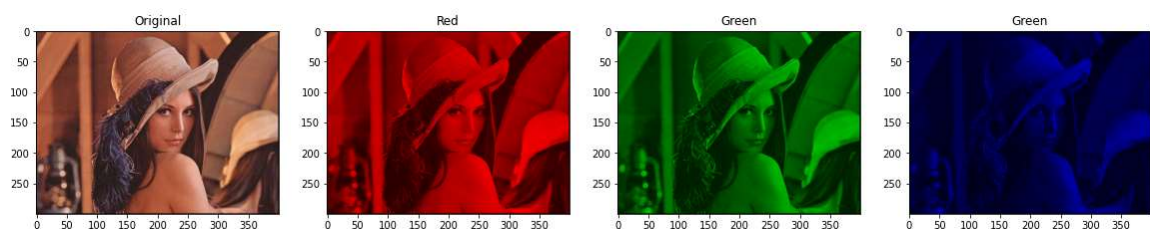
ax2 = fig.add_subplot(1,4,3)
ax2.imshow(Green)
ax2.set_title('Green')

ax3 = fig.add_subplot(1,4,4)
ax3.imshow(Blue)
ax3.set_title('Blue')

plt.show()

```

ผลลัพธ์ :



ภาพประกอบที่ 11.7 แสดงการแยกและรวมสีภาพด้วย split() และ merge()

ตัวอย่างที่ 11.4 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง wget ทำการโหลดภาพ lena.jpg จากเว็บ dsdi.msu.ac.th จากนั้นอิมพอร์ต cv2 เป็นไลบรารีด้าน computer vision ตัวแปร img ทำการโหลดภาพ lena ด้วยคำสั่ง imread() และตัวแปร blue, gree, red เป็นสีที่ได้จากการแยกสีจากตัวแปร img ซึ่งเป็นภาพ lena ตัวแปร black สร้างขึ้นสำหรับค่าสีดำขนาดเท่ากับภาพ lena เพื่อจะนำมารวมสีด้วยคำสั่ง merge() ตัวแปร Blue คือรวมสี blue,black,black ตัวแปร Green คือการรวมสี black, green, black และตัวแปร Red เป็นการรวมสี black, black, red เมื่อรวมภาพแล้วนำมาแสดงผลได้ผลลัพธ์ดังตัวอย่างด้านบน

11.8 การกลับสีภาพ (Invert Color)

โมเดลสีแบบ RGB ประกอบด้วยสีแดง เขียว น้ำเงิน โดยแต่ละสีใช้ 8 บิต ทำให้มีความแตกต่างกัน 256 ความแตกต่าง (0-255) สูตรสำหรับการกลับสี (Invert Color) คือ $R' = 256 - 1 - R$, $G' = 256 - 1 - G$ และ $B' = 256 - 1 - B$

ไลบรารี Pillow ใช้สำหรับการจัดการภาพเป็นอีกหนึ่งไลบรารีที่อนุญาตให้เข้าถึงจุดสีภายในภาพ ด้วยคำสั่ง putpixel() และ getpixel()ติดตั้งด้วยคำสั่ง !pip install Pillow

ตัวอย่างที่ 11.5 การกลับสีภาพ (invert color)

```
%pylab inline
from PIL import Image
import cv2

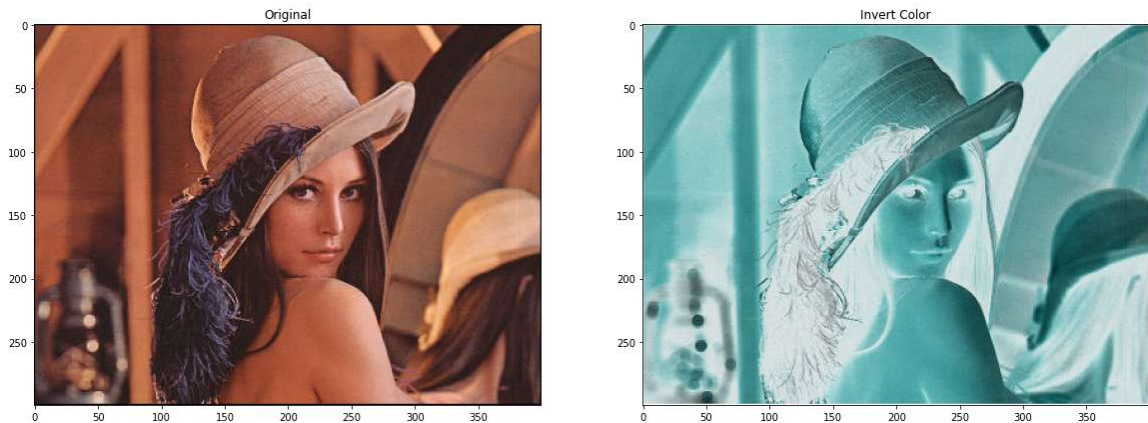
!wget "http://dsdi.msu.ac.th/articles/programming/lena.jpg"
img1 = Image.open("lena.jpg").convert("RGB")
img2 = Image.new("RGB", (img1.size[0], img1.size[1]), "white")
for x in range(0, img1.size[0]):
    for y in range(0, img1.size[1]):
        r,g,b = img1.getpixel((x,y))
        red = 255-r
        green = 255-g
        blue = 255-b
        img2.putpixel((x,y),(red,green,blue))

fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,2,1)
ax.imshow(img1)
ax.set_title('Original')

ax1 = fig.add_subplot(1,2,2)
```

```
ax1.imshow(img2)
ax1.set_title('Invert Color')
plt.show()
```

ผลลัพธ์ :



ภาพประกอบที่ 11.8 แสดงการกลับสีภาพ (Invert color)

ตัวอย่างที่ 11.5 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง from PIL import Image สำหรับจัดการภาพด้วยไลบรารี PIL และ import cv2 อิมพอร์ตไลบรารี opencv คำสั่ง wget สำหรับอ่านภาพจากเว็บ ตัวแปร img1 เก็บภาพ lena และคำสั่ง img2 สร้างภาพสี่เหลี่ยมขนาดเท่ากับภาพลิ้นา การวนเข้าไปในภาพในแนวนอนและแนวตั้ง จากนั้นอ่านค่าสีด้วยคำสั่ง r,g,b = getpixel(พิกัดx, พิกัดy) จากนั้นนำสีแดง, เขียว และน้ำเงินมาลบด้วย 255 แล้ว putpixel ลงไปในตัวแปร img2 ด้วยค่าสีใหม่ที่ได้จากการลบด้วย 255 นำไปพล็อตภาพจะได้ดังภาพด้านบน

11.9 การเขียนโปรแกรมเพื่อปรับค่าแกมมาของภาพ

gamma คือการปรับโทนสีเข้มมีค่าระหว่าง 0 - 1 สมการขดเซยค่าแกมมา คือ $R' = R^{\frac{1}{\gamma}}$
 $G = G^{\frac{1}{\gamma}}$ และ $B' = B^{\frac{1}{\gamma}}$

ตัวอย่างที่ 11.6 การปรับค่าแกมมา

```
%pylab inline
from PIL import Image
img1 = Image.open("lena.jpg").convert("RGB")
gamma = 0.8
img2 = Image.new("RGB", (img1.size[0], img1.size[1]), "white")
for x in range(0, img1.size[0]):
    for y in range(0, img1.size[1]):
        r,g,b = list(img1.getpixel((x,y)))
```

```

    red = int(r**(1/gamma))
    green = int(g**(1/gamma))
    blue = int(b**(1/gamma))
    img2.putpixel((x,y),(red,green,blue))
gamma = 0.7
img3 = Image.new("RGB", (img1.size[0], img1.size[1]), "white")
for x in range(0, img1.size[0]):
    for y in range(0, img1.size[1]):
        r,g,b = list(img1.getpixel((x,y)))
        red = int(r**(1/gamma))
        green = int(g**(1/gamma))
        blue = int(b**(1/gamma))
        img3.putpixel((x,y),(red,green,blue))

fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,3,1)
ax.imshow(img1)
ax.set_title('Original')

ax1 = fig.add_subplot(1,3,2)
ax1.imshow(img2)
ax1.set_title('Gamma 0.8')

ax2 = fig.add_subplot(1,3,3)
ax2.imshow(img3)
ax2.set_title('Gamma 0.7')
plt.show()

```

ผลลัพธ์ :



ภาพประกอบที่ 11.9 แสดงผลลัพธ์การปรับค่าแกมมาของภาพ

ตัวอย่างที่ 11.6 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง from PIL import Image สำหรับจัดการภาพด้วยไลบรารี PIL ตัวแปร img1 เก็บภาพ lena ตัวแปร gamma กำหนดให้มีค่า 0.8 ตัวแปร img2 สร้างภาพสีขาขนาดเท่าภาพ lena จากนั้นวนเข้าไปในแนวนอนและแนวตั้งของภาพ ทำการอ่านค่าสีด้วยคำสั่ง getpixel ที่ตำแหน่ง x,y เก็บไว้ในตัวแปร r,g,b จากนั้นแปลงแกมมาด้วย สูตร $R' = R^{\frac{1}{\gamma}}$ $G = G^{\frac{1}{\gamma}}$ และ $B' = B^{\frac{1}{\gamma}}$ แล้วทำการเขียนค่าสีใส่ลงในตัวแปร img2 จนครบทุกจุดสี จากนั้นวาดภาพด้วยคำสั่ง imshow() จะได้ผลลัพธ์ดังภาพด้านบน

11.10 การเลือกดำเนินการบางส่วนภายในภาพ

การระบายพื้นที่ภายในภาพจะระบุช่วงตำแหน่งและกำหนดค่าใหม่ด้วยสีที่ต้องการ เช่น `img[y:y+h, x:x+w] = (0,255,0)`

ตัวอย่างที่ 11.7 การดำเนินการบางส่วนภายในภาพ

```
%pylab inline
import cv2
img = cv2.imread("lena.jpg")
x = 200; y = 130;
w = 60; h = 10
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img2 = img.copy()
img2[y:y + h, x : x + w] = (0,255,0)
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,2,1)
ax.imshow(img)
ax.set_title('Original')

ax1 = fig.add_subplot(1,2,2)
ax1.imshow(img2)
ax1.set_title('Area Fill')

plt.show()
```

ผลลัพธ์ :



ภาพประกอบที่ 11.10 แสดงการเข้าถึงพื้นที่บางส่วนภายในภาพ

ตัวอย่างที่ 11.7 คำสั่ง `%pylab inline` สำหรับขอใช้งาน `numpy` และ `matplotlib` คำสั่ง `import cv2` ขอใช้คำสั่ง `opencv` จากนั้นอ่านไฟล์ `img` ด้วยคำสั่ง `imread()` กำหนดค่า `x=200, y=130, w=60, h=10` คือ ที่พิกัด `(200,130)` ให้ระบายสี `(0,255,0)` ลงในพื้นที่ดังกล่าว ด้วยคำสั่ง `img2[y:y + h, x : x + w] = (0,255,0)` จากนั้นพล็อตภาพด้วยคำสั่ง `imshow()` ผลลัพธ์จะเป็นการระบายสีเขียวลงในพื้นที่ดังกล่าว ดังภาพด้านบน

11.11 การกลับภาพซ้ายขวา/บนล่าง

ฟังก์ชัน `flip()` ใช้สำหรับการกลับภาพโดยพารามิเตอร์การกลับภาพระบุด้วยตัวเลข ดังนี้ 1 คือ ซ้าย/ขวา 0 คือกลับด้านบนล่าง -1 คือ กลับซ้ายขวาและบนล่าง

ตัวอย่างที่ 11.8 การกลับภาพซ้ายขวา/บนล่างด้วยคำสั่ง `flip()`

```
%pylab inline
import cv2
img = cv2.imread("lena.jpg")
a = cv2.flip(img, 1)
b = cv2.flip(img, 0)
c = cv2.flip(img, -1)

img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
a = cv2.cvtColor(a, cv2.COLOR_RGB2BGR)
b = cv2.cvtColor(b, cv2.COLOR_RGB2BGR)
c = cv2.cvtColor(c, cv2.COLOR_RGB2BGR)

fig = plt.figure(figsize=(10,5))
```

```

ax = fig.add_subplot(1,4,1)
ax.imshow(img)
ax.set_title('Original')

ax1 = fig.add_subplot(1,4,2)
ax1.imshow(a)
ax1.set_title('flip 1')

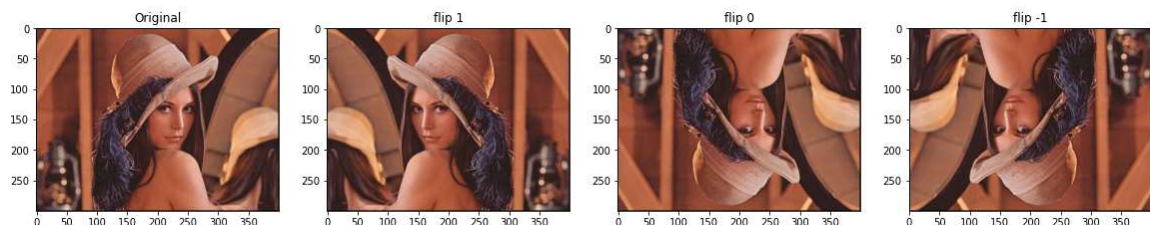
ax2 = fig.add_subplot(1,4,3)
ax2.imshow(b)
ax2.set_title('flip 0')

ax3 = fig.add_subplot(1,4,4)
ax3.imshow(c)
ax3.set_title('flip -1')

plt.show()

```

ผลลัพธ์ :



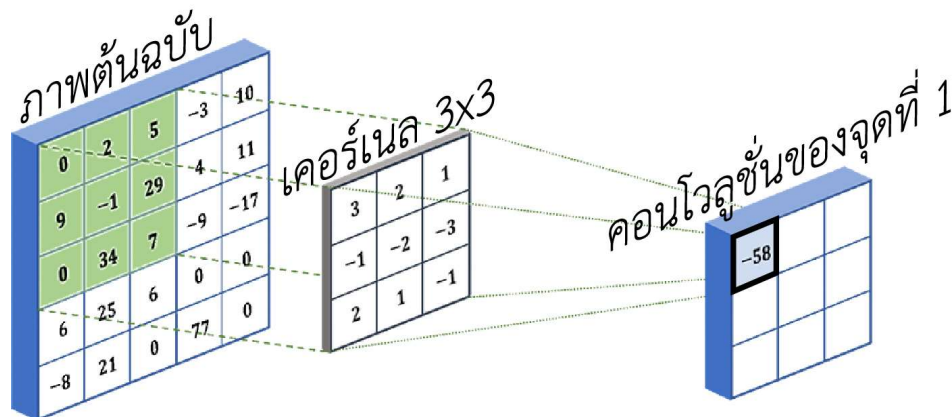
ภาพประกอบที่ 11.11 แสดงการกลับภาพซ้ายขวา/บนล่างด้วยคำสั่ง flip()

ตัวอย่างที่ 11.8 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง import cv2 เป็นการขอใช้ opencv ตัวแปร img อ่านภาพ lena ด้วยคำสั่ง imread() กำหนดให้ตัวแปร a,b,c เป็นการ flip() ภาพ lena ด้วยพารามิเตอร์ 1, 0 และ -1 จากนั้นนำภาพที่ได้ไปแปลงสีจาก BGR เป็น RGB เนื่องจาก OpenCV จะเปิดภาพด้วยโหมด BGR (Blue Green Red) จากนั้นพล็อตภาพจะเป็นการกลับภาพในทิศทางต่าง ๆ ดังภาพด้านบน

11.12 คอนโวลูชันและเคอร์เนล (Convolution และ Kernel)

คำสั่ง filter2D() เป็นคำสั่งสำหรับการทำคอนโวลูชันระหว่างเคอร์เนลกับภาพ ขอให้พิจารณาภาพด้านล่างต่อไปนี้ ซึ่งประกอบด้วย ภาพต้นฉบับ เคอร์เนลขนาด 3x3 และภาพที่ผ่านการคอนโวลูชัน การคอนโวลูชันเป็นการนำเคอร์เนลไปหาบลงบนภาพทีละจุดแล้วทำการรวมผลรวมของการ dot

product ได้ผลลัพธ์เป็นตัวเลขสเกลาร์หนึ่งตัวจากนั้นนำไปสร้างเป็นจุดสีของผลลัพธ์ วาดจากซ้ายไปขวาและบนลงล่าง จากตัวอย่างนี้ เคอร์เนลมีค่า $\begin{bmatrix} 3 & 2 & 1 \\ -1 & -2 & -3 \\ 2 & 1 & -1 \end{bmatrix}$ เมื่อนำไปวางทับกับภาพต้นฉบับแล้วคำนวณผลรวมของ dot product จะมีค่า $(0 \times 3) + (2 \times 2) + (5 \times 1) + (9 \times -1) + (-1 \times -2) + (29 \times -3) + (0 \times 2) + (34 \times 1) + (7 \times -1) = (0, 4, 5, -9, 2, -87, 0, 34, -7) = -58$ จากตัวอย่างนี้ผลลัพธ์การทำคอนโวลูชันกับเคอร์เนลมีค่าเท่ากับ -58 ให้เลื่อนเคอร์เนลไปทางขวาและลงด้านล่างจนครบจะได้ผลลัพธ์เป็นภาพที่ผ่านการคอนโวลูชัน



ภาพประกอบที่ 11.12 แสดงการคอนโวลูชัน

ตัวอย่างที่ 11.9 การคอนโวลูชันด้วยเคอร์เนลแบบต่าง ๆ

```
%pylab inline
import cv2
!wget "http://dsdi.msu.ac.th/articles/programming/lena.jpg"

img = cv2.imread('lena.jpg')

sobel_kernel_y = np.array([[+1, +2, +1], [0, 0, 0], [-1, -2, -1]])
img1 = cv2.filter2D(src=img, ddepth=-1, kernel=sobel_kernel_y)

sharpen_kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
img2 = cv2.filter2D(src=img, ddepth=-1, kernel=sharpen_kernel)

blur_kernel = np.array([[0.1, 0.1, 0.1], [0.1, 0.1, 0.1], [0.1, 0.1, 0.1]])
img3 = cv2.filter2D(src=img, ddepth=-1, kernel=blur_kernel)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
```

```

img3 = cv2.cvtColor(img3, cv2.COLOR_BGR2RGB)
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(1,4,1)
ax.imshow(img)
ax.set_title('Original')

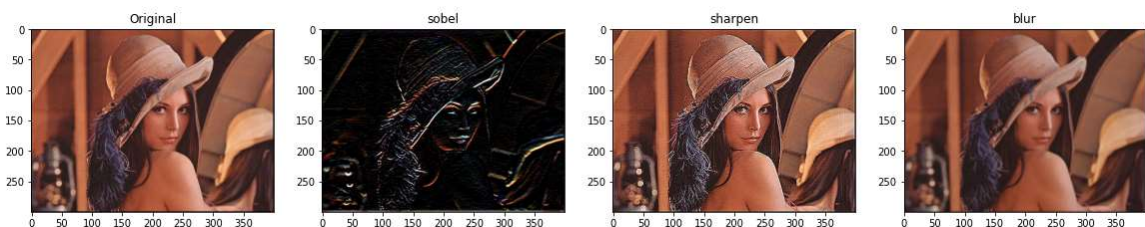
ax1 = fig.add_subplot(1,4,2)
ax1.imshow(img1)
ax1.set_title('sobel')

ax2 = fig.add_subplot(1,4,3)
ax2.imshow(img2)
ax2.set_title('sharpen')

ax3 = fig.add_subplot(1,4,4)
ax3.imshow(img3)
ax3.set_title('blur')
plt.show()

```

ผลลัพธ์ :



ภาพประกอบที่ 11.13 แสดงการคอนโวลูชันด้วยเคอร์เนลแบบต่าง ๆ

ตัวอย่างที่ 11.9 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง Import cv2 เป็นการขอใช้ opencv คำสั่ง wget เป็นการโหลดภาพจากเว็บ ตัวแปร img เป็นการโหลดภาพ lena ด้วยคำสั่ง imread() ตัวแปร sobel_kernel_y เป็นเคอร์เนล ขนาด 3x3 มีค่า $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ จากนั้นทำคอนโวลูชันด้วยคำสั่ง filter2D() และเก็บในตัวแปร img1 ตัวแปร sharpen_kernel เป็นเคอร์เนล 3x3 มีค่า $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ นำไปคอนโวลูชันกับภาพ lena และเก็บไว้ในตัวแปร img2 จากนั้น blur_kernel เป็นเคอร์เนลขนาด 3x3 มีค่า $\begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}$ นำไปคอนโวลูชันกับภาพ img และเก็บผลลัพธ์ใน img3 จากนั้นนำภาพทั้งหมดพล็อตจะได้ผลลัพธ์ดังภาพด้านบน

11.13 การตรวจจับใบหน้าด้วย Haar Cascade

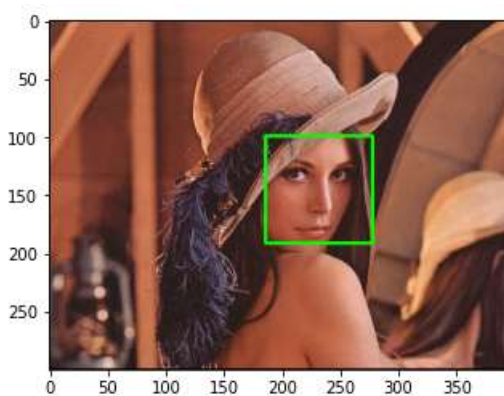
Haar Cascades เป็นการตรวจจับใบหน้า เผยแพร่ในปี พ.ศ. 2544 เป็นการตรวจจับใบหน้าภายในภาพ โดย Viola และ Jones ไบรารี OpenCV เป็นเครื่องมือเกี่ยวกับการประมวลผลภาพได้รวมอัลกอริทึมตรวจจับใบหน้าของ Haar Cascades เอาไว้ในไฟล์ haarcascade_frontalface_default.xml

ตัวอย่างที่ 11.10 การตรวจจับใบหน้าด้วย Haar Cascades

```
%pylab inline
import cv2

!wget "https://dsdi.msu.ac.th/programming/haarcascade_frontalface_default.xml"
!wget "http://dsdi.msu.ac.th/articles/programming/lena.jpg"
model = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
img = cv2.imread("lena.jpg")
img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
faces = model.detectMultiScale(img, 1.3, 5)
for (x, y, w, h) in faces :
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
plt.imshow(img)
```

ผลลัพธ์ :



ภาพประกอบที่ 11.14 แสดงการตรวจจับใบหน้าด้วย Haar Cascade

ตัวอย่างที่ 11.10 คำสั่ง %pylab inline เป็นการขอใช้ numpy และ matplotlib คำสั่ง Import cv2 เป็นการขอใช้ opencv คำสั่ง wget เป็นการโหลดไฟล์จากเว็บ ในตัวอย่างนี้โหลดไฟล์ haarcascade_frontalface_default.xml เป็นโมเดลในการตรวจจับใบหน้า ตัวแปร model ใช้สำหรับตรวจจับใบหน้าโดยสร้างจากคำสั่ง CascadeClassifier() โดยโหลดค่าน้ำหนักจากไฟล์ xml จากนั้นทำการโหลดภาพ lena ด้วยคำสั่ง imread() และทำการตรวจจับด้วยคำสั่ง detectMultiScale() ผลลัพธ์จะ

เก็บไว้ในตัวแปร faces เมื่อวนเข้าไปในตัวแปรดังกล่าวจะทำการเขียนสีเหลี่ยมครอบคลุมพื้นที่ใบหน้าด้วยตำแหน่ง x,y,w,h โดยวาดสีเขียว (0,255,0) และความหนาเส้น 2 พิกเซล แสดงผลลัพธ์ได้ดังภาพด้านบน

11.14 การตรวจจับใบหน้าขนาดเล็กด้วย Tinyface

Tiny Face พัฒนาโดย Peiyun Hu, Deva Ramanan เป็นการค้นหาใบหน้าที่มีขนาดเล็ก 25x20 พิกเซลทำได้แม่นยำมาก ถ้าใช้เทมเพลตของใบหน้าที่ใหญ่ 2 เท่า ซึ่งคือ 50x40 พิกเซลจะใช้เทมเพลตขนาด 25x20 ส่วนการค้นหาใบหน้าที่ใหญ่ 250x200 พิกเซลจะแม่นยำหากใช้เทมเพลตที่เล็กลง 2 เท่า คือ 125x100 พิกเซล

- 1) เปิด colab เพื่อใช้ python version 2 ด้วยลิงค์ <https://colab.research.google.com/notebook#create=true&language=python2>
- 2) เลือก Runtime -> Change Runtime Type -> เลือก Python2 และ Hardware -> เลือก GPU -> กดปุ่ม Save
- 3) เขียนโค้ด python ต่อไปนี้

ตัวอย่างที่ 11.11 การตรวจจับใบหน้าขนาดเล็กด้วย Tinyface

```
!wget "http://dsdi.msu.ac.th/programming/tiny/tiny_face_model.py"
!wget "http://dsdi.msu.ac.th/programming/tiny/util.py"
!wget "http://dsdi.msu.ac.th/programming/tiny/tiny_face_eval4nb.py"
!wget "http://dsdi.msu.ac.th/programming/tiny/tinyface.pkl"

# -*- coding: utf-8 -*-
%tensorflow_version 1.x
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
import tiny_face_model
import util
code = open("tiny_face_eval4nb.py","r").read()
exec(code)
!mkdir input
!mkdir output
!wget "http://dsdi.msu.ac.th/programming/tiny/ex02.jpg" -O input/ex02.jpg

with tf.Graph().as_default():
    z = evaluate("tinyface.pkl","input", "output")
```

ตัวอย่างที่ 11.11 คำสั่ง wget เป็นการดาวน์โหลดไฟล์ tiny_face_model.py , util.py , tiny_face_eval4nb.py และ tinyface.pkl ซึ่งอยู่บนเว็บ dsdi.msu.ac.th จากนั้น คำการขอใช้ tensorflow_version 1.x และทำการ disable_v2_behavior() เนื่องจาก tinyface เวอร์ชันนี้พัฒนา

ด้วย tensorflow 1.x ทำการอิมพอร์ต tiny_face_model และ util จากนั้นโหลดไฟล์ tiny_face_eval4nb.py และทำการ exec() หลังจากนั้นสร้างโฟลเดอร์ input และ output ด้วยคำสั่ง !mkdir โหลดภาพ ex02.jpg ด้วยคำสั่ง !wget โดยบันทึกไฟล์ไว้ในโฟลเดอร์ input ตั้งชื่อ ex02.jpg

จากนั้นประมวลผลด้วยคำสั่ง with tf.Graph().as_default(): ให้เรียกคำสั่ง z = evaluate("tinyface.pkl", "input", "output") และใส่อาร์กิวเมนต์ 3 ตัว คือ ไฟล์โมเดล tinyface.pkl และโฟลเดอร์ input และ output โปรแกรมจะทำการอ่านไฟล์ในโฟลเดอร์ input แล้วจะตรวจจับภาพใบหน้าและเก็บผลลัพธ์ไว้ในโฟลเดอร์ output

ตัวอย่างที่ 11.12 การเขียนโปรแกรมแสดงภาพผลลัพธ์การประมวลผล Tinyface

```
%pylab inline
from IPython.display import display, Image
from glob import glob
fn = glob("output/*.jpg")
for i in fn:
    img= Image(i)
    display(img)
    print(i)
```

ตัวอย่างที่ 11.12 ผลลัพธ์เก็บไว้ในโฟลเดอร์ output ใช้คำสั่ง glob() เพื่อแสดงรายชื่อไฟล์ในโฟลเดอร์ คำสั่งในการแสดงผลคือ display และคำสั่ง Image ในไลบรารี lpython.display โดยตัวแปร fn จะเก็บรายชื่อไฟล์ที่ผ่านการประมวลผลและเก็บไว้ในโฟลเดอร์ output และนำมาแสดงผลด้วยคำสั่ง display() จะได้ผลลัพธ์ดังภาพต่อไปนี้

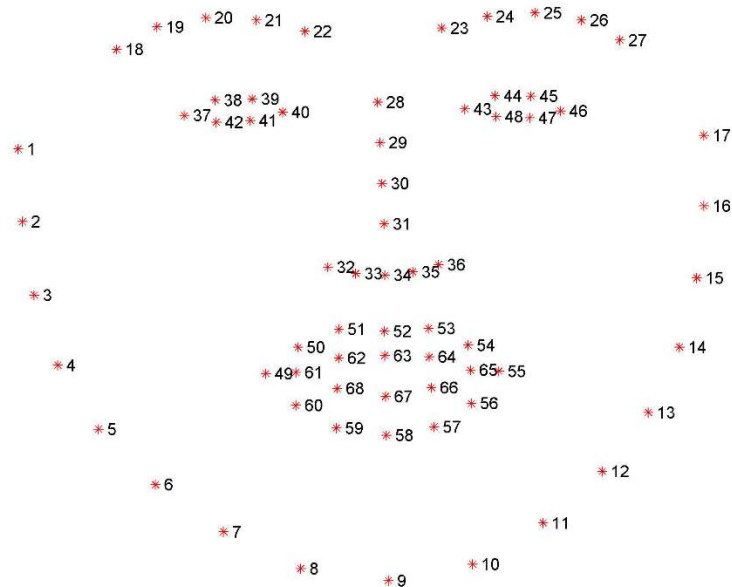
ผลลัพธ์:



ภาพประกอบที่ 11.15 แสดงการตรวจจับใบหน้าขนาดเล็กด้วย tinyface

11.15 การอ่านพิกัดแลนด์มาร์คใบหน้าด้วย dlib

ไลบรารี dlib ใช้สำหรับสกัดจุดตำแหน่งต่าง ๆ บนใบหน้า (แลนด์มาร์ค) ซึ่งประกอบด้วย 68 จุด ดังภาพต่อไปนี้



ภาพประกอบที่ 11.16 ตำแหน่งพิกัดของจุดบนใบหน้า (แลนด์มาร์ค)

ที่มา: <https://www.researchgate.net>

1. การติดตั้ง dlib ใช้คำสั่ง `!pip install dlib` หรือติดตั้งจากลิงค์ <https://pypi.org/project/dlib/>

```
!pip install dlib
```
2. ดาวน์โหลดโมเดลตรวจจับแลนด์มาร์คใบหน้าด้วยคำสั่ง `wget` ชื่อ `shape_predictor_68_face_landmarks.dat` และสามารถดาวน์โหลดได้จากเว็บอื่น ๆ ที่เป็นแหล่งข้อมูลได้เช่นกัน เมื่อดาวน์โหลดแล้วถูกเก็บไฟล์ไว้ในไดเรกทอรีปัจจุบัน มีขนาด 98MB

```
!wget "http://dsdi.msu.ac.th/programming/dlib/shape_predictor_68_face_landmarks.dat"
```
3. ดาวน์โหลดภาพ เจียเจีย ซึ่งเป็นหุ่นยนต์สาวจากประเทศจีน ด้วยคำสั่ง `wget` ดังนี้

```
!wget "http://dsdi.msu.ac.th/programming/dlib/humanoid-robot-Jiajia.jpg"
```

ตัวอย่างที่ 11.13 การเขียนโปรแกรมตรวจจับพิกัดจุดแลนด์มาร์คบนใบหน้าด้วย dlib

```
%pylab inline
import cv2, dlib
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
img = cv2.imread("humanoid-robot-Jiajia.jpg")
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```

rects = detector(img, 1)
if len(rects) > 1:
    print("มีใบหน้าหลายใบในภาพนี้")
elif len(rects) == 0:
    print("ไม่มีใบหน้าในภาพนี้")
else:
    points = numpy.matrix([[p.x, p.y] for p in predictor(img, rects[0]).parts()]).tolist()
    for i in range(len(points)):
        cv2.circle(img, tuple(points[i]), 2, color=(0, 255, 0), thickness=1)
    plt.figure(figsize = (10,10))
    plt.imshow(img);
    plt.show()

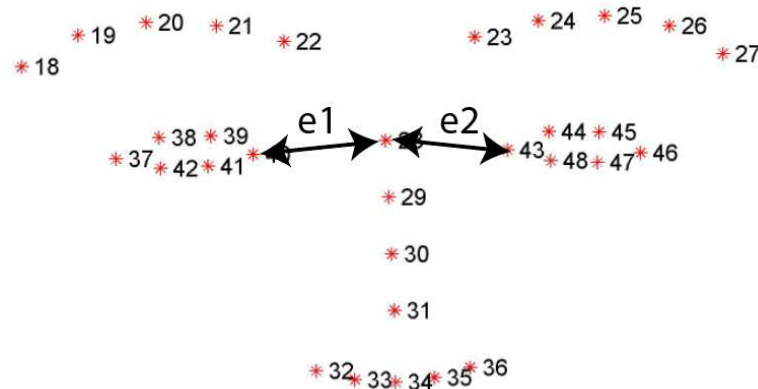
```

ตัวอย่างที่ 11.13 ขอใช้ไลบรารี opencv และ dlib ด้วยคำสั่ง import cv2, dlib ตัวแปร detector เป็นอ็อบเจกต์ของคลาส get_frontal_face_detector() ตัวแปร predictor เป็นอ็อบเจกต์ของคลาส shape_predictor() โดยคลาสนี้ต้องการอาร์กิวเมนต์ของโมเดลที่ชื่อว่า “shape_predictor_68_face_landmarks.dat” ตัวแปร img เป็นการโหลดภาพหุ่นยนต์สาวเจียเจีย ด้วยชื่อไฟล์ humanoid-robot-jiajia.jpg และแปลงจากภาพ BGR (น้ำเงิน เขียว แดง) เป็นภาพ RGB (แดง เขียว น้ำเงิน) เนื่องจาก opencv จะเปิดภาพด้วยโหมด BGR ตัวแปร rect เป็นผลลัพธ์ของการตรวจจับแลนด์มาร์คโดยส่งตัวแปร img ให้เป็นอาร์กิวเมนต์ของฟังก์ชัน detector() จากนั้นตรวจสอบความยาวของตัวแปร rects ว่ามีขนาดเท่าไร ถ้ามากกว่า 1 ให้พิมพ์ว่าภาพที่ตรวจจับพบใบหน้ามากกว่าหนึ่งใบ และตรวจสอบต่อว่ามีขนาดเท่ากับ 0 หรือไม่ถ้าเท่ากับศูนย์ให้พิมพ์บอกว่า “ไม่มีใบหน้าในภาพนี้” ตัวแปร points เป็นการวนเข้าไปใน rects และแปลงเป็นตัวแปรชนิดลิสต์ใช้เก็บตำแหน่งพิกัดของจุดบนใบหน้าทั้งหมด ในขั้นสุดท้ายเป็นการวนเข้าไปในตัวแปร points ที่ละจุดด้วยการสร้างตัวเลขที่มีความยาวเท่ากับจำนวนภายในตัวแปร points ด้วยคำสั่ง range(len(points)) และเขียนวงกลม ด้วยคำสั่ง circle() ป้อน img คือภาพ img และพิกัดจุดแต่ละจุดด้วยคำสั่ง tuple(points[i]) คือแปลงพิกัดให้อยู่ในรูปตัวแปรทูเพิล เลข 2 คือ ขนาดเส้นผ่านศูนย์กลางวงกลม color=(0,255,0) คือสีเขียว (R,G,B) และความหนาของเส้นวงกลมขนาด 1 พิกเซล ด้วยตัวแปร thicknetss=1

พล็อตภาพโดยกำหนดขนาดด้วยคำสั่ง figure(figsize=(10,10)) ซึ่งเป็นอัตราความกว้างและสูง จากนั้นแสดงภาพด้วยคำสั่ง imshow() และ show() ผลลัพธ์จะวาดจุดวงกลมบนพิกัดแลนด์มาร์คบนใบหน้าของหุ่นยนต์สาวเจียเจียออกมาดังภาพที่ xxxxx ให้ทดลองเปลี่ยนภาพอื่นตามที่ต้องการ

11.16 การตรวจสอบคนในภาพหันหน้าไปทางซ้ายหรือขวา

ในการตรวจสอบการหันหน้าซ้ายหรือขวาจะตรวจสอบจากพิกัดที่อยู่ระหว่างขอบตาซ้ายและขวามาเปรียบเทียบกัน ถ้ามีค่าใกล้เคียงกันแสดงว่าเป็นหน้าตรง แต่ถ้ามีขนาดที่แตกต่างกันมากแสดงว่าหันไปข้างใดข้างหนึ่ง โดยขนาดความต่างกันจะขึ้นอยู่กับความละเอียดของภาพ ถ้าละเอียดสูง ๆ แม้ค่าจะต่างกันเยอะมากแต่ก็ยังเป็นภาพหน้าตรงได้ ดังนั้น จึงควรกำหนดค่า threshold ที่บอกว่าระยะเท่าไรจะถือว่าเป็นการหันหน้าเอียงให้เหมาะสมกับขนาดภาพ



ภาพประกอบที่ 11.17 แสดงระยะห่างระหว่างตาซ้ายและขวา (e1 และ e2)

ตัวอย่างที่ 11. 14 การตรวจสอบว่าคนในภาพหันหน้าไปทางซ้ายหรือขวา

```
from scipy.spatial import distance
e1 = distance.euclidean(points[39],points[27])
e2 = distance.euclidean(points[42],points[27])
print("ระยะ e1 = %.2f และระยะ e2 = %.2f"%(e1,e2))
threshold = 2
if abs(e1-e2)> threshold:
    print("มีการเอียงหน้าเกิดขึ้น ยังไม่รู้ว่าจะเอียงไปทางไหน")
    if e1 < e2:
        print("รู้แล้ว.. คนในภาพหันหน้าไปทางซ้ายนั่นเอง")
    else:
        print("รู้แล้ว.. คนในภาพหันหน้าไปทางขวามันเอง")
else:
    print("ภาพนี้หน้าตรง")
```

ผลลัพธ์

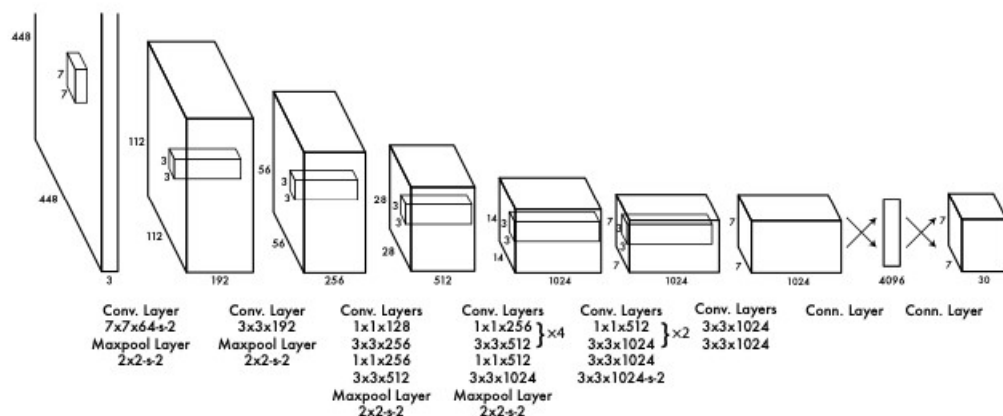
```
ระยะ e1 = 9.49 และระยะ e2 = 15.13
มีการเอียงหน้าเกิดขึ้น ยังไม่รู้ว่าจะเอียงไปทางไหน
รู้แล้ว.. คนในภาพหันหน้าไปทางซ้ายนั่นเอง
```

ตัวอย่างที่ 11.15 การตรวจสอบระยะจุดสองจุดจะใช้คำสั่ง distance ภายในโมดูล scipy ตัวแปร e1 เป็นระยะห่างระหว่างตาซ้ายถึงกึ่งกลางระหว่างคิ้ว ตัวแปร e2 เป็นระยะห่างตาขวาถึงกึ่งกลางระหว่างคิ้ว เมื่อ points[27] คือตำแหน่งกึ่งกลางระหว่างคิ้ว มีค่าเป็น 27 เนื่องจากในภาพเริ่มต้นนับที่จุดที่ 1 แต่ในภาษาไพธอนจะเริ่มตัวเลขลำดับที่ 0 ดังนั้น จึงลดค่าจากจุด 28 เป็นลำดับที่ 27 ตำแหน่งที่ points[39] คือ มุมของขอบตาซ้ายด้านในและ points[42] คือมุมของขอบตาขวาด้านใน

ค่า threshold กำหนดให้เท่ากับ 2 ถ้ามีค่ามากกว่าค่านี้จะถือว่ามีการหันหน้าเอียงออกจากตรงกลาง ตรวจสอบระยะห่างระหว่าง $e1$ และ $e2$ ด้วยคำสั่ง $\text{abs}(e1-e2)$ ผลลัพธ์จะเป็นตัวเลขบวก โดยตรวจสอบหากมีค่ามากกว่า threshold จะถือว่ามีการหันหน้าเอียงไปข้างใดข้างหนึ่ง แต่ถ้ามีค่าน้อยกว่า threshold จะถือว่าเป็นภาพหน้าตรง ตรวจสอบตัวแปร $e1$ และ $e2$ ถ้าระยะ $e1$ น้อยกว่า ระยะ $e2$ คือหันหน้าไปทางซ้าย ในทางกลับกัน ถ้าระยะ $e1$ มากกว่าระยะ $e2$ คือ หันหน้าไปทางขวา

11.17 การตรวจจับวัตถุด้วย Yolo3

YOLO หรือ Only Look Once เป็นอัลกอริทึมตรวจจับวัตถุที่เร็วและใช้วิธีการคอนโวลูชันนอลนิวรอนเน็ตเวิร์ค (Convolutional Neural Network:CNN)



ภาพประกอบที่ 11.18 แสดงสถาปัตยกรรมการตรวจจับวัตถุของ YOLO

ที่มา: <https://iq.opengenus.org/>

ปี ค.ศ. 2016 ได้มีการพัฒนา Yolo สำหรับตรวจจับวัตถุและได้รับการพัฒนาด้านความเร็วให้ดีขึ้น เป็นเวอร์ชัน 3 เรียก YOLOv3 โดยความสามารถในการตรวจจับวัตถุของ YOLO v3 ได้แก่ 'คน', 'จักรยาน', 'รถยนต์', 'รถจักรยานยนต์', 'เครื่องบิน', 'รถบัส', 'รถไฟ', 'รถบรรทุก', 'เรือ', 'ไฟจราจร', 'ดับเพลิง', 'ป้ายหยุด', 'ที่จอดรถ', 'ม้านั่ง', 'นก', 'แมว', 'สุนัข', 'ม้า', 'แกะ', 'วัว', 'ช้าง', 'หมี', 'ม้าลาย', 'ยีราฟ', 'เป้', 'ร่ม', 'กระเป๋าถือ', 'เนคไท', 'กระเป๋าเดินทาง', 'จานร้อน', 'สกี', 'สโนว์บอร์ด', 'ลูกกอล์ฟ', 'วาว', 'ไม้เบสบอล', 'ถุงมือเบสบอล', 'สเก็ตบอร์ด', 'กระดานโต้คลื่น', 'ไม้เทนนิส', 'ขวด', 'แก้วไวน์', 'ถ้วย', 'ส้อม', 'มีด', 'ช้อน', 'ชาม', 'กล้วย', 'แอปเปิ้ล', 'แซนวิช', 'ส้ม', 'บร็อคโคลี่', 'แครอท', 'ฮอตดอก', 'พิซซ่า', 'โดนัท', 'เค้ก', 'แกอี่', 'โซฟา', 'กระถางต้นไม้', 'เตียง', 'โต๊ะรับประทานอาหาร', 'ห้องน้ำ', 'ทีวี', 'แล็ปท็อป', 'เมาส์', 'รีโมท', 'คีย์บอร์ด', 'โทรศัพท์มือถือ', 'ไมโครเวฟ', 'เตาอบ', 'เครื่องปั่นขนมปัง', 'อ่างล้างจาน', 'ตู้เย็น', 'หนังสือ', 'นาฬิกา', 'แจกัน', 'กรรไกร', 'ตุ๊กตาหมี', 'เครื่องเป่าผม', 'แปรงสีฟัน']

YOLO v3 ทำการตรวจจับด้วยการขยาย 3ระดับเพื่อรองรับวัตถุที่แตกต่างกันในความกว้าง 32, 16 และ 8 พิกเซล หากป้อนภาพขนาด 416x416 พิกเซล มันจะร่างเอาวัตถุที่แตกต่างกัน 3 แบบ คือ 13x13x255, 26x26x255 และ 52x52x255 เป็นต้น

ตัวอย่างที่ 11.15 การตรวจจับวัตถุด้วย yolo v3

```
!wget "http://dsdi.msu.ac.th/programming/yolo3/yolov3.weights"
```

```

!wget "http://dsdi.msu.ac.th/programming/yolo3/yolov3.cfg"
!wget "http://dsdi.msu.ac.th/programming/yolo3/yolo3_2564.py"
!wget "http://dsdi.msu.ac.th/programming/yolo3/horse.jpg"

from yolo3_2564 import *
img, target, score = yolo3("horse.jpg")
plt.imshow(img)
plt.show()

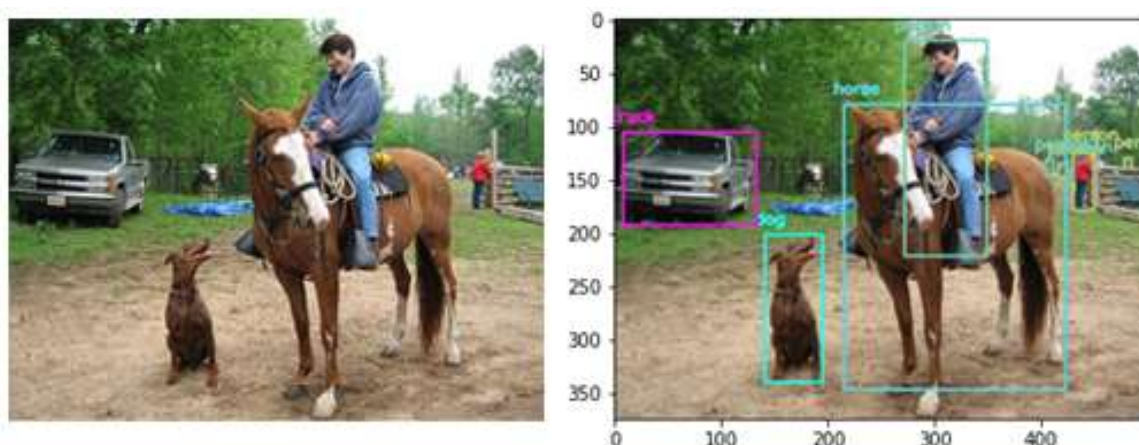
```

ผลลัพธ์

```

horse 0.9938613176345825
person 0.9908708930015564
person 0.9784489870071411
dog 0.9306532144546509
truck 0.8417966365814209

```



ภาพประกอบที่ 11.19 แสดงผลลัพธ์การตรวจจับวัตถุด้วย Yolo v3

ตัวอย่างที่ 11.12 คำสั่ง wget ทำการโหลดไฟล์จากเว็บมาเก็บลงในเครื่อง โดยดาวน์โหลดไฟล์ yolov3.weights yolov3.cfg yolo3_2564.py และ horse.jpg จากนั้นอิมพอร์ต yolo3_2564 ด้วยคำสั่ง from yolo3_2564 import * และเรียกใช้คำสั่ง yolo3() และป้อนไฟล์ภาพที่ต้องการตรวจจับวัตถุ โดยผลลัพธ์จะเก็บไว้ในตัวแปร img, target และ score จากนั้นทำการแสดงผลด้วยคำสั่ง imshow() ผลลัพธ์ที่ได้ตรวจจับพบ ม้า คนสองคน สุนัข และรถยนต์ บอกเป็นค่าความน่าจะเป็นดังผลลัพธ์ด้านบน

แบบฝึกหัดท้ายบท

1. จงอธิบายการเขียนโปรแกรมวาดจุดเส้นตรงและวงกลมด้วย OpenCV
2. จงอธิบายการดำเนินการเกี่ยวกับบิต การรวมสี แยกสี ปรับแกมมาและกลับภาพด้วย OpenCV
3. จงอธิบายการทำงานของคอนโวลูชันและเคอร์เนล
4. จงอธิบายการเขียนโปรแกรมตรวจจับใบหน้าด้วย Haar Cascade และ Tinyface
5. จงอธิบายการเขียนโปรแกรมตรวจจับวัตถุด้วย Yolo3
6. จงเขียนโปรแกรมเพื่อนับจำนวนใบหน้าภายในภาพด้วย tinyface

บทที่ 12

การเขียนโปรแกรมภาษาไพธอนสำหรับ Machine Learning เบื้องต้น

แนวคิด

การเขียนโปรแกรมแบบเดิมที่ไม่ใช้วิธีการทาง Machine Learning จะต้องกำหนดกฎเกณฑ์และเงื่อนไขอย่างชัดเจน ในขณะที่ Machine Learning จะสร้างเงื่อนไขต่าง ๆ เก็บไว้ในโมเดลที่เครื่องจักรเรียนรู้และสร้างเป็นสมการคณิตศาสตร์เอาไว้ภายในโมเดลทำให้การพัฒนาโปรแกรมมีความสะดวกและมีประสิทธิภาพ นอกจากนั้นมีการประยุกต์ใช้ Machine Learning ในวงการต่าง ๆ อย่างกว้างขวาง

ในบทนี้จะนำเสนอเนื้อหาเกี่ยวกับการเขียนโปรแกรมแสดงผลตัวเลขเจ็ดส่วน Seven Segment แบบกำหนดเงื่อนไขและแบบใช้งาน Machine Learning และกล่าวถึง Simple Linear Regression การพล็อตข้อมูล การเขียนโปรแกรมคำนวณสมการเส้นตรงวิธี Simple Linear Regression การนำสมการ Simple Linear Regression ไปใช้พยากรณ์ข้อมูล การเขียนโปรแกรม Metric สำหรับวัดประสิทธิภาพของโมเดล เช่น RMSE และ R2 หลักการ Machine Learning นิวรอนและค่าน้ำหนัก Activation Functions Optimizer การจำแนกประเภทด้วย Decision Tree การจำแนกประเภทและพยากรณ์คำตอบด้วย Support Vector Machine (SVM) การจำแนกประเภทและพยากรณ์คำตอบด้วย Neural Network การจัดกลุ่มข้อมูลด้วย K-Means การเขียนโปรแกรมพยากรณ์และจัดหมวดหมู่ข้อมูล การเขียนโปรแกรมจัดหมวดหมู่ภาพภูเขาและทะเล

วัตถุประสงค์

1. อธิบายความแตกต่างระหว่างการเขียนโปรแกรมใช้/ไม่ใช้ Machine Learning ได้
2. อธิบายการสร้างและใช้งานสมการ Simple Linear Regression ได้
3. อธิบายการสร้างและใช้งานโครงข่ายประสาทเทียมแก้ปัญหา And Gate ได้
4. อธิบายการสร้างและใช้งาน Logistic Regression พยากรณ์ภาพภูเขาและทะเลได้

12.1 ความหมายและประเภทของ Machine Learning

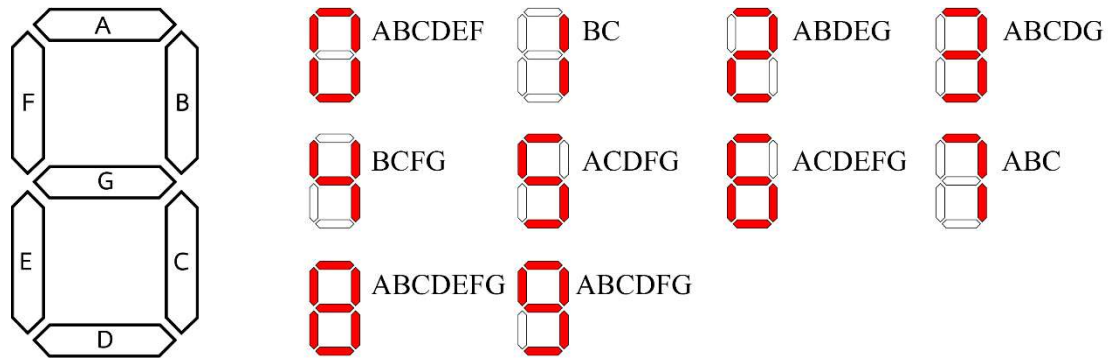
การเรียนรู้ของเครื่องจักร (Machine Learning) คือ การนำข้อมูลส่งให้เครื่องจักรเรียนรู้คำตอบซึ่งแบ่งออกเป็น 3 แบบ

- 1) Supervised Learning คือ การเรียนรู้แบบมีคำตอบ วิธีการนี้เป็นการป้อนข้อมูลและคำตอบเพื่อให้เครื่องจักรค้นหาวิธีการแก้ปัญหาให้ได้มาซึ่งคำตอบ
- 2) Unsupervised Learning คือ การเรียนรู้แบบไม่มีคำตอบ วิธีการนี้จะป้อนข้อมูลโดยไม่มีคำตอบ ดังนั้น เครื่องจักรจะพยายามจัดกลุ่มข้อมูลที่มีความคล้ายคลึงกันและแยกเป็นกลุ่ม
- 3) Reinforcement Learning คือ วิธีการเรียนรู้แบบให้รางวัล (reward) หรือลงโทษ (punishment) แนวคิดนี้สามารถสอนเครื่องจักรให้เอาชนะแชมป์การเล่นเกมโกะระดับสากลได้

12.2 การเขียนโปรแกรมเพื่อตรวจสอบเงื่อนไขและแสดงตัวเลขเจ็ดส่วน (7-Segment)

ถ้ามีตัวแปร 7 ตัว คือ A B C D E F และ G การจะทราบว่าเลขผลลัพธ์มีค่าเท่าใดจะต้องตรวจสอบค่าภายในตัวแปร A-G และใช้คำสั่งเพื่อตรวจสอบเงื่อนไข if / else ดังต่อไปนี้

ตาราง



ภาพประกอบที่ 12.1 แสดงส่วนประกอบของเลข 7-Segment

ตารางที่ 12.1 ส่วนประกอบของเลข 7-Segment

A	B	C	D	E	F	G	เลข
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

ตัวอย่างที่ 12.1 การเขียนโปรแกรมตรวจสอบเงื่อนไข

```

A = 1; B=1; C=1; D=1; E=0; F=1; G=1
if ((A==1) & (B==1) & (C==1) & (D==1) & (E==0) & (F==1) & (G==1)): z = 9
elif ((A==1) & (B==1) & (C==1) & (D==1) & (E==1) & (F==1) & (G==1)): z = 8
elif ((A==1) & (B==1) & (C==1) & (D==0) & (E==0) & (F==0) & (G==0)): z = 7
elif ((A==1) & (B==0) & (C==1) & (D==1) & (E==1) & (F==1) & (G==1)): z = 6
elif ((A==1) & (B==0) & (C==1) & (D==1) & (E==0) & (F==1) & (G==1)): z = 5
elif ((A==0) & (B==1) & (C==1) & (D==0) & (E==0) & (F==1) & (G==1)): z = 4
elif ((A==1) & (B==1) & (C==1) & (D==1) & (E==0) & (F==0) & (G==1)): z = 3

```

```
elif ((A==1) & (B==1) & (C==0) & (D==1) & (E==1) & (F==0) & (G==1)): z = 2
elif ((A==0) & (B==1) & (C==1) & (D==0) & (E==0) & (F==0) & (G==0)): z = 1
elif ((A==1) & (B==1) & (C==1) & (D==1) & (E==1) & (F==1) & (G==0)): z = 0
print("A=%d B=%d C=%d D=%d E=%d F=%d G=%d คือเลข %d"%(A,B,C,D,E,F,G,z))
```

ผลลัพธ์ :

```
A=1 B=1 C=1 D=1 E=0 F=1 G=1 คือเลข 9
```

ตัวอย่างที่ 12.1 การตรวจสอบเงื่อนไขจะใช้ & (And Operator) เพื่อตรวจสอบเงื่อนไขตัวแปร A-G ทีละตัว เช่น ถ้าตัวแปร A-G มีค่าเท่ากับ 1 ทั้งหมด หมายถึงเลข 8 วิธีการเขียนโปรแกรมแบบนี้ต้องตรวจสอบเงื่อนไขอย่างละเอียดถี่ถ้วน ในขณะที่ตัวอย่างถัดไปจะแสดงการเขียนโปรแกรมโดยใช้ Machine Learning เข้ามาช่วยแก้ปัญหา

12.3 การเขียนโปรแกรมตรวจสอบเลข 7-Segment ด้วยการเรียนรู้ของเครื่องจักร

ในหัวข้อนี้แก้ปัญหการตรวจสอบเลข 7-Segment ด้วยวิธีการเรียนรู้ของเครื่องจักร (Machine Learning) โดยใช้โมเดลการแก้ปัญหาด้วยวิธีโครงข่ายประสาทเทียม ซึ่งเป็นหนึ่งในอัลกอริทึมด้าน Machine Learning โดยจะกล่าวถึงรายละเอียดของนิเวศน์ตเวอ์คในหัวข้อถัดไป ในตัวอย่างนี้จะแสดงให้เห็นการนำแนวคิดของการเรียนรู้ของเครื่องจักรเพื่อนำมาแก้ปัญหการตรวจสอบเลข 7-Segment แสดงได้ดังนี้

ตัวอย่างที่ 12.2 การเขียนโปรแกรมด้วย Machine Learning

```
import numpy as np
from sklearn.neural_network import MLPClassifier
x = np.array([
    [1,1,1,1,1,0], [0,1,1,0,0,0], [1,1,0,1,1,0], [1,1,1,1,0,0], [0,1,1,0,0,1],
    [1,0,1,1,0,1], [1,0,1,1,1,1], [1,1,1,0,0,1], [1,1,1,1,1,1], [1,1,1,1,0,1]
])
y = np.array([0,1,2,3,4,5,6,7,8,9])
model = MLPClassifier(max_iter=1000)
model.fit(x, y)

A = 1; B=1; C=1; D=1; E=0; F=1; G=1
test = [A, B, C, D, E, F, G]
z = model.predict([test])[0]
print("A=%d B=%d C=%d D=%d E=%d F=%d G=%d คือเลข %d"%(A,B,C,D,E,F,G,z))
```

ผลลัพธ์ :

A=1 B=1 C=1 D=1 E=0 F=1 G=1 คือเลข 9

ตัวอย่างที่ 12.2 คำสั่ง numpy เป็นไลบรารีสำหรับการคำนวณทางคณิตศาสตร์ และ MLPClassifier เป็นคลาสที่ใช้สร้างโมเดลและทำนายผลจากโมเดลที่สร้างขึ้นด้วยวิธี Neural Network ภายในไลบรารี sklearn ตัวแปร x จะเก็บข้อมูล input ตัวแปร y เก็บข้อมูลผลลัพธ์ จากนั้นสร้างโมเดลด้วยวิธี Neural Network โดยกำหนด iterat คือการวนรอบการสอน 1000 รอบ จากนั้นเริ่ม train ด้วยคำสั่ง model.fit(x,y) ซึ่งเป็นการเรียนรู้จากข้อมูลเข้าคือ x และข้อมูลผลลัพธ์ คือ y โมเดลจะเรียนรู้จนกระทั่งได้โมเดลที่สามารถนำไปใช้เป็นตัวแทนของข้อมูล 7-segment ชุดนี้ได้ คำสั่ง predict() ใช้ทำนายผลลัพธ์ เช่น กำหนดให้ตัวแปร A-G มีค่าเท่ากับ 1 ทั้งหมดยกเว้น F=0 ซึ่งเมื่อส่งเข้าไปทำนายผลโมเดลจะตอบว่าเป็นเลข 9 ซึ่งทำงานได้อย่างถูกต้อง วิธีการทาง Machine Learning เป็นวิธีที่ง่ายและสะดวกมากเมื่อเทียบกับการเขียนโปรแกรมแบบเก่าที่ต้องกำหนดเงื่อนไขต่าง ๆ ยุ่งยากและวุ่นวายในขณะการเรียนรู้ของเครื่องจักรแก้ปัญหาเดียวกันได้อย่างสะดวกรวดเร็วและมีประสิทธิภาพ

ตัวอย่างที่ 12.3 การแสดงคำตอบของ seven segment จากโมเดลที่สร้างขึ้น

```
for i in x:
    z = model.predict([i])[0]
    print("%s = %d"%(str(i),z))
```

ผลลัพธ์ :

```
[1 1 1 1 1 1 0] = 0
[0 1 1 0 0 0 0] = 1
[1 1 0 1 1 0 1] = 2
[1 1 1 1 0 0 1] = 3
[0 1 1 0 0 1 1] = 4
[1 0 1 1 0 1 1] = 5
[1 0 1 1 1 1 1] = 6
[1 1 1 0 0 1 0] = 7
[1 1 1 1 1 1 1] = 8
[1 1 1 1 0 1 1] = 9
```

ตัวอย่างที่ 12.3 การวนเข้าไปในตัวแปร x ซึ่งมีค่าตัวแปร A-G แต่ละแบบที่แตกต่างกันไปและแทนด้วยตัวเลข 7-Segment ของเลขแต่ละตัว พบว่าโมเดลที่สร้างขึ้นจาก Machine Learning ให้ผลลัพธ์ที่ถูกต้องทุก ๆ คำตอบ ของเลข 7-Segment

12.4 การบันทึกและโหลดโมเดลกลับมาใช้งาน

โมเดลที่สร้างขึ้นมาแล้วนั้นสามารถบันทึกและนำกลับมาใช้ทำนายผลได้โดยไม่ต้องมาสร้างโมเดลขึ้นใหม่ โดยใช้คำสั่ง pickle เพื่อเก็บโมเดลนั้นลงในไฟล์ และเรียกกลับมาใช้ทำนายได้อีกเท่าที่ต้องการแสดงการบันทึกโมเดล ดังนี้

ตัวอย่างที่ 12.4 การเขียนโปรแกรมตรวจสอบเงื่อนไข

```
import pickle
s = pickle.dumps(model)
open("7segment-model.bin","wb").write(s)
```

ตัวอย่างที่ 12.4 เป็นการใช้งานโมดูล pickle สำหรับเก็บตัวแปรในรูปของสตริงและบันทึกลงในระบบไฟล์ โดยคำสั่ง dump() ทำหน้าที่แปลงตัวแปร model ให้เป็นสตริงและบันทึกลงในไฟล์ ตั้งชื่อว่า 7segment-model.bin

ตัวอย่างที่ 12.5 การโหลดโมเดลกลับมาใช้งาน

```
import pickle
model7 = pickle.loads(open("7segment-model.bin","rb").read())
A = 1; B=1; C=1; D=1; E=0; F=1; G=1
test = [A, B, C, D, E, F, G]
z = model7.predict([test])[0]
print("A=%d B=%d C=%d D=%d E=%d F=%d G=%d คือเลข %d"%(A,B,C,D,E,F,G,z))
```

ผลลัพธ์

```
A=1 B=1 C=1 D=1 E=0 F=1 G=1 คือเลข 9
```

ตัวอย่างที่ 12.5 โมเดลที่สร้างขึ้นเมื่อบันทึกเก็บไว้ใช้งานสามารถนำมาใช้ได้หลังจากการโหลดโมเดลขึ้นมาแล้วนำมาใช้ทำนายค่าตอบได้ โดยใช้ไลบรารี pickle และคำสั่ง load() เมื่ออาร์กิวเมนต์ของ load() คือชื่อไฟล์ที่เก็บโมเดลที่เทรนเอาไว้โดยโหลดด้วยโหมด rb (read-binary) ในตัวอย่างเป็นการทำนายว่า A-G ที่มีค่าเป็น 1 ทั้งหมดยกเว้น F=0 จะมีค่าเท่าใด ซึ่งคำตอบที่ได้จากโมเดลคือ เลข 9

12.5 การสร้างโมเดลด้วยวิธี Simple Linear Regression

สูตรของ Simple Linear Regression ดังนี้

$$\hat{Y} = \beta_0 + (\beta_1 \times x)$$

$$\beta_1 = \frac{\sum x \times y - \frac{\sum x \sum y}{n}}{\sum x^2 - n(\bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1(\bar{x})$$

พิจารณาน้ำหนักและเส้นรอบอกหมีจำนวน 10 ตัว



ภาพประกอบที่ 12.2 น้ำหนักหมีและเส้นรอบอกหมีจำนวน 10 ตัว

ตารางที่ 12.2 น้ำหนักหมีและเส้นรอบอกหมีจำนวน 10 ตัว

ลำดับ	น้ำหนักหมี (x)	เส้นรอบอกหมี (y)
1	220	113
2	210	132
3	450	150
4	440	153
5	520	162
6	710	170
7	908	215
8	1,200	250
9	1,300	320
10	1,350	343

คำนวณค่าและเขียนลงในตาราง ดังต่อไปนี้

เมื่อ

ตัวแปร x แทนน้ำหนักหมี

ตัวแปร y แทนเส้นรอบอกหมี

ตารางที่ 12.3 การคำนวณค่าจากตัวแปร x และ y

ลำดับ	x (น้ำหนักหมี)	y (รอบอกหมี)	x^2	$x \times y$	y^2
1	220	113	$220^2 = 48,400$	24,860	$113^2 = 12,769$
2	210	132	$210^2 = 44,100$	27,720	$132^2 = 17,424$
3	450	150	$450^2 = 202,500$	67,500	$150^2 = 22,500$
4	440	153	$440^2 = 193,600$	67,320	$153^2 = 23,409$
5	520	162	$520^2 = 270,400$	84,240	$162^2 = 26,244$
6	710	170	$710^2 = 504,100$	120,700	$170^2 = 28,900$
7	908	215	$908^2 = 824,464$	195,220	$215^2 = 46,225$
8	1200	250	$1200^2 = 1,440,000$	300,000	$250^2 = 62,500$
9	1300	320	$1300^2 = 1,690,000$	416,000	$320^2 = 102,400$

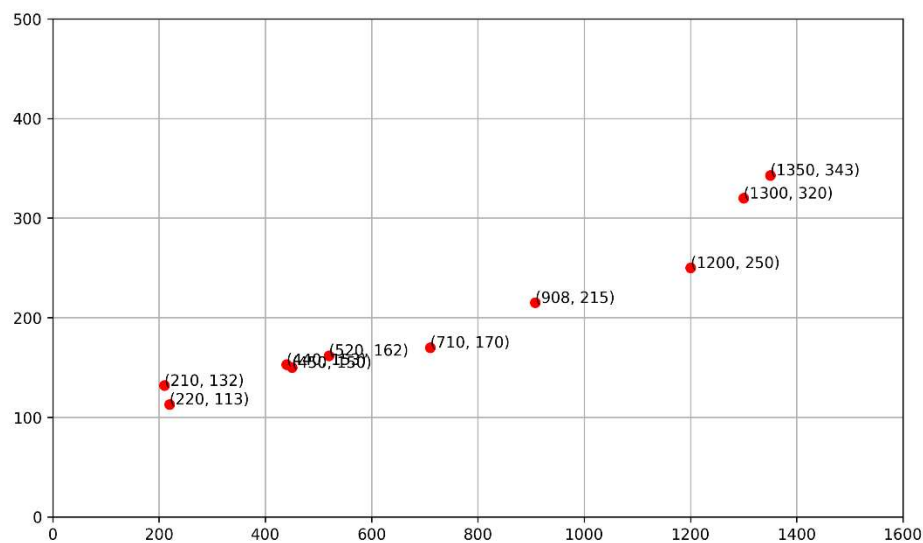
ลำดับ	x (น้ำหนักหมี)	y (รอบอกหมี)	x^2	$x \times y$	y^2
10	1350	343	$1350^2 = 1,822,500$	463,050	$343^2 = 117,649$
รวม	7,308	2,008	7,040,064	1,766,610	460,020

ตัวอย่างที่ 12.6 การพล็อตข้อมูลน้ำหนักเทียบกับเส้นรอบอกหมี

```
%pylab inline
x = [220, 210, 450, 440, 520, 710, 908, 1200, 1300, 1350]
y = [113, 132, 150, 153, 162, 170, 215, 250, 320, 343]

figure(figsize=(10, 6), dpi=80)
plt.grid(1)
plt.plot(x, y, 'ro')
plt.axis([0,1600, 0, 500])
for xi, yi in zip(x, y):
    plt.text(xi, yi, '({}, {})'.format(xi, yi))
plt.show()
```

ผลลัพธ์



ภาพประกอบที่ 12.3 แสดงผลลัพธ์การพล็อตน้ำหนักและความยาวรอบอกหมี

ตัวอย่างที่ 12.6 กำหนดให้ตัวแปร `fig` มีความละเอียด 80 dpi และให้วาดเส้นกริด ด้วยคำสั่ง `grid(1)` และพล็อต `ro` คือวงกลมสีแดง `r = red`, `o = วงกลม` ให้แกน มีความกว้าง 0 - 1600 และสูง 0 - 500 วนเข้าไปใน `zip(x,y)` โดยเขียนข้อความ ในรูปแบบ `"({}, {})".format(xi,yi)` มีค่าเหมือนการใช้คำสั่งควบคุมรูปแบบการแสดงผล `"(%d,%d)"` จากนั้นเรียกคำสั่ง `show()` เพื่อแสดงกราฟ

ตัวอย่างที่ 12.7 การเขียนโปรแกรมสร้างสมการ Simple Linear Regression

```
x = [220, 210, 450, 440, 520, 710, 908, 1200, 1300, 1350]
y = [113, 132, 150, 153, 162, 170, 215, 250, 320, 343]
n = len(x)
xmean=sum(x)/len(x)
ymean=sum(y)/len(y)
x2 = list(map(lambda a:a**2, x))
y2 = list(map(lambda a:a**2, y))
xy = list(map(lambda a,b:a*b, x,y))
B1 = (sum(xy) - ((sum(x)*sum(y)) / n)) / (sum(x2) - (n * (xmean**2)))
B0 = ymean - (B1*xmean)
model = lambda xi: B0 + (B1 * xi)
print("สมการ Simple Linear Regression คือ : Y = %.2f + (%.2f * x)"%(B0,B1))
```

ผลลัพธ์ :

```
สมการ Simple Linear Regression คือ : Y = 72.15 + (0.18 * x)
```

ตัวอย่างที่ 12.8 การทำนายเส้นรอบอกหมีจากสมการรีเกรสชันที่สร้างขึ้น

```
model = lambda xi: B0 + (B1 * xi)
z = model(230)
print("ทำนายว่าหมีหนัก %d มีความยาวรอบอกเท่ากับ %.2f หน่วย"%(230,z))
```

ผลลัพธ์ :

```
ทำนายว่าหมีหนัก 230 มีความยาวรอบอกเท่ากับ 112.64 หน่วย
```

ตัวอย่างที่ 12.7 และ 12.8 เป็นการสร้างสมการเส้นตรงจากข้อมูล x และ y โดยคำนวณหาค่า b0 และ b1 เมื่อ lambda a:a**2 หมายถึง การยกกำลังสองของข้อมูล คำสั่ง lambda a,b:a*b หมายถึง รับอาร์กิวเมนต์ 2 ตัวคือ a และ b จากนั้นนำมาคูณกัน เมื่อทดลองป้อนน้ำหนักหมี 230 กิโลกรัม จะได้เส้นรอบอกมีค่า 112.64 หน่วย

12.6 การสร้างสมการ Simple Linear Regression จากการทดสอบแรงดึงของกาว

การศึกษาเรื่องแรงดึงของกาวในการยึดชิ้นงานเข้าด้วยกัน ขั้นตอนการดำเนินการ คือ ทากาวลงบนชิ้นงานทั้งสองชิ้นจากนั้นนำมาติดเข้าด้วยกัน จากนั้นนำไปอบเพื่อให้กาวแห้งติดกัน

ผู้ศึกษาต้องการทราบความสัมพันธ์ระหว่าง อุณหภูมิที่ใช้อบ กับแรงดึงของกาว โดยทดลอง 3 ตัวอย่าง รอบที่ 1 ใช้ตัวอย่าง 3 ชิ้น โดยให้ความร้อน 70°C เวลาอบ 15 นาที นำไปวางในอุณหภูมิห้อง 20 นาที นำไปทดสอบแรงดึงจนชิ้นงานฉีกขาดออกจากกัน ทนแรงได้ 2.3 2.6 และ 2.1 ตามลำดับ รอบที่ 2 ใช้ตัวอย่าง 3 ชิ้น โดยให้ความร้อน 80°C เวลาอบ 15 นาที นำไปวางในอุณหภูมิห้อง 20 นาที นำไปทดสอบแรงดึงจนชิ้นงานฉีกขาดออกจากกัน ทนแรงได้ 2.5 2.9 และ 2.4 ตามลำดับ รอบที่ 3 ใช้ตัวอย่าง

3 ชั้น โดยให้ความร้อน 90°C เวลาอบ 15 นาที นำไปวางในอุณหภูมิห้อง 20 นาที นำไปทดสอบแรงดึงจนชิ้นงานฉีกขาดออกจากกัน ทนแรงได้ 3.0 3.1 และ 2.8 ตามลำดับ รอบที่ 4 ใช้ตัวอย่าง 3 ชั้น โดยให้ความร้อน 100°C เวลาอบ 15 นาที นำไปวางในอุณหภูมิห้อง 20 นาที นำไปทดสอบแรงดึงจนชิ้นงานฉีกขาดออกจากกัน ทนแรงได้ 3.3 3.5 และ 3.0 ตามลำดับ

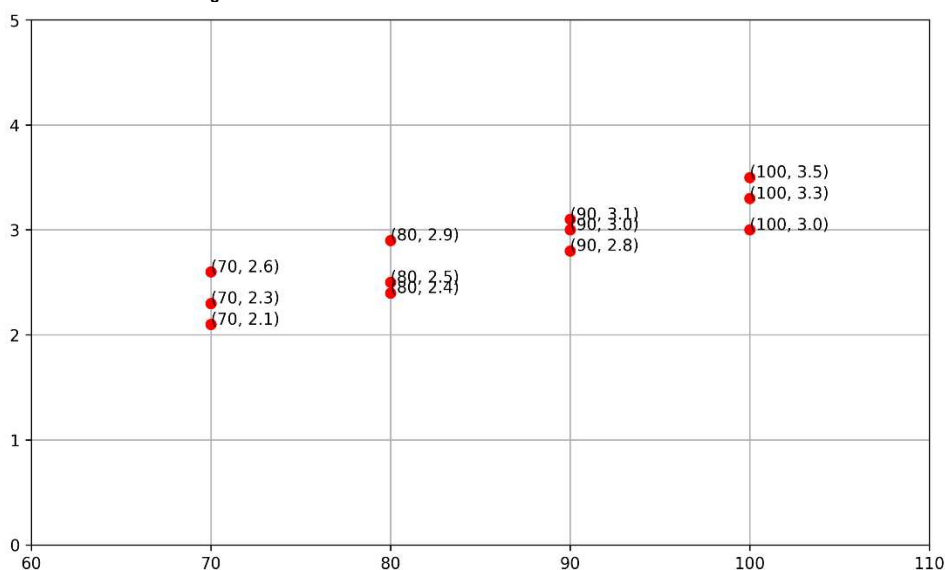
ตารางที่ 12.4 การทดสอบวัดแรงดึงของชิ้นงานที่อุณหภูมิที่แตกต่างกัน

70°C	80°C	90°C	100°C
2.3	2.5	3.0	3.3
2.6	2.9	3.1	3.5
2.1	2.4	2.8	3.0

ตัวอย่างที่ 12.9 การพล็อตกราฟข้อมูล

```
%pylab inline
x = [70,70,70,80,80,80,90,90,90,100,100,100]
y = [2.3,2.6,2.1,2.5,2.9,2.4,3.0,3.1,2.8,3.3,3.5,3.0]
figure(figsize=(10, 6), dpi=80); plt.grid(1); plt.plot(x, y, 'ro')
plt.axis([60,110, 0, 5])
for xi, yi in zip(x, y):
    plt.text(xi, yi, '{}, {}'.format(xi, yi))
plt.show()
```

ผลลัพธ์การพล็อตกราฟข้อมูล



ภาพประกอบที่ 12.4 แสดงผลลัพธ์อุณหภูมิและการทนแรงดึงของชิ้นงาน

ตัวอย่างที่ 12.9 กำหนดให้ตัวแปร `finger()` มีความละเอียด 80 dpi และให้วาดเส้นกริด ด้วยคำสั่ง `grid(1)` และพล็อต `ro` คือวงกลมสีแดง `r = red`, `o =` วงกลม ให้แกน มีความกว้าง 60 - 110 และสูง 0 - 5 วนเข้าไปใน `zip(x,y)` โดยเขียนข้อความ ในรูปแบบ `"({}, {})".format(xi,yi)` มีค่าเหมือนการใช้คำสั่งควบคุมรูปแบบการแสดงผล `"(%d,%d)"` จากนั้นเรียกคำสั่ง `show()` เพื่อแสดงกราฟ

ตัวอย่างที่ 12.10 การเขียนโปรแกรมสร้างสมการ Simple Linear Regression

```
x = [70, 70, 70, 80, 80, 80, 90, 90, 90, 100, 100, 100]
y = [2.3, 2.6, 2.1, 2.5, 2.9, 2.4, 3.0, 3.1, 2.8, 3.3, 3.5, 3.0]
n = len(x)
xmean=sum(x)/len(x)
ymean=sum(y)/len(y)
x2 = list(map(lambda a:a**2, x))
y2 = list(map(lambda a:a**2, y))
xy = list(map(lambda a,b:a*b, x,y))
B1 = (sum(xy) - ((sum(x)*sum(y)) / n)) / (sum(x2) - (n * (xmean**2)))
B0 = ymean - (B1*xmean)
model = lambda xi: B0 + (B1 * xi)
print("สมการ Simple Linear Regression คือ : Y = %.2f + (%.2f * x)"%(B0,B1))
```

ผลลัพธ์ :

สมการ Simple Linear Regression คือ : $Y = 0.10 + (0.03 * x)$

ตัวอย่างที่ 12.11 การทำนายแรงดึงของชิ้นงานจากสมการรีเกรสชันที่สร้างขึ้น

```
for i in range(70,101,2):
    predict = B0 + (B1 * i)
    print("อบชิ้นงานที่อุณหภูมิ %d°C คาดว่าทนแรงดึงได้ %.2f หน่วย"%(i, predict))
```

ผลลัพธ์ :

อบชิ้นงานที่อุณหภูมิ 70°C คาดว่าทนแรงดึงได้ 2.32 หน่วย
 อบชิ้นงานที่อุณหภูมิ 72°C คาดว่าทนแรงดึงได้ 2.38 หน่วย
 อบชิ้นงานที่อุณหภูมิ 74°C คาดว่าทนแรงดึงได้ 2.44 หน่วย
 อบชิ้นงานที่อุณหภูมิ 76°C คาดว่าทนแรงดึงได้ 2.51 หน่วย
 อบชิ้นงานที่อุณหภูมิ 78°C คาดว่าทนแรงดึงได้ 2.57 หน่วย
 อบชิ้นงานที่อุณหภูมิ 80°C คาดว่าทนแรงดึงได้ 2.63 หน่วย
 อบชิ้นงานที่อุณหภูมิ 82°C คาดว่าทนแรงดึงได้ 2.70 หน่วย
 อบชิ้นงานที่อุณหภูมิ 84°C คาดว่าทนแรงดึงได้ 2.76 หน่วย
 อบชิ้นงานที่อุณหภูมิ 86°C คาดว่าทนแรงดึงได้ 2.82 หน่วย

อบชื้นงานที่อุณหภูมิ 88°C คาดว่าทนแรงดึงได้ 2.89 หน่วย
 อบชื้นงานที่อุณหภูมิ 90°C คาดว่าทนแรงดึงได้ 2.95 หน่วย
 อบชื้นงานที่อุณหภูมิ 92°C คาดว่าทนแรงดึงได้ 3.01 หน่วย
 อบชื้นงานที่อุณหภูมิ 94°C คาดว่าทนแรงดึงได้ 3.08 หน่วย
 อบชื้นงานที่อุณหภูมิ 96°C คาดว่าทนแรงดึงได้ 3.14 หน่วย
 อบชื้นงานที่อุณหภูมิ 98°C คาดว่าทนแรงดึงได้ 3.20 หน่วย
 อบชื้นงานที่อุณหภูมิ 100°C คาดว่าทนแรงดึงได้ 3.27 หน่วย

ตัวอย่างที่ 12.10 และ 12.11 เป็นการสร้างสมการเส้นตรงจากข้อมูล x และ y โดยคำนวณหาค่า b_0 และ b_1 เมื่อ $\lambda = a^2$ หมายถึง การยกกำลังสองของข้อมูล คำสั่ง $\lambda = a \cdot b$ หมายถึง รับอาร์กิวเมนต์ 2 ตัวคือ a และ b จากนั้นนำมาคูณกัน เมื่อทดลองป้อนอุณหภูมิจาก 70°C ถึง 100°C จะได้ค่าทนแรงดึงตามผลลัพธ์ตัวอย่างที่ 12.11

12.7 Metric สำหรับวัดประสิทธิภาพของโมเดล

การวัดความถูกต้องของโมเดลมีหลายตัว แบ่งเป็น 2 กลุ่มใหญ่ 1) ปัญหาด้าน Regression ได้แก่ MAE, MAD, MSE, MAPE, RMSE และ R2 และ 2) ปัญหาด้าน Classification ได้แก่ Accuracy, Precision, Recall, F1-Score, F-Beta Score และ AUC

Machine Learning เรียก การคำนวณค่าผิดพลาดจากโมเดลที่สร้างขึ้นว่า Error function หรือ "Loss function" โดยเป้าหมายของการ train คือการลดค่า loss ให้มีค่าต่ำที่สุด

12.8 การคำนวณค่าความผิดพลาดของโมเดลด้วย RMSE (Root Mean Square Error)

การคำนวณความผิดพลาดของโมเดลด้วยวิธี RMSE คำนวณจากสูตรต่อไปนี้

$$RMSE = \sqrt{\frac{1}{n} \times \sum (prediction - actual)^2}$$

เมื่อ prediction คือ ค่าที่คำนวณได้จากโมเดลหรือสมการที่สร้างขึ้นจากกลุ่มข้อมูล
 actual คือ ค่าจริงที่ได้จากการเก็บรวบรวมข้อมูล

RMSE ย่อจาก "Root Mean Square Error" เป็นการนำ MSE มาถอด Square Root เพื่อให้ได้ค่า loss มีหน่วยเดียวกับคำตอบ ค่า RMSE จะแปลผลง่าย เช่น ถ้ามี RMSE เท่ากับ 2.5 แปลว่าโดยเฉลี่ยโมเดลทำนายผิดไป +/- 2.5 หน่วย

ตัวอย่างที่ 12.12 การเขียนโปรแกรมคำนวณ RMSE (Root Mean Square Error)

```

from math import sqrt
x = [70,70,70,80,80,80,90,90,90,100,100,100]
y = [2.3,2.6,2.1,2.5,2.9,2.4,3.0,3.1,2.8,3.3,3.5,3.0]
n = len(x)
xmean=sum(x)/n ; ymean=sum(y)/n
  
```

```

x2 = list(map(lambda a:a**2, x))
y2 = list(map(lambda a:a**2, y))
xy = list(map(lambda a,b:a*b, x,y))
B1 = (sum(xy) - ((sum(x)*sum(y)) / n)) / (sum(x2) - (n * (xmean**2)))
B0 = ymean - (B1*xmean)
print("สมการ Regression คือ : Y = %.2f + (%.2f * x)"%(B0,B1))
error= []
for i in range(len(x)):
    forecast = B0 + (B1 * x[i])
    actual = y[i]
    E = sqrt(1/n * (actual-forecast)**2)
    error.append(E)
    print("อุณหภูมิ %d°C ทนแรงฉีกได้ %.2f หน่วย คำนวนจากสมการได้ = %.2f หน่วย \
มีความผิดพลาด = %.2f "%(x[i], y[i], forecast, E))
RMSE = sum(error)
print("ค่า RMSE ของโมเดลที่สร้างขึ้นมีค่า = %.2f หน่วย"%RMSE)

```

ผลลัพธ์ :

```

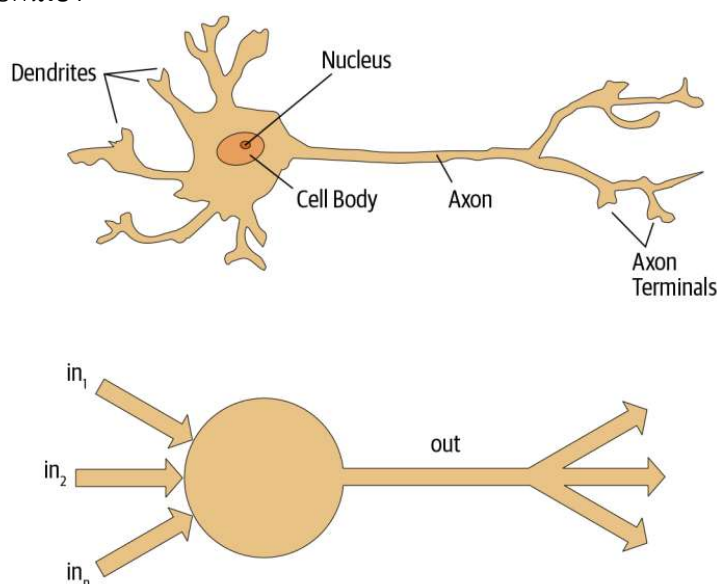
สมการ Regression คือ : Y = 0.10 + (0.03 * x)
อุณหภูมิ 70°C ทนแรงฉีกได้ 2.30 หน่วย คำนวนจากสมการได้ = 2.32 หน่วย มีความผิดพลาด = 0.00
อุณหภูมิ 70°C ทนแรงฉีกได้ 2.60 หน่วย คำนวนจากสมการได้ = 2.32 หน่วย มีความผิดพลาด = 0.08
อุณหภูมิ 70°C ทนแรงฉีกได้ 2.10 หน่วย คำนวนจากสมการได้ = 2.32 หน่วย มีความผิดพลาด = 0.06
อุณหภูมิ 80°C ทนแรงฉีกได้ 2.50 หน่วย คำนวนจากสมการได้ = 2.63 หน่วย มีความผิดพลาด = 0.04
อุณหภูมิ 80°C ทนแรงฉีกได้ 2.90 หน่วย คำนวนจากสมการได้ = 2.63 หน่วย มีความผิดพลาด = 0.08
อุณหภูมิ 80°C ทนแรงฉีกได้ 2.40 หน่วย คำนวนจากสมการได้ = 2.63 หน่วย มีความผิดพลาด = 0.07
อุณหภูมิ 90°C ทนแรงฉีกได้ 3.00 หน่วย คำนวนจากสมการได้ = 2.95 หน่วย มีความผิดพลาด = 0.01
อุณหภูมิ 90°C ทนแรงฉีกได้ 3.10 หน่วย คำนวนจากสมการได้ = 2.95 หน่วย มีความผิดพลาด = 0.04
อุณหภูมิ 90°C ทนแรงฉีกได้ 2.80 หน่วย คำนวนจากสมการได้ = 2.95 หน่วย มีความผิดพลาด = 0.04
อุณหภูมิ 100°C ทนแรงฉีกได้ 3.30 หน่วย คำนวนจากสมการได้ = 3.27 หน่วย มีความผิดพลาด = 0.01
อุณหภูมิ 100°C ทนแรงฉีกได้ 3.50 หน่วย คำนวนจากสมการได้ = 3.27 หน่วย มีความผิดพลาด = 0.07
อุณหภูมิ 100°C ทนแรงฉีกได้ 3.00 หน่วย คำนวนจากสมการได้ = 3.27 หน่วย มีความผิดพลาด = 0.08
ค่า RMSE ของโมเดลที่สร้างขึ้นมีค่า = 0.59 หน่วย

```

ตัวอย่างที่ 12.12 เป็นการคำนวณหาค่า B0 และ B1 เมื่อ $\lambda a:a^2$ คือ ฟังก์ชันยกกำลังสอง $\lambda a,b:a*b$ คือฟังก์ชันคูณค่าตัวเลขสองค่า หลังจากได้สมการถดถอยแล้วให้คำนวณค่า Error ของค่า x และค่าตอบ y เมื่อ $E = \sqrt{1/n * (actual - forecast)^2}$ ค่า actual คือค่าคำตอบจริงซึ่งอยู่ในตัวแปร y ส่วน forecast เป็นคำตอบที่ได้จากการคำนวณจากสมการเส้นตรงที่สร้างขึ้น ค่าความผิดพลาดจะเก็บไว้ในตัวแปร error ให้คำนวณผลรวมด้วยคำสั่ง `sum(error)` จะเป็นค่า RMSE

12.9 โครงข่ายประสาทเทียม (Artificial Neural Network)

ในปี ค.ศ. 1943 Warren McCulloch นักประสาทวิทยาและ Walter Pitts นักตรรกวิทยาได้ร่วมมือกันพัฒนาแบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทเทียมในบทความเรื่อง "A Logical Calculus of the Ideas Immanent in Nervous Activity" แบบจำลองอย่างง่ายของเซลล์ประสาทสามารถเรียนรู้ได้ด้วยตนเอง



ภาพประกอบที่ 12.5 แสดงเซลล์ประสาทและเซลล์ประสาทเทียม

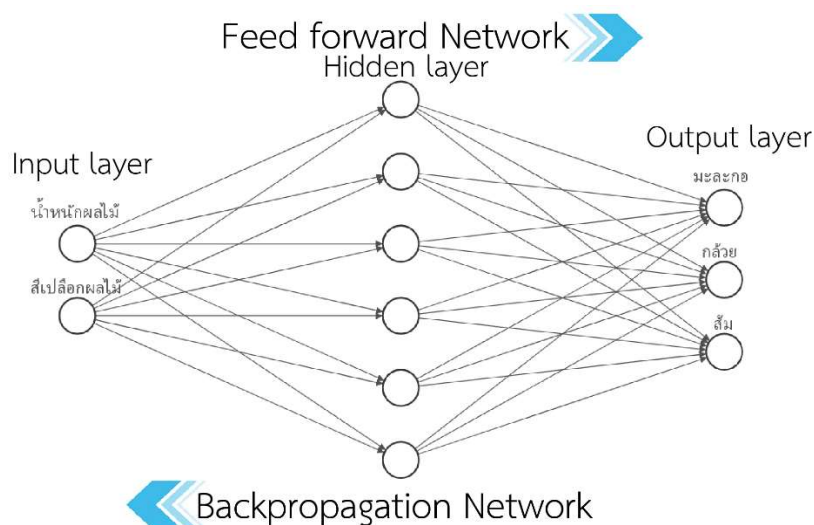
ที่มา: <https://www.researchgate.net>

ศาสตราจารย์ Marvin Minsky จาก MIT เขียนหนังสือ "Perceptrons (MIT news)" กล่าวว่า perceptron ชั้นเดียวไม่สามารถเรียนรู้ฟังก์ชันคณิตศาสตร์ที่เรียบง่าย เช่น Exclusive OR และกล่าวว่า การใช้โครงข่ายประสาทเทียมหลายชั้นจะแก้ปัญหาดังกล่าวได้ ส่งผลให้นักวิชาการระดับโลกแทบทั้งสิ้น เลิกใช้โครงข่ายประสาทเทียม ต่อมาแบบจำลองส่วนใหญ่สร้างขึ้นด้วยเซลล์ประสาทสองชั้นและถูกใช้งานอย่างแพร่หลาย

โครงข่ายประสาทเทียม (Artificial Neural Network) เป็นแบบจำลองที่ได้รับความนิยมอย่างมากสามารถเลียนแบบการทำงานของสมองมนุษย์ แบ่งเป็น 2 ประเภท คือ 1) การเรียนรู้แบบมีคำตอบ (Supervised Learning) โดยโครงข่ายจะปรับตัวเองเพื่อให้ได้คำตอบที่ดีขึ้นจนใกล้เคียงกับคำตอบจริงมากที่สุด เช่น Backpropagation Learning เป็นต้น 2) การเรียนรู้แบบไม่มีคำตอบ (Unsupervised Learning) โครงข่ายจะสร้างโครงข่ายเพื่อจัดกลุ่มข้อมูลที่มีลักษณะใกล้เคียงกันออกเป็นกลุ่มได้เอง เช่น Self Organizing Map เป็นต้น

สถาปัตยกรรมของโครงข่ายประสาทเทียมแบ่งเป็น 2 ประเภท 1) โครงข่ายประสาทเทียมแบบตรงไปข้างหน้า (Feedforward Network) เป็นการส่งข้อมูลจากอินพุตไปยังชั้นซ่อนและส่งออกไปยังเอาพุต โดยข้อมูลจะไหลไปข้างหน้าในทิศทางเดียวจนถึงเอาพุต 2) โครงข่ายประสาทเทียมแบบย้อนกลับ

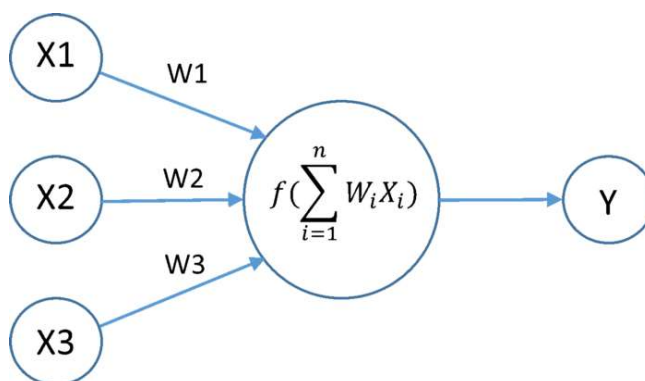
(Backpropagation Network) มีการนำข้อมูลส่งกลับเข้าไปในเน็ตเวิร์คเพื่อปรับปรุงค่าความผิดพลาดของคำตอบ (Error)



ภาพประกอบที่ 12.6 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบไม่ย้อนกลับและแบบแพร่ย้อนกลับ

การแบ่งประเภทตามระดับชั้นของโครงข่ายประสาทเป็น 2 ประเภท

1) โครงข่ายประสาทเทียมชั้นเดียว (Single layer Perceptron) ไม่มีชั้นซ่อน (Hidden layer) ประกอบด้วยชั้นอินพุต (input layer) และผ่านฟังก์ชันกระตุ้นและส่งออกไปยังชั้นเอาพุต (Output layer)



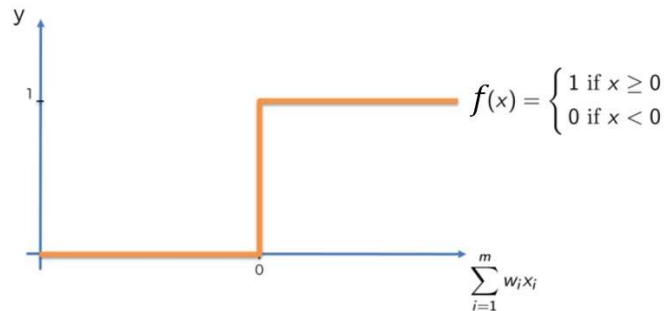
ภาพประกอบที่ 12.7 แสดงโครงสร้างประสาทเทียมแบบชั้นเดียว

2. โครงข่ายประสาทเทียมแบบหลายชั้น (Multi-layer Perceptron) ประกอบด้วย ชั้นอินพุต (Input layer) ชั้นซ่อน (Hidden layer) ก่อนส่งข้อมูลไปยังชั้นเอาพุตจะผ่านฟังก์ชันกระตุ้น (Activate function) โดยในชั้นซ่อนสามารถมีได้หลายชั้น

การเลือกใช้ฟังก์ชันกระตุ้น (Activate function) ขึ้นอยู่กับลักษณะข้อมูลของคลาส เช่น ถ้าผลลัพธ์เป็นค่าไม่ต่อเนื่อง สามารถใช้ Threshold function จากสูตร

$$f(x) = \begin{cases} 1 & \text{if } x \geq T \\ 0 & \text{if } x < T \end{cases}$$

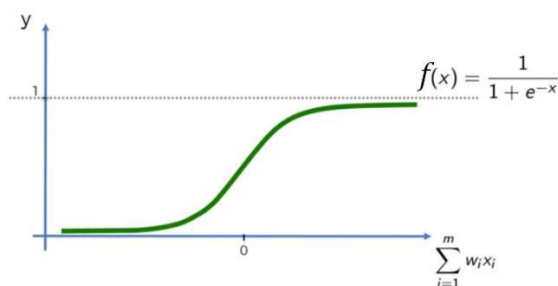
เมื่อ T คือค่าคงที่ เช่น 0 เป็นต้น



ภาพประกอบที่ 12.8 กราฟตัวอย่างของฟังก์ชัน Threshold

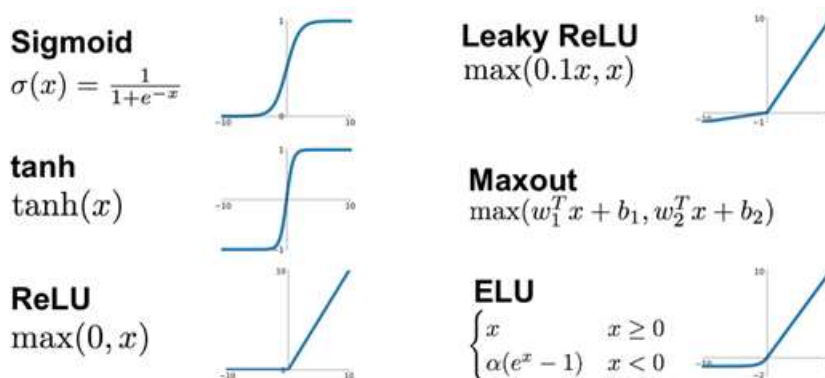
ถ้าผลลัพธ์เป็นตัวเลขต่อเนื่องสามารถใช้ continuous function เช่น Sigmoid function ตามสูตรต่อไปนี้

$$f(x) = \frac{1}{1 + e^{-x}}$$



ภาพประกอบที่ 12.9 กราฟตัวอย่างของฟังก์ชัน Sigmoid

ฟังก์ชันกระตุ้น (Activation Function) จะเป็นตัวกำหนดว่าน้ำหนักของนิวรอนมีผลกับคำตอบของมากน้อยเพียงใด ถ้าฟังก์ชันกระตุ้นให้ผลลัพธ์ค่าน้ำหนักมีค่า 0 หมายถึงน้ำหนักอันนั้นไม่มีผลต่อคำตอบ เป็นต้น ฟังก์ชันกระตุ้นมีอยู่หลายฟังก์ชันเลือกใช้ให้เหมาะกับข้อมูล ดังฟังก์ชันกระตุ้นประเภทต่าง ๆ ดังนี้



ภาพประกอบที่ 12.10 แสดงฟังก์ชันกระตุ้นแบบต่าง ๆ

ที่มา: https://en.wikipedia.org/wiki/Activation_function

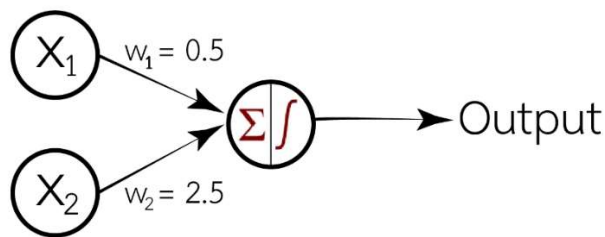
12.10 การคำนวณ Single Layer Perceptron ของ And Gate

ในตัวอย่างนี้เป็นการแก้ปัญหาลอจิก And Gate ด้วย Neural Network ซึ่งเป็น Machine Learning Algorithm โดยมีตารางความจริงดังต่อไปนี้

ตารางที่ 12.5 ตารางความจริงของ And Gate

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

จากตารางด้านบนสามารถสร้างสถาปัตยกรรมโครงข่ายประสาทเทียมแบบชั้นเดียว โดยสุ่มค่าเริ่มต้นให้น้ำหนัก w_1 มีค่า 0.5 และ w_2 เท่ากับ 2.5 ดังนี้



ภาพประกอบที่ 12.11 สถาปัตยกรรมโครงข่ายประสาทเทียมของ And Gate

กำหนดให้ค่า Bias (θ) มีค่าเท่ากับ 1.0 และอัตราการเรียนรู้ (α) กำหนดให้เท่ากับ 0.2 ขั้นตอนการสร้างโครงข่ายประสาทเทียมแบบชั้นเดียว ดังนี้

1. ป้อนค่าอินพุตเข้าโครงข่าย
2. คำนวณคำตอบ $\hat{y} = f(w_1x_1 + w_2x_2 - \theta)$

โดยที่

$$\hat{y} = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 - \theta \geq 0 \\ 0 & \text{if } w_1x_1 + w_2x_2 - \theta < 0 \end{cases}$$

3. คำนวณหาค่าความผิดพลาด

$$\epsilon = y - \hat{y}$$

4. ปรับค่า Weight และ Bias ใหม่

$$\begin{aligned} w_{new} &= w_{old} + \Delta w \\ \theta_{new} &= \theta_{old} + \Delta \theta \end{aligned}$$

โดยที่

$$\begin{aligned} \Delta w &= \alpha (y - \hat{y})x_i \\ \Delta \theta &= -\alpha (y - \hat{y}) \end{aligned}$$


```

for i in range(4):
    for j in range(len(X1)):
        counter+=1
        x1 = X1[j] ; x2 = X2[j] ; y = Y[j]
        y2 = F(x1, x2, y, w1, w2,theta)
        E = y - y2
        dw1 = alpha*(y-y2)*x1
        dw2 = alpha*(y-y2)*x2
        dtheta = -alpha * (y-y2)
        theta+=dtheta
        w1 += dw1
        w2 += dw2
        print("%d\t%d\t%d\t%d\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f"%\
              (counter,x1,x2,y,y2,E, w1,dw1,w2,dw2,theta,dtheta))

```

ผลลัพธ์

n	x1	x2	y	y'	E	w1	dw1	w2	dw2	theta	dtheta
1)	0	0	0	0	0	0.50	0.00	2.50	0.00	1.00	-0.00
2)	0	1	0	1	-1	0.50	-0.00	2.30	-0.20	1.20	0.20
3)	1	0	0	0	0	0.50	0.00	2.30	0.00	1.20	-0.00
4)	1	1	1	1	0	0.50	0.00	2.30	0.00	1.20	-0.00
5)	0	0	0	0	0	0.50	0.00	2.30	0.00	1.20	-0.00
6)	0	1	0	1	-1	0.50	-0.00	2.10	-0.20	1.40	0.20
7)	1	0	0	0	0	0.50	0.00	2.10	0.00	1.40	-0.00
8)	1	1	1	1	0	0.50	0.00	2.10	0.00	1.40	-0.00
9)	0	0	0	0	0	0.50	0.00	2.10	0.00	1.40	-0.00
10)	0	1	0	1	-1	0.50	-0.00	1.90	-0.20	1.60	0.20
11)	1	0	0	0	0	0.50	0.00	1.90	0.00	1.60	-0.00
12)	1	1	1	1	0	0.50	0.00	1.90	0.00	1.60	-0.00
13)	0	0	0	0	0	0.50	0.00	1.90	0.00	1.60	-0.00
14)	0	1	0	1	-1	0.50	-0.00	1.70	-0.20	1.80	0.20
15)	1	0	0	0	0	0.50	0.00	1.70	0.00	1.80	-0.00
16)	1	1	1	1	0	0.50	0.00	1.70	0.00	1.80	-0.00

12.11 การสร้างโมเดลจากโครงข่ายประสาทเทียมที่สร้างขึ้น

จากตาราง 12.6 พบว่าค่าน้ำหนัก w_1 และ w_2 ที่ไม่มีการเปลี่ยนแปลงในรอบที่ 16 เป็นต้นไป ดังนั้น สูตรที่ได้จากการเทรนโมเดลนิวรอนเน็ตเวิร์คแบบ backpropagation มีค่า $w_1 = 0.5$ และ $w_2 = 1.7$ และค่า $\theta = 1.80$ นำมาสร้างเป็นสมการเพื่อแก้ปัญหา And Gate ได้ดังนี้

ตัวอย่างที่ 12.14 การสร้างโมเดลจากโครงข่ายประสาทเทียมที่สร้างขึ้น

```
def predict(x1,x2):
    w1 = 0.5
    w2 = 1.7
    theta = 1.80
    z = x1*w1 + x2*w2 - theta
    if z >=0: return 1
    else: return 0

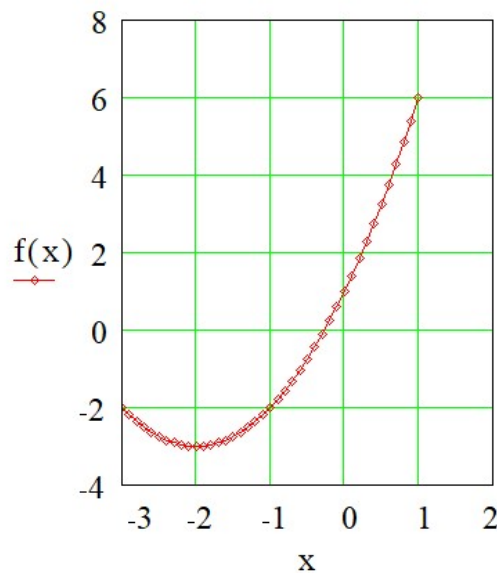
print(predict(0,0))
print(predict(0,1))
print(predict(1,0))
print(predict(1,1))
```

ตัวอย่างที่ 12.14 การสอนโมเดลเพื่อต้องการค่าน้ำหนักที่มีค่าความผิดพลาดน้อยที่สุดแล้วนำค่าน้ำหนักเหล่านั้นสร้างเป็นสูตรเพื่อทำนายผลลัพธ์ของข้อมูล โดยในตัวอย่างนี้คือการแก้ปัญหาของ AND Gate พบว่าภายหลังการ train model ได้ผลลัพธ์ที่รอบที่ 16 ค่าน้ำหนักไม่มีการเปลี่ยนแปลง ตัวแปร $w1 = 0.5$ ตัวแปร $w2 = 1.7$ และตัวแปร $\theta = 1.80$ นำมาสร้างสูตรการคำนวณ คือ $x1*w1 + x2*w2 - \theta$ เมื่อผลลัพธ์มีค่ามากกว่าหรือเท่ากับ 0 ให้คืนค่าเป็น 1 หากไม่ใช่ให้คืนค่าเป็น 0 เขียนสูตรลงในฟังก์ชัน `predict(x1,x2)` โดยรับอาร์กิวเมนต์ 2 ตัว คือ $x1$ และ $x2$ จากนั้นทำการเรียกฟังก์ชัน `predict()` โดยป้อนอาร์กิวเมนต์ให้ค่า (0,0) (0,1) (1,0) และ (1,1) พบว่าผลลัพธ์มีค่า 0, 0, 0 และ 1 แก้ปัญหา And Gate ได้ถูกต้อง 100%

12.11 Gradient Descent

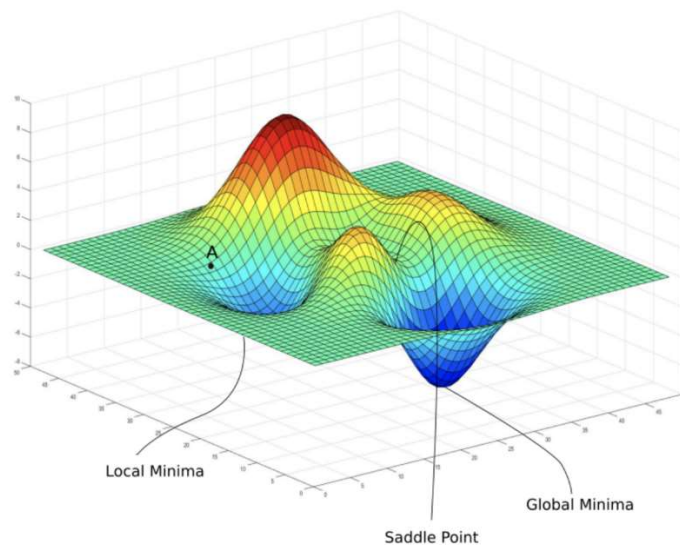
Gradient Descent คือ กระบวนการค้นหาค่าของตัวแปร weight และ bias ที่ทำให้ฟังก์ชัน $J(\theta)$ หรือ cost function มีค่าต่ำสุดโดยไม่ใช้วิธีการแก้สมการหลายตัวแปร แต่ใช้การเคลื่อนที่ตามแนวความชันของสมการซึ่งทำงานได้รวดเร็วและมีประสิทธิภาพ

พิจารณาสมการพาราโบลา $f(x) = (x + 2)^2 + 3$ ต้องการหาค่าตัวแปร x ที่ทำให้ค่า y ต่ำสุด ซึ่งจะใช้แคลคูลัส คือ $\frac{dy}{dx} f(x) = 0$ เมื่อ $\frac{dy}{dx}$ คือ ความชัน ณ จุดใด ๆ เมื่อความชัน = 0 คือเป็นเส้นตรงขนานกับแกน x วิธีการหาจุดต่ำสุดของสมการ คือการหาอนุพันธ์ของฟังก์ชัน ในตัวอย่างนี้มีค่าเท่ากับ $2x + 4$ แก้สมการหาค่า $x = -2$ ดังนั้น ค่า x ที่ -2 จะทำให้ค่า y ต่ำสุด



ภาพประกอบที่ 12.12 แสดงการหาค่าต่ำสุดของ loss function ด้วยการหาอนุพันธ์

ในนิเวศน์เน็ตเวิร์คต้องใช้การแก้สมการหลายตัวแปร ในปี ค.ศ. 1944 ได้มีการนำเสนอวิธีแก้สมการหลายตัวแปรด้วยวิธี Gradient Descent โดยใช้วิธีการเคลื่อนที่ตามพื้นผิวลงสู่จุดต่ำสุด ดังภาพต่อไปนี้



ภาพประกอบที่ 12.13 แสดงการหาค่าต่ำสุดของสมการหลายตัวแปรด้วยวิธี Gradient Descent
ที่มา: <https://www.researchgate.net>

วิธีการทำงานของ Gradient Descent คือ การสุ่มค่าตำแหน่งใด ๆ บนพื้นผิวจากนั้น คำนวณความชันรอบจุดนั้น เลือกทิศทางที่มีความชันมากที่สุด และก้าวไปยังตำแหน่งใหม่ จากนั้นทำซ้ำ ๆ จนสุดท้ายจะมาถึงจุดต่ำสุดของสมการ ซึ่งคือจุดที่ทำให้โมเดลมีความผิดพลาดน้อยที่สุด โดยหลุมที่ตื้นเรียกว่า Local minima และหลุมที่อยู่ลึกที่สุดเรียกว่า global minima

สมการของ Gradient Descent คือ

$$\theta_{new} = \theta - \eta \times \nabla_{\theta} J(\theta)$$

θ_{new} คือ θ ในลำดับถัดไป

θ คือ ตำแหน่งปัจจุบัน

η คือ Learning Rate

∇ คือ partial differential ของ cost function

J คือ cost function

12.12 การแยกภาพภูเขาและทะเลด้วย Logistic Regression

โมเดลโลจิสติกเรกเรสชัน (logistic regression) ใช้สำหรับวิเคราะห์ข้อมูลที่มีคำตอบ 2 คลาส ในตัวอย่างนี้จะทำการแยกระหว่างภาพภูเขาและภาพทะเลด้วยโมเดลโลจิสติกเรกเรสชัน

mahotas เป็นไลบรารีการประมวลผลภาพ (Image Processing) ใช้สกัดคุณลักษณะเด่นภายในภาพ ติดตั้งด้วยคำสั่ง !pip install mahotas หรือจากเว็บ <https://pypi.org/project/mahotas>

Mahotas ประกอบด้วยอัลกอริทึมจำนวนมากสำหรับเพิ่มความเร็วในการดำเนินการใน numpy array มีฟังก์ชันมากกว่า 100 ฟังก์ชันสำหรับประมวลผลภาพ ประกอบด้วยคุณสมบัติที่สำคัญได้แก่ 1) การสกัดฟีเจอร์ global และ local features รวมถึง SURF และ Haralick features 2) Convolve ความสามารถในการ convolution และ wavelets การ decomposition และ local feature computation 3) Morph ภาพได้แก่ erode() , dilate() 4) Watershed เป็นการอิมพลีเมนต์ watershed 5) imread/imsave สำหรับอ่านเขียนภาพ



ภาพประกอบที่ 12.14 แสดงภาพภูเขาและทะเลเพื่อใช้ train model



ภาพประกอบที่ 12.15 แสดงภาพภูเขาและทะเลเพื่อใช้ทำนายผลจากโมเดลที่สร้างขึ้น

ตัวอย่างที่ 12.15 การสร้างโมเดลเพื่อพยากรณ์ภาพภูเขาหรือทะเล

```
%pylab inline
import mahotas as mh
features = []
labels = []

img = mh.imread("sea1.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("sea2.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("sea3.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("sea4.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("sea5.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("mountain1.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())
```

```

img = mh.imread("mountain2.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())

img = mh.imread("mountain3.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())
img = mh.imread("mountain4.jpg")
img = mh.colors.rgb2gray(img, dtype=np.uint8)
features.append(mh.features.haralick(img).ravel())
labels = ['sea','sea','sea','sea','sea','mountain','mountain','mountain','mountain']
features = np.array(features)
labels = np.array(labels)

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
model = Pipeline([('preproc', StandardScaler()),('classifier', LogisticRegression())])
model.fit(features,labels)

```

ตัวอย่างที่ 12.15 คำสั่ง %pylab inline เป็นการขอใช้คำสั่ง numpy และ matplotlib ในการสกัดคุณลักษณะเด่นของภาพใช้ไลบรารี mahotas โดยอิมพอร์ตและตั้งชื่อว่า mh ด้วยคำสั่ง import mahotas as mh จากนั้น ให้ตัวแปร features และ labels เป็นลิสต์ว่าง คำสั่ง imread() จะอ่านภาพที่ละไฟล์ โดยแปลงให้อยู่ในรูป rgb2gray() คือแปลงจากสีเป็นภาพเกรย์สเกล จากนั้น ใช้คำสั่ง me.features.haralick() เพื่อสกัดคุณลักษณะเด่นของภาพด้วยวิธี haralick() แล้วแปลงให้เป็นอาร์เรย์เพียงแถวเดียวด้วยคำสั่ง ravel() จากนั้นนำคุณลักษณะเด่นที่สกัดได้เก็บไว้ในตัวแปร features ด้วยคำสั่ง append() ทำซ้ำครบจำนวนภาพที่จะใช้สอนโมเดล ได้แก่ sea1.jpg ถึง sea5.jpg และ mountain1.jpg ถึง mountain4.jpg

ตัวแปร labels เป็นคำตอบ จะต้องเรียงให้ถูกต้องตามลำดับภาพที่ใส่ลงในตัวแปร features ถ้าป้อนลำดับผิดคือสอนคอมพิวเตอร์ด้วยคำตอบที่ผิด จะส่งผลต่อการพยากรณ์ หลังจากนั้นแปลงตัวแปร features และ labels ให้เป็นชนิด numpy array ด้วยคำสั่ง np.array(ตัวแปร)

การสร้างโมเดล Logistic Regression ใช้คำสั่ง Pipeline([('preproc', StandardScaler()), ('classifier', LogisticRegression())]) คือการกำหนดให้ตัว classifier เป็นคลาสของ LogisticRegression จากนั้นสร้างโมเดลด้วยคำสั่ง model.fit(features,labels) เมื่อ features คือคุณลักษณะเด่นของภาพมีทั้งหมด 9 ชุด คือภาพทะเล 5 และภาพภูเขา 4 ภาพ ส่วนตัวแปร labels คือคำตอบของภาพ คือ 5 ตัวแปรคือ "sea" และ 4 ตัวหลังคือ "mountain"

ตัวอย่างที่ 12.16 การทำนายภาพภูเขาหรือทะเลจากโมเดลที่สร้างขึ้น

```
test = mh.imread("test1.jpg")
test = mh.colors.rgb2gray(test, dtype=np.uint8)
test_feature = mh.features.haralick(test).ravel()
print(model.predict([test_feature]))

test = mh.imread("test2.jpg")
test = mh.colors.rgb2gray(test, dtype=np.uint8)
test_feature = mh.features.haralick(test).ravel()
print(model.predict([test_feature]))
```

ผลลัพธ์

```
['sea']
['mountain']
```

ตัวอย่างที่ 12.16 หลังจากสร้างโมเดลด้วยคำสั่ง fit() แล้วต่อไปเป็นการทำนายว่าภาพ test1.jpg และ test2.jpg เป็นภาพทะเลหรือภูเขา โดยใช้คำสั่งอ่านภาพและสกัดคุณลักษณะเด่นจากนั้นป้อนเข้าไปในฟังก์ชัน predict() ผลลัพธ์ทำนายว่าภาพ test1.jpg เป็นภาพทะเล และภาพ test2.jpg เป็นภาพภูเขา

แบบฝึกหัดท้ายบท

1. จงอธิบายความแตกต่างระหว่างการเขียนโปรแกรมใช้/ไม่ใช่ Machine Learning
2. จงอธิบายการเขียนโปรแกรมสร้างและพยากรณ์ข้อมูลด้วย Simple Linear Regression
3. จงเขียนโปรแกรมสร้างและใช้งานโครงข่ายประสาทเทียมแก้ปัญหา And Gate
4. จงเขียนโปรแกรมสร้างและใช้งาน Logistic Regression พยากรณ์ภาพภูเขาและทะเล

บรรณานุกรม

- Garreta, R., & Moncecchi, G. (2013). *Learning Scikit-learn : Machine Learning in Python: Experience the Benefits of Machine Learning Techniques by Applying Them to Real-world Problems Using Python and the Open Source Scikit-learn Library*. Birmingham, UK: Packt Publishing.
- GNU Free Documentation License. (2020, October 3). Retrieved from <http://rigaux.org/language-study/diagram.html>
- Gollapudi, S. (2019). *Learn Computer Vision Using OpenCV: With Deep Learning CNNs and RNNs*. Berkeley, CA: Apress.
- Harrington, P. (2012). *Machine Learning in Action*. Manning. Shelter Island, N.Y.: Manning.
- Howard, J. & Gugger, S. (2020). *Deep Learning for Coders with Fastai and PyTorch*. Sebastopol, CA: O'Reilly Media.
- Johansson, R. (2019). *Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib*. 2nd ed. Berkeley, CA: Apress.
- Ketkar, N.& Moolayil, J. (2021). *Deep Learning with Python : Learn Best Practices of Deep Learning Models with PyTorch*. 2nd ed. Berkeley, CA: Apress.
- Kopec, D. (2019). *Classic Computer Science Problems in Python*. [N.p.]: Manning.
- Korites, B. J. (2018). *Python Graphics: A Reference for Creating 2D and 3D Images*. Berkeley, CA: Apress
- Korstanje, J. (2021). *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*. [S.l.] : Apress.
- Krogh, J. W. (2018). *MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers*. Berkeley, CA: Apress.
- Kruk, S. (2018). *Practical Python AI Projects: Mathematical Models of Optimization Problems with Google OR-Tools*. Berkeley, CA: Apress.
- Lancaster, A., and Webster, G. (2019). *Python for the Life Sciences: A Gentle Introduction to Python for Life Scientists*. [New York]: Apress.
- Li, R. (2020). *Essential Statistics for Non-STEM Data Analysts Get to Grips with the Statistics and Math Knowledge Needed to Enter the World of Data Science with Python*. [N.p.]: Packt Publishing.
- Lukaszewski, A. (2010). *MySQL for Python: Integrate the Flexibility of Python and the Power of MySQL to Boost the Productivity of Your Applications*. Birmingham: Packt Publishing.
- Mathur, P. (2019). *Machine Learning Applications Using Python: Cases Studies from Healthcare, Retail, and Finance*. Berkeley, CA: Apress.
- McKinney, Wes. (2013). *Python for Data Analysis*. Sebastopol, CA: O'Reilly Media.

- Mukhopadhyay, S. (2018). *Advanced Data Analytics Using Python: With Machine Learning, Deep Learning and NLP Examples*. Berkeley, CA: Apress.
- Pajankar, A. (2021). *Practical Python Data Visualization: A Fast Track Approach to Learning Data Visualization with Python*. Berkeley, CA: Apress.
- Raghavendra, S. (2020). *Python Testing with Selenium Learn to Implement Different Testing Techniques Using the Selenium WebDriver*. Berkeley, CA: Apress.
- Research Gate. (2021, January 12). Retrieved from <https://www.researchgate.net>
- Sarkar, D., Bali, R., & Sharma, T. (2018). *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems*. Berkeley, CA: Apress.
- TIOBE Index History. (2020, October 3). Retrieved from <https://tiobe.com>
- Solem, J. E. (2012). *Programming Computer Vision with Python*. Sebastopol, CA: O'Reilly Media.
- Varga, E. (2019). *Practical Data Science with Python 3: Synthesizing Actionable Insights from Data*. New York: Apress.
- Westra, E. (2010). *Python Geospatial Development: Build a Complete and Sophisticated Mapping Application from Scratch Using Python Tools for GIS Development*. Olton, Birmingham: Packt Publishing.
- Wikipedia. (2021, January 3). Retrieved from https://en.wikipedia.org/Activation_function
- Wilkes, M. (2020). *Advanced Python Development: Using Powerful Language Features in Real-World Applications*. [Berkeley, CA] : Apress.
- Ziadé, T. (2008). *Expert Python Programming: Learn Best Practices to Designing, Coding, and Distributing Your Python Software*. Birmingham, UK: Packt Publishing.