

เอกสารประกอบการสอนรายวิชา  
477-201 การเขียนโปรแกรมคอมพิวเตอร์  
การเขียนโปรแกรมภาษา Python เบื้องต้น  
(Basic Python Programming)

ดร. จันทวรรณ ปิยะวัฒน์

สาขาวิชาระบบสารสนเทศทางธุรกิจ  
ภาควิชาบริหารธุรกิจ คณะวิทยาการจัดการ  
มหาวิทยาลัยสงขลานครินทร์  
ภาคการศึกษาที่ 1/2562

---

---

# คำนำ

เอกสารประกอบการสอนเล่มนี้จัดทำขึ้นสำหรับการสอนรายวิชา 477-201 การเขียนโปรแกรมคอมพิวเตอร์ (Computer Programming) ในภาคการศึกษาที่ 1 ปีการศึกษา 2562 ซึ่งเป็นรายวิชาบังคับของนักศึกษาหลักสูตรระบบสารสนเทศทางธุรกิจ ชั้นปีที่ 2 ภาควิชาบริหารธุรกิจ คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ จำนวน 3 หน่วยกิต 3(2-2-5) เป็นการสอนทฤษฎี 2 ชั่วโมงต่อสัปดาห์ ปฏิบัติ 2 ชั่วโมงต่อสัปดาห์ และนักศึกษาควรศึกษาค้นคว้าด้วยตัวเอง 5 ชั่วโมงต่อสัปดาห์

วิชา 477-201 การเขียนโปรแกรมคอมพิวเตอร์ มีจุดมุ่งหมายให้นักศึกษาได้มีพื้นฐานความรู้ความเข้าใจในหลักการเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา Python ส่วนประกอบต่างๆของโปรแกรมคอมพิวเตอร์ และภาษา Python สามารถเขียนโปรแกรมคอมพิวเตอร์พื้นฐานด้วยภาษา Python ได้ตามการวิเคราะห์และออกแบบขั้นตอนการทำงานของโปรแกรมอย่างมีระบบ สามารถเขียนโปรแกรมแบบมีเงื่อนไขเพื่อการตัดสินใจ เขียนคำสั่งเพื่อให้โปรแกรมทำงานวนซ้ำได้ และเข้าใจการใช้งานโมดูลส่วนเสริมต่างๆ ของโปรแกรมภาษา Python เพื่อนำความรู้เหล่านี้ไปใช้ในการเขียนโปรแกรมระดับในระดับที่ยากขึ้นซึ่งได้แก่ การเขียนโปรแกรมแบบฟังก์ชันและการเขียนโปรแกรมเชิงวัตถุได้

หนังสือเล่มนี้ได้จัดแบ่งเนื้อหาออกเป็น 11 บท ในแต่ละบทจะมีแบบฝึกหัดท้ายบทเพื่อให้ผู้เรียนได้ลองวิเคราะห์และออกแบบแนวทางแก้ไขปัญหาและพัฒนาออกมาเป็นโปรแกรมด้วยภาษา Python ที่ได้เรียนรู้ไปแล้วได้ ทั้งนี้ผู้จัดทำหวังเป็นอย่างยิ่งว่าเอกสารประกอบการสอนฉบับนี้จะให้ความรู้และเป็นประโยชน์แก่ผู้เรียนและผู้อ่านทุกๆ ท่าน เพื่อสร้างความรู้ความเข้าใจในการฝึกเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นให้ดียิ่งขึ้น หากมีข้อเสนอแนะประการใด ผู้จัดทำขอรับไว้ด้วยความขอบพระคุณยิ่ง

**ดร.จันทวรรณ ปิยะวัฒน์**

สาขาวิชาระบบสารสนเทศทางธุรกิจ

ภาควิชาบริหารธุรกิจ คณะวิทยาการจัดการ

มหาวิทยาลัยสงขลานครินทร์



# สารบัญ

คำนำ	iii
แผนการสอน	xi
References	xxxviii
0.1 Python คืออะไร . . . . .	1
0.2 Python ทำงานอย่างไร . . . . .	2
0.3 อัลกอริทึม (Algorithm) และ ผังงาน (Flowchart) . . . . .	2
0.4 การติดตั้งโปรแกรม Python Runtime . . . . .	3
บรรณานุกรม . . . . .	9



# สารบัญรูป

1	TIOBE Index for Python ในปี พ.ศ. 2562 . . . . .	1
2	สัญลักษณ์ผังงาน (Flowchart) . . . . .	4
3	ตัวอย่างการเขียนผังงาน . . . . .	5
4	การ Download Python 3.7 . . . . .	6
5	เลือก Add Python 3.7 to PATH . . . . .	6
6	ไฟล์เตอร์ Python 3.7 . . . . .	7
7	ตัวอย่างหน้าโปรแกรม Idle หน้า Editor . . . . .	7
8	ตัวอย่างหน้าโปรแกรม Idle หน้า Python Shell . . . . .	8





# สารบัญตาราง



# แผนการสอน

## คำอธิบายรายวิชาและวัตถุประสงค์ที่ระบุไว้ในหลักสูตร

แนวความคิดเรื่องการเขียนโปรแกรม ขั้นตอนวิธีในการแก้ไขปัญหา การสร้างคำสั่งสำหรับเขียนขั้นตอนวิธีการ เขียนผังงาน นิพจน์ คำสั่งในการเขียนโปรแกรม หลักไวยากรณ์ของภาษาโปรแกรมระดับสูง การเขียนโปรแกรมสมัยใหม่ การทดสอบ การแก้ไขโปรแกรม การติดตั้ง และการเขียนเอกสารประกอบโปรแกรม

Concept of programming, Algorithm to solve the problem, Flowchart, Expression and instruction, High-level language syntax, Modern programming, Testing, Debugging, Installation and software documentation

## วัตถุประสงค์ของวิชา

มีจุดมุ่งหมายให้นักศึกษาได้มีพื้นฐานความรู้ความเข้าใจในหลักการเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา Python ส่วนประกอบต่างๆ ของโปรแกรมคอมพิวเตอร์และภาษา Python สามารถเขียนโปรแกรมคอมพิวเตอร์อย่างง่ายด้วยภาษา Python ได้ตามการวิเคราะห์และออกแบบขั้นตอนการทำงานของโปรแกรมอย่างมีระบบ และมีความรู้ความเข้าใจในการเขียนโปรแกรมแบบมีเงื่อนไขเพื่อการตัดสินใจ การเขียนคำสั่งเพื่อการทำงานซ้ำ และโมดูลส่วนเสริมต่างๆ ของโปรแกรมภาษา Python เพื่อเรียนรู้เรื่องการเขียนโปรแกรมแบบฟังก์ชันและการเขียนโปรแกรมเชิงวัตถุได้

## เนื้อหาวิชา

### สัปดาห์ที่ 1

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

#### เค้าโครงวิชา

- วัตถุประสงค์รายวิชา
- รายละเอียดเนื้อหาวิชา
- การวัดผลและการประเมินผล
- เงื่อนไขและข้อตกลงอื่น
- วิธีการเรียนการสอน
- เว็บไซต์และหนังสืออ่านประกอบ

#### ระบบจัดการการเรียนรู้ (ClassStart.org)

- ระบบในภาพรวม
- การสมัครสมาชิก
- การเข้าห้องเรียนออนไลน์ของรายวิชา
- การใช้งานระบบ
- การเข้าอ่านเอกสารการสอนและคลิป
- การส่งแบบฝึกหัดทางออนไลน์

- การทำข้อสอบออนไลน์
- การตรวจสอบคะแนนเก็บ
- การบันทึกการเรียนรู้ (Reflections)
- การสื่อสารออนไลน์

#### เว็บไซต์ Code.org

- การสมัครสมาชิก
- ฝึกการเขียนโปรแกรมง่ายๆ (Game-based Learning) แบบ Block-based Programming

#### **กิจกรรมการเรียนรู้การสอน/สื่อที่ใช้**

- บรรยาย
- ปฏิบัติการใช้ระบบ ClassStart.org
- ปฏิบัติการเขียนโปรแกรมทางออนไลน์ที่ Code.org

## สัปดาห์ที่ 2

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 1 ความรู้เบื้องต้นเกี่ยวกับ Python

- Python คืออะไร
- Python ทำงานอย่างไร
- อัลกอริทึมและผังงาน
- การติดตั้งโปรแกรม Python Runtime

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ฝึกการเขียนผังงาน
- ปฏิบัติการติดตั้ง Python Runtime
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

### สัปดาห์ที่ 3

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 2 ส่วนประกอบของ Python

- ตัวแปร
- ประเภทของข้อมูล
- การคำนวณ
- Expressions และ Statements
- Comments
- Source Code
- คำสั่ง `print()`
- คำสั่ง `input()`

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

#### สัปดาห์ที่ 4

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 3 ประโยคเงื่อนไข

- Boolean Expressions
- การใช้ `if`, `elif` และ `else`

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้



## สัปดาห์ที่ 5

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 3 ประโยคเงื่อนไข (ต่อ)

- Chained Expressions
- Nested Expressions

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 6

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 4 การเขียนและใช้งานฟังก์ชัน

- การเรียกใช้ฟังก์ชัน
- การเรียกใช้โมดูล
- ฟังก์ชันซ้อน

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 7

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 4 การเขียนและใช้งานฟังก์ชัน (ต่อ)

- การสร้างฟังก์ชัน
- การคืนค่าของฟังก์ชัน
- การเขียนโปรแกรมเชิงฟังก์ชัน
- การเขียนคำอธิบายโปรแกรม

ทบทวนเนื้อหาก่อนสอบกลางภาค

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 8

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 5 การใช้ประโยชน์ลิ่งทำงานวนซ้ำ

- ฟังก์ชัน `range()`
- คำสั่ง `for`

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 9

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 5 การใช้ประโยชน์คำสั่งทำงานวนซ้ำ (ต่อ)

- คำสั่ง `while`
- คำสั่ง `break`
- ฟังก์ชันที่เรียกตัวเอง

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 10

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

### บทที่ 6 การใช้งาน String

- ฟังก์ชัน `len()`
- การเดินทางตามตัวชี้ของ String
- การตัดคำใน String
- โครงสร้างข้อมูลที่ไม่เปลี่ยนแปลงไม่ได้
- การค้นหาตัวอักษรใน String
- String Methods
- การใช้ `in`
- การเปรียบเทียบ String
- การจัดวางรูปแบบของ String

### กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 11

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

### บทที่ 7 ลิสต์ (List)

- การเข้าถึงค่าในลิสต์
- การแบ่งข้อมูลในลิสต์
- การใช้ in กับลิสต์
- การเดินทางในลิสต์
- ตัวเนินการของลิสต์
- List Methods
- Map, reduce, and filter
- Lists กับ String
- Objects กับ values

### กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 12

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 8 ดิกชันนารี (Dictionary)

- การอ่านค่าใน Dictionary
- การหาค่าของ Key ใน Dictionary
- Dictionary and List
- ฟังก์ชันที่รับ Parameters ได้ไม่จำกัด

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้



### สัปดาห์ที่ 13

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 9 ทูเปิล (Tuple)

- ความหมายของ Tuple
- การสลับค่าของ Tuple
- การเก็บค่าการดำเนินการใน Tuple
- ฟังก์ชัน `list()`
- Dictionary และ Tuple

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 14

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

บทที่ 10 การจัดการไฟล์ (Files)

- การทำงานกับ Directories
- การเปิดไฟล์
- การอ่านไฟล์
- การจัดการข้อผิดพลาด
- ฐานข้อมูลแบบ Key-Value
- การเรียกใช้โปรแกรมอื่น

กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม
- บันทึกการเรียนรู้

## สัปดาห์ที่ 15

ผู้สอน จันทวรรณ ปิยะวัฒน์

จำนวนชั่วโมงบรรยาย 2

จำนวนชั่วโมงปฏิบัติ 2

หัวข้อ/รายละเอียด

### บทที่ 11 Object-Oriented Programming

- คลาสและออบเจกต์
- การสร้างคลาส
- การสร้างออบเจกต์
- ฟังก์ชัน `__init__()`
- การสร้างเมธอดของออบเจกต์
- การแก้ไขค่าแอตทริบิวต์ของออบเจกต์
- การลบแอตทริบิวต์ของออบเจกต์
- การลบออบเจกต์
- การสืบทอดคลาส

### ทบทวนเนื้อหาก่อนสอบปลายภาค

### กิจกรรมการเรียนการสอน/สื่อที่ใช้

- ทดสอบทบทวนความรู้
- บรรยายและยกตัวอย่างการเขียนโปรแกรม
- ถามตอบในชั้นเรียน
- ปฏิบัติการเขียนโปรแกรม

- บันทึกการเรียนรู้

## การจัดการประสบการณ์การเรียนรู้

- บรรยายและถ่ายทอดประสบการณ์แก่ผู้เรียน
- ถามตอบในชั้นเรียน
- ฝึกปฏิบัติการเขียนโปรแกรม
- ฝึกนำเสนอผลงานการเขียนโปรแกรมที่พัฒนาด้วยตนเอง
- บันทึกสะท้อนสิ่งที่ได้เรียนรู้
- ทดสอบย่อยเพื่อทบทวนความรู้ความเข้าใจในแต่ละบท
- สอบปฏิบัติเขียนโปรแกรมเพื่อการบูรณาการความรู้ที่ได้รับทั้งกลางภาคและปลายภาค
- ใช้เอกสารประกอบการสอนเพื่อใช้ในการทบทวนความรู้ที่ได้รับและฝึกทำแบบฝึกหัดท้ายบท

## สื่อการเรียนรู้

- เอกสารประกอบการสอนรายวิชา 477-201 การเขียนโปรแกรมคอมพิวเตอร์
- คลิปวิดีโอออนไลน์สื่อการสอนรายวิชา 477-201 การเขียนโปรแกรมคอมพิวเตอร์
  - <https://www.youtube.com/classstartacademy>
- ชั้นเรียนออนไลน์รายวิชา 477-201 การเขียนโปรแกรมคอมพิวเตอร์
  - <https://classstart.org>
- เว็บไซต์
  - <https://www.python.org/>
  - <https://code.org/>
  - <https://www.tutorialspoint.com/python3/index.htm>
  - <https://www.w3schools.com/python/>

## การประเมินผล

- ร้อยละ 30 สอบกลางภาคแบบปฏิบัติการเขียนโปรแกรม
- ร้อยละ 30 สอบปลายภาคแบบปฏิบัติการเขียนโปรแกรม
- ร้อยละ 20 แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- ร้อยละ 20 แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ทำการประเมินผล

### สัปดาห์ที่ 3

#### วัตถุประสงค์การประเมิน

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่1-2 ความรู้เบื้องต้นเกี่ยวกับ Python และ ส่วนประกอบต่างๆ ของภาษา Python แล้วผู้เรียนสามารถ

- อธิบายหลักการหน้าที่ของโปรแกรมคอมพิวเตอร์ได้
- อธิบายการทำงานของ Python ได้
- วิเคราะห์อัลกอริทึมและเขียนผังงานได้
- ติดตั้งโปรแกรมจัดการ Python Runtime และ IDE ได้
- ใช้ตัวแปรและเครื่องหมายคำนวณทางคณิตศาสตร์ในการเขียนโปรแกรมได้
- ใช้คำสั่งพื้นฐาน print และ input ในการเขียนโปรแกรมได้
- เขียนโปรแกรมภาษา Python ที่มีโครงสร้างตามลำดับได้

#### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม

- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 5

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรมภาษา Python เกี่ยวกับเนื้อหาบทที่ 3 ประโยคเงื่อนไขในภาษา Python แล้วผู้เรียนสามารถ

- อธิบายหลักการของ Boolean Expression ได้
- เขียนโปรแกรม Python ที่มีโครงสร้างทางเลือกโดยมีเงื่อนไขได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 7

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 4 การเขียนฟังก์ชันในภาษา Python แล้วผู้เรียนสามารถ

- อธิบายหลักการของฟังก์ชันหรือโปรแกรมย่อย
- เข้าใจวิธีการแบ่งโปรแกรมใหญ่เป็นโปรแกรมย่อยและเรียกใช้โปรแกรมย่อยได้
- เขียนฟังก์ชันในภาษา Python ได้

#### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

### สัปดาห์ที่ 8

#### วัตถุประสงค์การประเมิน

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 5 การใช้ประโยคสั่งทำงานซ้ำในภาษา Python แล้วผู้เรียนสามารถ

- เข้าใจกระบวนการทำงานแบบวนซ้ำ
- เขียนโปรแกรมภาษา Python โดยใช้โครงสร้างการทำงานซ้ำแบบ **for** และ **while** ได้
- เขียนโปรแกรมภาษา Python โดยใช้ฟังก์ชัน **range()** ได้
- เขียนโปรแกรมภาษา Python สร้างฟังก์ชันที่เรียกตัวเองได้

#### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs



### สัปดาห์ที่ 9 (สัปดาห์สอบกลางภาคเชิงปฏิบัติการ)

#### วัตถุประสงค์การประเมิน

เพื่อประเมินความรู้ความเข้าใจทั้งหมดของเนื้อหาบทที่ 1-5

#### วิธีการประเมิน

สอบปฏิบัติการเขียนโปรแกรม

### สัปดาห์ที่ 10

#### วัตถุประสงค์การประเมิน

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 6 ชนิดข้อมูล String แล้วผู้เรียนสามารถ

- อธิบายหลักการใช้และจัดการกับ String และตัวชี้
- อธิบายโครงสร้างข้อมูลที่ไม่เปลี่ยนแปลงไม่ได้
- เขียนโปรแกรมภาษา Python เพื่อจัดการกับ String ได้
- อธิบายและใช้เมธอดของ String ได้

#### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 11

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 7 ชนิดข้อมูล List แล้วผู้เรียนสามารถ

- อธิบายหลักการใช้และจัดการกับ List และตัวชี้
- อธิบายโครงสร้างข้อมูลที่สามารถเปลี่ยนแปลงได้
- เขียนโปรแกรมภาษา Python เพื่อจัดการกับ List ได้
- อธิบายและใช้เมธอดของ List ได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 12

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 8 ชนิดข้อมูล Dictionary แล้วผู้เรียนสามารถ

- อธิบายหลักการใช้และจัดการกับ Dictionary และตัวชี้
- เขียนโปรแกรมภาษา Python เพื่อจัดการกับ Dictionary ได้
- อธิบายและใช้เมธอดของ Dictionary ได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 13

### วัตถุประสงค์การประเมิน

เมื่อฟังการบรรยาย ถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 9 ชนิดข้อมูล Tuple แล้วผู้เรียนสามารถ

- อธิบายหลักการใช้และจัดการกับ Tuple และตัวชี้
- เขียนโปรแกรมภาษา Python เพื่อจัดการกับ Tuple ได้
- อธิบายและใช้เมธอดของ Tuple ได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 14

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถาถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 10 การจัดการ Files แล้วผู้เรียนสามารถ

- อธิบายหลักการใช้และจัดการกับ Files ใน Python ได้
- เขียนโปรแกรม Python เพื่อจัดการกับ Files ได้
- เขียนโปรแกรม Python เพื่อจัดการกับฐานข้อมูลแบบ Key-Value ได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

## สัปดาห์ที่ 15

### วัตถุประสงค์การเรียนรู้

เมื่อฟังการบรรยาย ถาถามตอบในชั้นเรียน และฝึกปฏิบัติการเขียนโปรแกรม Python เกี่ยวกับเนื้อหาบทที่ 11 Object-oriented programming (OOP) แล้วผู้เรียนสามารถ

- อธิบายหลักการพื้นฐานเขียนโปรแกรม Python แบบ OOP ได้
- อธิบายหลักการทำงานของคลาสและออบเจกต์
- อธิบายหลักการทำงานแบบ Inheritance และ Polymorphism ได้
- เขียนโปรแกรม Python แบบ OOP พื้นฐานได้

### วิธีการประเมิน

- แบบฝึกหัดย่อยแบบปฏิบัติการเขียนโปรแกรม
- แบบทดสอบย่อยแบบ MCQs

### สัปดาห์ที่ 16 (สัปดาห์สอบปลายภาคเชิงปฏิบัติการ)

#### วัตถุประสงค์การประเมิน

เพื่อประเมินความรู้ความเข้าใจทั้งหมดของเนื้อหาบทที่ 6-11

#### วิธีการประเมิน

สอบปฏิบัติการเขียนโปรแกรม

### เอกสารอ้างอิงที่ใช้ในการสอน

- Barry, P. (2016). *Head first python: A brain-friendly guide*. Sebastopol, CA, USA: O'Reilly Media.
- Beazley, D., & Jones, B. K. (2013). *Python cookbook*. Sebastopol, CA, USA: O'Reilly Media.
- Bouras, A. S. (2019). *Python and algorithmic thinking for the complete beginner (2nd edition): Learn to think like a programmer*. Independently published.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Cambridge, MA, USA: The MIT Press.
- Downey, A. B. (2015). *Think python: How to think like a computer scientist*. Sebastopol, CA, USA: O'Reilly Media.
- Foundation, P. S. (2019, January). *Python*. Retrieved from <https://www.python.org/>
- Guido, V. R. (2019, January). *Guido van rossum - personal home page*. Retrieved from <https://gvanrossum.github.io//help.html>
- Lubanovic, B. (2015). *Introducing python: Modern computing in simple packages*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2011). *Programming python: Powerful object-oriented programming*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2013). *Learning python*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2014). *Python pocket reference: Python in your pocket*. Sebastopol, CA, USA: O'Reilly Media.
- Ramalho, L. (2015). *Fluent python: Clear, concise, and effective programming*. Sebastopol, CA, USA: O'Reilly Media.
- Shuup. (2019, April). *25 of the most popular python and django websites*. Retrieved from <https://www.shuup.com/django/25-of-the-most-popular-python-and-django-websites/>

TIOBE. (2019, August). *The python programming language*. Retrieved from <https://www.tiobe.com/tiobe-index/python/>

## นักศึกษา

นักศึกษาที่เรียนวิชานี้เป็นนักศึกษาคณะวิทยาการจัดการ ภาควิชาบริหารธุรกิจ สาขาวิชาระบบสารสนเทศ  
ชั้นปีที่ 2 จำนวน 42 คน

## ผลการสอน

ระดับคะแนน	ช่วงคะแนน	จำนวนผู้เรียน	ร้อยละ
A	80 - 100	13	30.95
B+	75 - 79.99	6	14.29
B	70 - 74.99	4	9.52
C+	65 - 69.99	7	16.67
C	60 - 64.99	2	4.76
D+	55 - 59.99	2	4.76
D	50 - 54.99	6	14.29
E	0 - 49.99	2	4.76

ลงชื่อ

ดร.จันทวรรณ ปิยะวัฒน์

ผู้สอน

## References

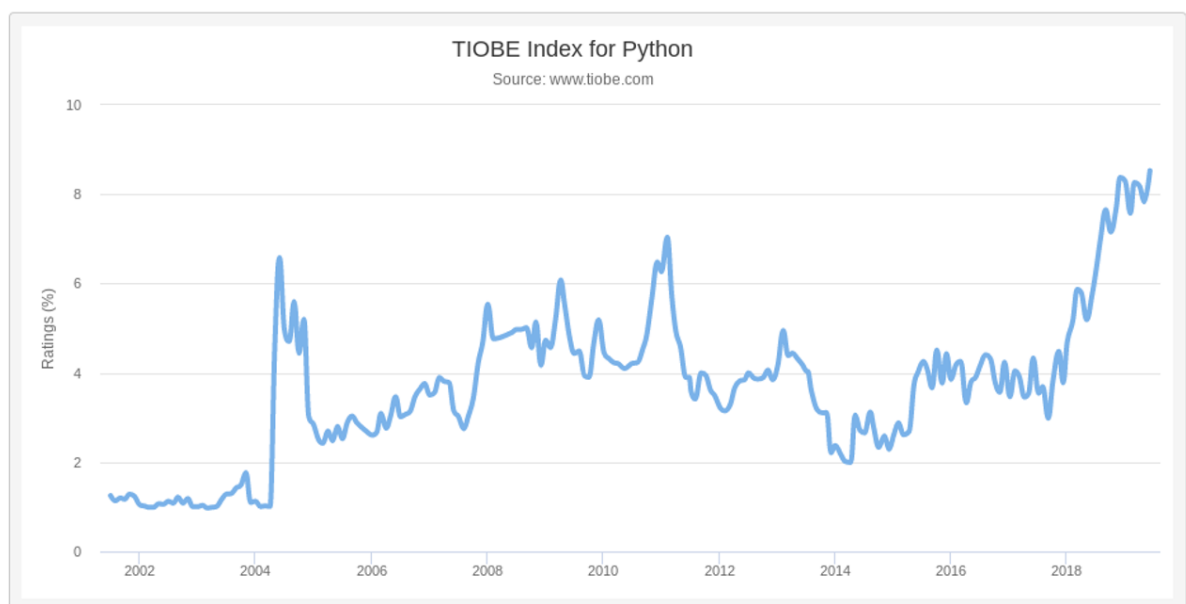
---



## 0.1 Python คืออะไร

ในปี ค.ศ. 1980 Mr. Guido van Rossum ได้พัฒนาภาษาโปรแกรมมิ่งขึ้นมาและให้ชื่อว่าภาษา Python และเผยแพร่ให้ใช้งานสู่สาธารณะชนในปี ค.ศ. 1991 (Guido, 2019) Python เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง ซึ่งไวยากรณ์ของภาษาระดับสูงนี้จะใกล้เคียงคำในภาษาอังกฤษทั่วไป (Downey, 2015) Python ถูกใช้ในการสร้างโมบายแอปพลิเคชัน เว็บไซต์ เว็บแอปพลิเคชัน ออนไลน์เซอร์วิส รวมทั้งใช้ในการวิเคราะห์ข้อมูล และคำนวณ ทางคณิตศาสตร์และวิทยาศาสตร์อย่างแพร่หลาย ตัวอย่างออนไลน์เซอร์วิสที่พัฒนาขึ้นด้วยภาษา Python ได้แก่ Instagram, Uber, Pinterest, Reddit, Spotify และ Dropbox (Shuup, 2019) โดยในระยะหลายปีที่ผ่านมา Python ได้รับความนิยมสูงขึ้นเรื่อยๆ โดยในเดือนมิถุนายน 2562 ดัชนีความนิยมภาษาโปรแกรมมิ่ง TIOBE ได้แสดงให้เห็นว่า Python เป็นภาษาโปรแกรมมิ่งที่ได้รับความนิยมเป็นอันดับที่ 3 เทียบกับภาษาโปรแกรมมิ่งอื่นๆ และมีความนิยมสูงสุดในรอบ 19 ปี (TIOBE, 2019)

รูปที่ 1: TIOBE Index for Python ในปี พ.ศ. 2562



หากเปรียบเทียบกับภาษาโปรแกรมมิ่งอื่นๆ แล้ว Python มีไวยากรณ์ภาษา (Syntax) ที่สามารถอ่านง่าย เข้าใจได้ง่าย และเรียนรู้ง่าย Python จึงเป็นภาษาที่เหมาะสมสำหรับการสอนการเขียนโปรแกรมโดยเฉพาะ

อย่างยิ่งในระดับเบื้องต้น อีกทั้งยังเป็นภาษาที่ยืดหยุ่นสามารถพัฒนาได้บนระบบปฏิบัติการที่หลากหลาย อาทิ Windows, Linux, OS/2, MacOS, iOS และ Android นอกจากนี้โปรแกรมเมอร์ทั่วโลกได้พัฒนาไลบรารี (Libraries) ขึ้นมาจำนวนมากสำหรับต่อยอด การทำงานของภาษา Python พื้นฐาน เช่น Django, Numpy, Pandas, Matplotlib, Flask, Web2py เป็นต้น (Foundation, 2019)

## 0.2 Python ทำงานอย่างไร

ภาษาโปรแกรมมีระดับสูงจะต้องถูกโปรแกรมแปลภาษา เช่น คอมไพเลอร์ (Compiler) หรือ อินเทอร์พรีเตอร์ (Interpreter) ทำการแปลภาษาระดับสูงให้กลายเป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจก่อน (Lutz, 2013) ภาษาตระกูลที่ต้องใช้ Compiler เพื่อแปลงเป็นภาษาคอมพิวเตอร์ซึ่งเป็นภาษาที่มนุษย์อ่านไม่ออกแล้วจึงจะทำงานได้ เช่น ภาษา Java ภาษา C หรือภาษา C++ ภาษาพวกนี้จะได้โปรแกรมที่ทำงานรวดเร็วมาก แต่ก็ยากที่จะเรียนรู้ในช่วงการฝึกฝนการเขียน Programming ใหม่ๆ (Barry, 2016)

แต่สำหรับภาษา Python เมื่อได้ Source code ที่เป็นนามสกุลไฟล์ `.py` แล้ว โปรแกรมจะถูกคอมไพล์โดยคอมไพเลอร์ของ Python เพื่อแปลคำสั่ง Python ให้เป็นคำสั่งแบบ Bytecode และบันทึกไว้ในไฟล์นามสกุล `.pyc` ต่อมาเมื่อผู้ใช้ต้องการ Run ไฟล์นี้ อินเทอร์พรีเตอร์ (Interpreter) ก็จะแปลง Bytecode เป็นภาษาเครื่องสำหรับการดำเนินการโดยตรงบนฮาร์ดแวร์ (Beazley & Jones, 2013) อาจเรียกได้ว่า Python เป็นภาษาลูกครึ่งและเรียนรู้ได้ง่าย เหตุผลที่ Python ทำการคอมไพล์เป็น Bytecode เป็นรหัสกลางไว้ก่อนนั้น นั่นก็เพราะ Python ถูกออกแบบมาให้เป็นภาษาการเขียนโปรแกรมที่ไม่ขึ้นกับแพลตฟอร์ม ซึ่งหมายความว่ามีการเขียนโปรแกรมหนึ่งครั้ง แต่สามารถเรียกใช้งานบนอุปกรณ์ใดก็ได้ แต่จะต้องติดตั้ง Python เวอร์ชันที่เหมาะสม

## 0.3 อัลกอริทึม (Algorithm) และ แผนผัง (Flowchart)

อัลกอริทึม (Algorithm) หมายถึง กระบวนการทีละขั้นตอนเพื่อแก้ไขปัญหาที่กำหนดอย่างชัดเจน โดยทั่วไปจะมีสามขั้นตอนหลัก คือ มีการนำเข้าสู่ข้อมูลหรืออินพุต แล้วนำมาประมวลผล และแสดงผลลัพธ์ออกมา

(Bouras, 2019) ตัวอย่างเช่น โจทย์ให้หาค่าเฉลี่ยของตัวเลขที่รับมาจากผู้ใช้งานสามค่า จะสามารถเขียนเป็นขั้นตอนได้ดังนี้คือ

1. ขั้นตอนที่หนึ่ง การนำเข้าข้อมูล
  - แจ้งให้ผู้ใช้งานป้อนหมายเลขที่หนึ่ง
  - แจ้งให้ผู้ใช้งานป้อนหมายเลขที่สอง
  - แจ้งให้ผู้ใช้งานป้อนหมายเลขที่สาม
2. ขั้นตอนที่สอง การประมวลผลข้อมูล
  - คำนวณผลรวมของเลขทั้งสามจำนวน
  - หาค่าเฉลี่ยด้วยสาม
3. ขั้นตอนที่สาม การแสดงผลลัพธ์
  - แสดงผลลัพธ์ออกทางหน้าจอ

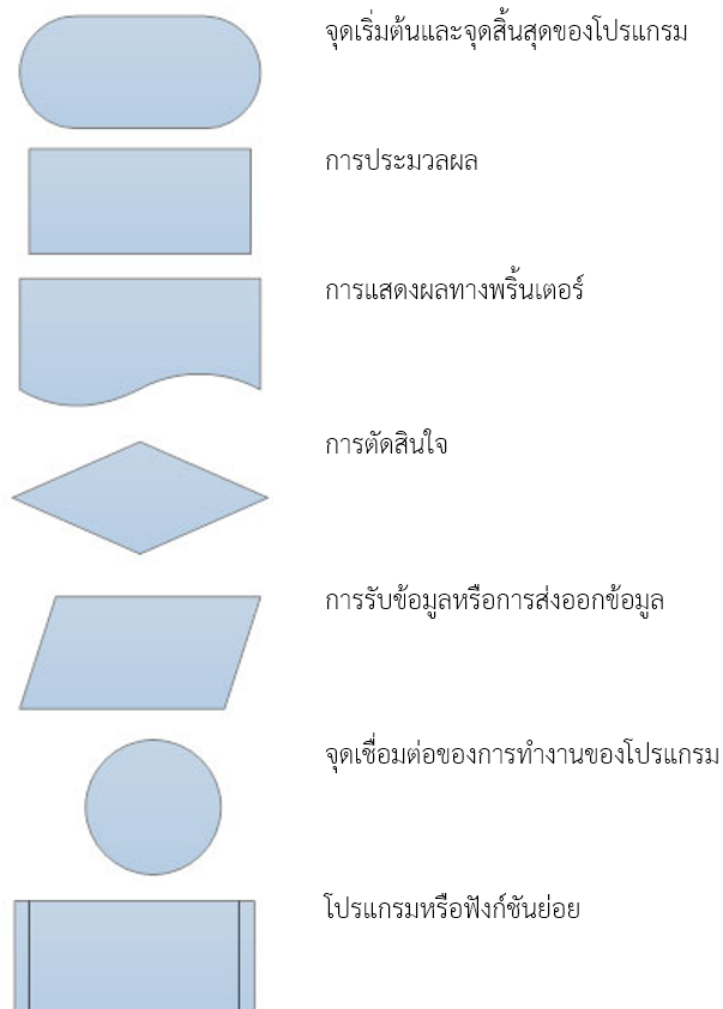
ส่วนผังงาน (Flowchart) เป็นการนำเสนอการไหลของอัลกอริทึมในรูปแบบของสัญลักษณ์จากคำสั่งหนึ่งไปยังอีกต่อไปจนถึงจุดสิ้นสุดของอัลกอริทึม (Cormen, Leiserson, Rivest, & Stein, 2009) สัญลักษณ์ที่ใช้บ่อยสำหรับผังงานมีดังต่อไปนี้

ตัวอย่างการเขียนผังงานจากอัลกอริทึมด้านบนสามารถเขียนได้ดังนี้

## 0.4 การติดตั้งโปรแกรม Python Runtime

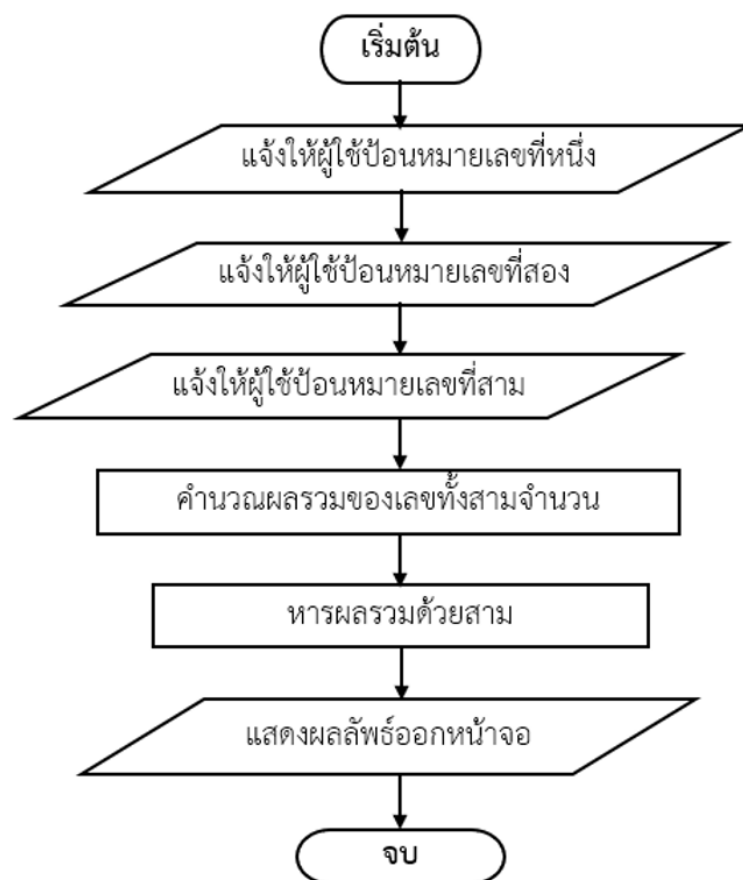
การติดตั้งโปรแกรม Python Runtime คือการติดตั้งโปรแกรมที่ทำให้นักพัฒนาซอฟต์แวร์สามารถใช้งานโปรแกรมที่เขียนขึ้นด้วยภาษา Python เองได้ ให้เข้าไปที่เว็บไซต์ <https://www.python.org/> (Foundation, 2019) ไปที่ Download for Windows แล้วเลือก Python 3.7.0 แล้วทำการติดตั้งให้เรียบร้อยลงในเครื่อง และให้เลือก Add Python 3.7 To Path เพื่อที่จะสามารถใช้ Python ได้ที่ Command Line หลังจากนั้นจะเห็นได้ว่าที่สตาร์ทเมนูโปรแกรม Python 3.7 จะถูกสร้างขึ้น ในโฟลเดอร์นี้จะมีโปรแกรมชื่อว่า Idle ซึ่งเป็น Integrated Development Environment หรือ เครื่องมือที่ช่วยในการพัฒนาโปรแกรมที่ใช้งานง่าย

รูปที่ 2: สัญลักษณ์ผังงาน (Flowchart)

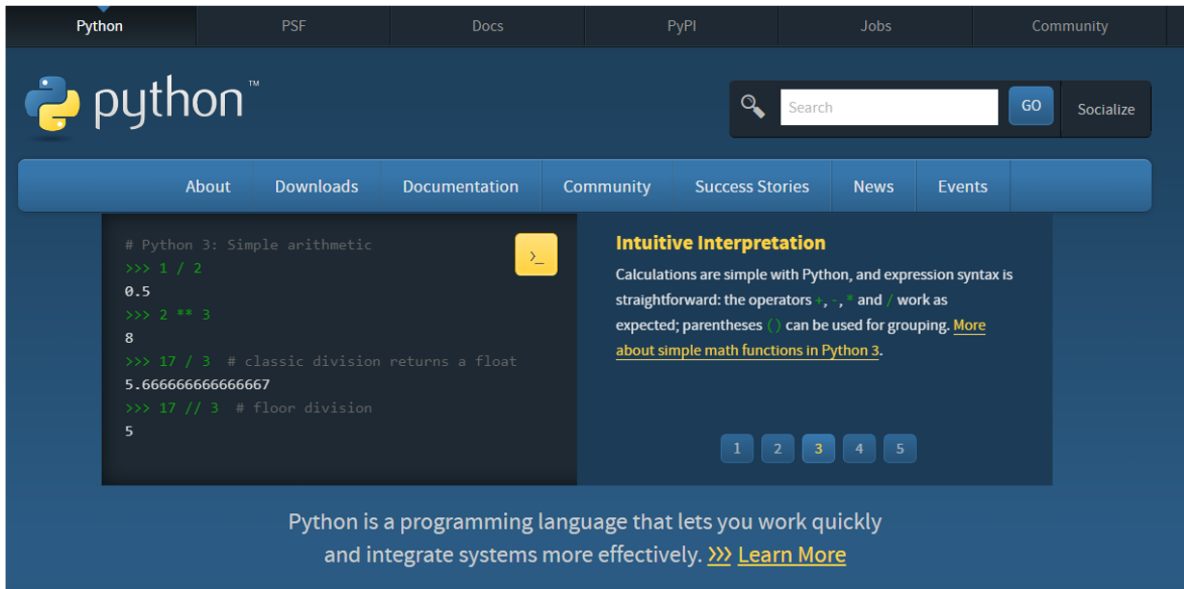


เหมาะแก่การเขียนโปรแกรมเบื้องต้น โดยจะมีทั้ง Text Editor และ Interactive Shell เวลาใช้งานควรเปิดไว้ 2 หน้าต่าง ด้านซ้ายมือเป็น Source code ด้านขวามือเป็น Python Shell เพื่อใช้ดูผลลัพธ์ในการ Run โปรแกรม ที่เขียนขึ้น

รูปที่ 3: ตัวอย่างการเขียนผังงาน



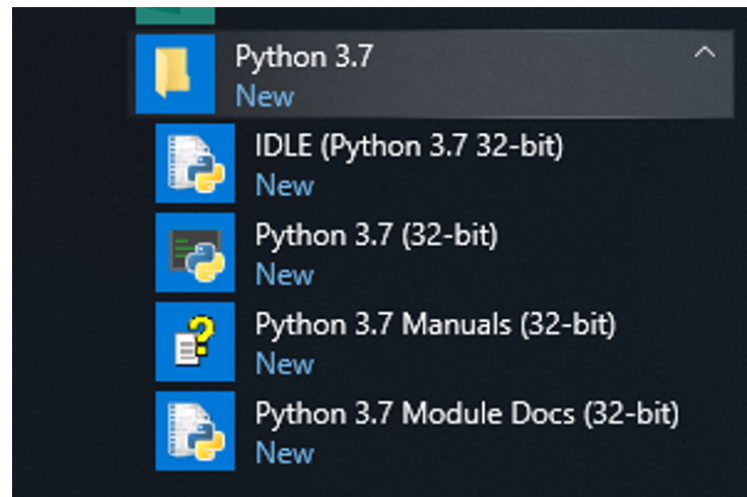
รูปที่ 4: การ Download Python 3.7



รูปที่ 5: เลือก Add Python 3.7 to PATH



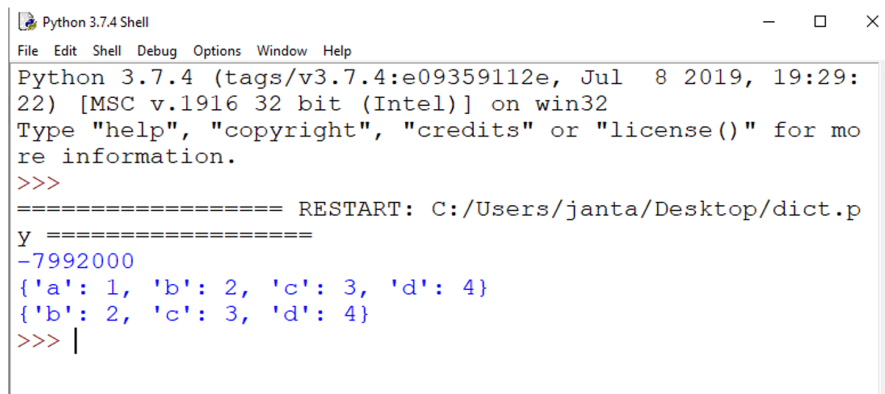
รูปที่ 6: โฟลเดอร์ Python 3.7



รูปที่ 7: ตัวอย่างหน้าโปรแกรม Idle หน้า Editor

```
*dict.py - C:\Users\janta\Desktop\temp4\Python dict\dict.py (3.7.4)*
File Edit Format Run Options Window Help
for x in range(1,n+1): d[x]=x*x
print(d)
#หาผลรวมของ values
pets={'cats':100,'dogs':60,'pigs':300}
print(sum(pets.values()))
#ตรวจสอบว่ามี key อยู่ใน dict หรือไม่
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
def is_key_present(x):
    if x in d:
        print('Key is present in the dictionary')
    else:
        print('Key is not present in the dictionary')
is_key_present(5)
is_key_present(9)
```

รูปที่ 8: ตัวอย่างหน้าโปรแกรม Idle หน้า Python Shell



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:
22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for mo
re information.
>>>
===== RESTART: C:/Users/janta/Desktop/dict.p
y =====
-7992000
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
{'b': 2, 'c': 3, 'd': 4}
>>> |
```



## 0.5 ตัวแปร (Variables)

ตัวแปร (Variables) คือชื่อที่ผู้เขียนโปรแกรมกำหนดขึ้นมาเอง เพื่อใช้สำหรับการเก็บค่าข้อมูลในการเขียนโปรแกรมไว้ในหน่วยความจำของเครื่องคอมพิวเตอร์ โดยในภาษา Python ไม่ต้องระบุประเภทของตัวแปรไว้ในตอนที่ประกาศการตั้งชื่อตัวแปร ดังตัวอย่างต่อไปนี้

```
1 >>> a = 1
2 >>> a
3 1
4 >>> b = 2
5 >>> b
6 2
7 >>> a + b
8 3
9 >>> vat = 7
10 >>> vat
11 7
```

## 0.6 การตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรสำหรับภาษา Python มีเงื่อนไขดังนี้

1. ให้ขึ้นต้นด้วยอักษรตัวภาษาอังกฤษตัวใหญ่หรือตัวเล็กตั้งแต่ Aa ถึง Zz เท่านั้น
2. ประกอบด้วยตัวอักษรหรือตัวเลข 0 ถึงเลข 9 หรือตัวขีดกลาง Underscore (\_) แต่ห้ามมีช่องว่าง
3. ตัวเลข 0-9 จะนำหน้าชื่อตัวแปรไม่ได้
4. ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่เป็นตัวแปรคนละตัวกัน (Case-Sensitive) เช่น Name ไม่ใช่ตัวแปรเดียวกันกับ name
5. ใช้ใส่เครื่องหมาย = ในการตั้งตัวแปรหรือให้ค่าแก่ตัวแปร
6. การตั้งชื่อตัวแปรควรตั้งอย่างสมเหตุสมผล
7. ภาษา Python จะมีคำที่ถูกสงวนไว้ในการเขียนโปรแกรม หรือ Keywords ซึ่งห้ามนำมาใช้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน หรือ ชื่อคลาส

## 0.7 การตั้งชื่อตัวแปรพร้อมกันหลายตัวแปร

ในการเขียนโปรแกรมภาษา Python ผู้เขียนโปรแกรมสามารถตั้งชื่อตัวแปรพร้อมกันได้หลายตัวแปร โดยพิมพ์ตัวแปรแต่ละตัวในบรรทัดเดียวกันและคั่นแต่ละตัวแปรด้วยเครื่องหมายคอมม่า (,) ตามด้วยเครื่องหมาย (=) และกำหนดค่าตามลงไปตามลำดับการวางตัวแปร ดังตัวอย่างต่อไปนี้

```
1 >>> a, b, c = 1, 'Jan', 2.36
2 >>> a
3 1
4 >>> b
5 'Jan'
6 >>> c
7 2.36
8 >>>
```

## 0.8 คำสงวน (Keywords)

คำสงวน (Keywords) ในภาษา Python จะมีการสงวนคำบางคำไว้เฉพาะเพื่อใช้เป็นคำสั่งของภาษา โดยผู้เขียนโปรแกรมไม่ควรนำมาใช้ในการตั้งชื่อตัวแปร โดยคำสงวนของภาษา Python มีดังต่อไปนี้ (Lutz, 2014)

ตารางที่ 1: คำสงวนในภาษา Python

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	break
except	in	raise		

## 0.9 เลขประจำตัวตำแหน่งของตัวแปร

ตัวแปรจะชี้ไปที่หน่วยความจำในเครื่องคอมพิวเตอร์ซึ่งเก็บค่าของตัวแปรหรือ Value นั้นๆ อยู่ ฉะนั้นเมื่อเราพิมพ์ `a` ดังในตัวอย่าง คอมพิวเตอร์จึงแสดงเลข 1 ออกมา นอกจากนี้พื้นที่ที่เก็บค่านั้นนั้นจะมีที่อยู่อยู่บนหน่วยความจำมีหมายเลขประจำตำแหน่งอีกด้วย โดยใช้คำสั่ง `id()` เพื่อแสดงเลขประจำตำแหน่ง

```
1 >>> a
2 1
3 >>> id(a)
4 1538021648
```

## 0.10 ชนิดของข้อมูล (Types)

สิ่งที่อยู่ในหน่วยความจำมีชนิดของข้อมูลหรือ Type อยู่ด้วย โดยใช้คำสั่ง `{type()}` เพื่อดูประเภทของข้อมูล ในภาษา Python มีประเภทของข้อมูลหลายๆ แบบ (Ramalho, 2015) ทั้งแบบที่เป็นตัวเลขแบบจำนวนเต็ม ตัวเลขแบบมีจุดทศนิยม ตัวเลขที่มีค่าเป็นบวกหรือลบ ตัวอักษร ข้อความ และตรรกศาสตร์

1. none คือ Nothing ไม่มีอะไร
2. int หรือ Integer คือตัวเลข เช่น 50 หรือ 630 เป็นต้น
3. bool หรือ Boolean คือค่าถูกผิด เช่น True หรือ False เป็นต้น
4. float หรือ floating Point คือจำนวนทศนิยม เช่น 5.6 หรือ 4.23 เป็นต้น
5. str หรือ String หรือข้อความ ซึ่งจะอยู่ภายใต้เครื่องหมาย ฟันหุ้ (" ") หรือ ฟันทอ (' ') เช่น `"This is my dog."` หรือ `'Jantawan'`

ตัวอย่างการแสดงประเภทของข้อมูลมีดังต่อไปนี้

```
1 >>> a = 1
2 >>> a
3 1
4 >>> type(a)
```

```
5 <class 'int'>
6 >>> firstname = 'Jantawan'
7 >>> firstname
8 'Jantawan'
9 >>> lastname = 'Piyawat'
10 >>> lastname
11 'Piyawat'
12 >>> id(firstname)
13 67626832
14 >>> type(firstname)
15 <class 'str'>
```

```
1 >>> n = None
2 >>> n
3 >>> id(n)
4 263420692
5 >>> type(n)
6 <class 'NoneType'>
7 >>> yes = True
8 >>> no = False
9 >>> type(yes)
10 <class 'bool'>
11 >>> degree = 1.1
12 >>> id(degree)
13 72213072
14 >>> type(degree)
15 <class 'float'>
```

## 0.11 เครื่องหมายสำหรับการคำนวณทางคณิตศาสตร์

### 0.11.1 การคำนวณทางคณิตศาสตร์ (Arithmetic Operators)

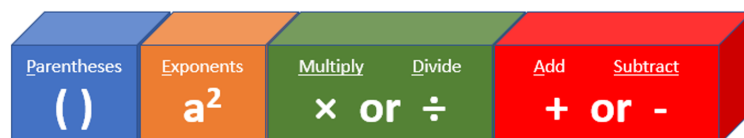
เครื่องหมายสำหรับการคำนวณเรียกว่า Arithmetic Operators เช่น เครื่องหมายบวก ลบ คูณ หาร การยกกำลัง การหารเอาเศษ การหารเอาจำนวนเต็ม เป็นต้น การคำนวณทางคณิตศาสตร์แบบซับซ้อนจะต้องมีลำดับในการคำนวณซึ่งเหมือนกับการคำนวณคณิตศาสตร์ทั่วไป คือ ในการแก้สมการทางคณิตศาสตร์จะต้องทำในวงเล็บก่อน ตามด้วยเลขยกกำลัง แล้วจึงตามด้วย คูณหรือหารโดยคำนวณจากซ้ายไปขวา แล้วตามด้วยบวกหรือลบโดยคำนวณจากซ้ายไปขวาเช่นกัน โดยให้จำคำว่า **PEMDAS** ซึ่งเป็นตัวอักษรภาษาอังกฤษ

ตารางที่ 2: สัญลักษณ์การคำนวณทางคณิตศาสตร์

สัญลักษณ์คำนวณ	ชื่อการคำนวณ	ตัวอย่าง
+	บวก	$a + b$
-	ลบ	$a - b$
*	คูณ	$a * b$
/	หาร	$a / b$
//	หารปัดเศษทิ้ง	$a // b$
%	เศษของการหาร	$a \% b$
**	ยกกำลัง	$a ** b$

ตัวแรกของการคำนวณทางคณิตศาสตร์ในภาษา Python คือ Parentheses (วงเล็บ), Exponents (ยกกำลัง), Multiply (คูณ), Divide (หาร), Add (บวก), และ Subtract (ลบ) โดยภาษา Python ใช้สัญลักษณ์การคำนวณทางคณิตศาสตร์ดังตารางต่อไปนี้ (Lubanovic, 2015)

รูปที่ 9: ลำดับในการคำนวณ



ตัวอย่างการคำนวณทางคณิตศาสตร์ในภาษา Python มีดังต่อไปนี้

```

1 >>> a=1
2 >>> b=2
3 >>> a+b
4 3
5 >>> a-b
6 -1
7 >>> c = b-a
8 >>> c
9 1

```

### 0.11.2 รูปแบบการเขียนการคำนวณทางคณิตศาสตร์แบบย่อ

ในภาษา Python ผู้เขียนโปรแกรมสามารถเขียนการคำนวณทางคณิตศาสตร์แบบลดรูปหรือแบบย่อได้ ดังตารางสัญลักษณ์การคำนวณทางคณิตศาสตร์แบบย่อต่อไปนี้

ตารางที่ 3: สัญลักษณ์การคำนวณทางคณิตศาสตร์แบบย่อ

การคำนวณ	ตัวอย่าง	เทียบเท่ากับ
+=	C += a	C = C + a
-=	C -= a	C = C - a
*=	C *= a	C = C * a
/=	C /= a	C = C / a
//=	C //= a	C = C // a
%=	C %= a	C = C % a
**=	C **= a	C = C ** a

### 0.11.3 การจัดการข้อความด้วยเครื่องหมายทางคณิตศาสตร์

เครื่องหมายที่ใช้ในการคำนวณทางคณิตศาสตร์เมื่อถูกนำมาใช้กับข้อความ (String) จะเป็นอีกความหมายหนึ่ง เช่น การใช้เครื่องหมายบวกเชื่อมต่อระหว่างสตริง 2 ตัว หรือ การใช้เครื่องหมายดอกจันเป็นการเพิ่มสตริงเดียวกันตามจำนวนครั้งของการคูณ

```
1 >>> firstname
2 'Jantawan'
3 >>> lastname
4 'Piyawat'
5 >>> firstname + lastname
6 'JantawanPiyawat'
7 >>> firstname + ' ' + lastname
8 'Jantawan Piyawat'
```

```

9 >>> firstname * 3
10 'JantawanJantawanJantawan'

```

## 0.12 Expressions และ Statements

Expression หมายถึงการใช้เครื่องหมายคำนวณและการใช้ตัวแปรและค่าของตัวแปรเพื่อหาผลลัพธ์ออกมา เอา Expression มาประกอบกันจะเรียกว่า Statement ดังนั้น Statement ก็คือคำสั่งเรียงต่อกันนั่นเอง เพื่อใช้ในการสั่งงานคอมพิวเตอร์ด้วยภาษาคอมพิวเตอร์

ตัวอย่าง Expression เป็นดังต่อไปนี้

```

1 >>> 1+2
2 3

```

ตัวอย่าง Statement เป็นดังต่อไปนี้

```

1 >>> c = a + b
2 >>> c
3 3
4 >>> print('hello world.')
5 hello world.

```

## 0.13 การเขียนข้อความอธิบายโปรแกรมโดยใช้ Comment

Comment คือสิ่งที่เราเขียนใน Source Code ของโปรแกรมแต่คอมพิวเตอร์ไม่ต้องแปลผล เพื่อใช้ในการเขียนข้อความประกอบคำอธิบายในการสื่อสารระหว่างผู้เขียนโปรแกรมด้วยกัน หรือเป็นการเตือนความจำของผู้เขียนโปรแกรมเอง โดย Comment ในภาษา Python นำหน้าด้วยเครื่องหมายชาร์ป (#) แล้วหลังจากนั้นตามด้วยข้อความอะไรก็ได้ ถ้าจะเขียน Comment หลายๆ บรรทัดจะต้องใช้เครื่องหมายฟันหนู (" - ") หรือฟันทอง (' - ') 3 ชุด ข้างในใส่ค่า ผลที่ได้จะมีเครื่องหมาย Backslash n (\n) หมายถึงการขึ้นบรรทัดใหม่

ตัวอย่างการใช้ Comment ในบรรทัดเดียว เป็นดังต่อไปนี้

```
1 >>> print('Hello world!')
2 Hello world!
3 >>> # Hello how are you doing?
```

ตัวอย่างการใช้ Comment ในหลายบรรทัด เป็นดังต่อไปนี้

```
1 >>> x = '''
2 hello
3 1
4 2
5 3
6 '''
7 >>> x
8 '\nhello\n1\n2\n3\n'
9 >>> print(x)
10 hello
11 1
12 2
13 3
```

## 0.14 Source Code

ที่ผ่านมาเป็นการเขียนโปรแกรมแบบ Interactive คือเขียนบน Python Shell แล้วโปรแกรมจะแสดงผลออกมาได้เลย ซึ่งเรียกว่าการทำงานแบบ Interpreter เป็นการใส่คำสั่งไปที่ Prompt และ Python จะแสดงผลของคำสั่งนั้นออกมาเลย แต่ในความเป็นจริงแล้วจะเขียนโปรแกรมหลายๆ บรรทัดแล้วสั่งโปรแกรมทำงานทีเดียวพร้อมกัน เราจะเขียนไว้ในไฟล์นั้นเรียกว่า Source Code โดยที่ Source Code ของภาษา Python นามสกุลจะเป็น **.py** เวลาใช้โปรแกรม Idle ให้กดที่เมนู File เลือก New เขียน Source Code แล้วให้กด Run ถ้าหากจะกดรันโปรแกรมอีกสักครั้งให้กด F5

ตัวอย่าง Python Source Code เป็นดังต่อไปนี้

```
1 x = 7
2 y = 6
3 if x == y: print('x and y are equal.')
4 else:
5     if x < y: print('x is less than y.')
6     else: print('x is greater than y.')
```



ตัวอย่างผลลัพธ์ที่ได้จากการประมวลผล Source Code เป็นดังต่อไปนี้

```
1 x is greater than y.
```

## 0.15 คำสั่ง print (ตัวแปรหรือข้อมูล)

`print()` เป็นฟังก์ชันที่ใช้ในการแสดงผลตัวแปรหรือข้อมูลออกทางหน้าจอ ดังตัวอย่างต่อไปนี้

```
1 >>> print('Hello world!')
2 Hello world!
3 >>>
```

## 0.16 การใช้คำสั่ง input() รับค่าจากแป้นพิมพ์

คำสั่ง `input(Prompt)` เป็นคำสั่งสำหรับรับข้อมูลจากผู้ใช้ในการพิมพ์ผ่านแป้นพิมพ์ ดังตัวอย่างต่อไปนี้

```
1 >>> name=input('What is your name? ')
2 What is your name? Jantawan
3 >>> print('Hello, ', name, end='.')
4 Hello, Jantawan.
```

## 0.17 แบบฝึกหัด

1. จงหาเลขประจำตำแหน่งของข้อมูลต่อไปนี้

- love = 2
- mom = “Jan”
- wed = True
- fah = 39.2

2. จงหาประเภทของข้อมูลต่อไปนี้

- `love = 2`
- `mom = 'Jan'`
- `wed = True`
- `fah = 39.2`
- `money = '22'`

### 3. จงแสดงผลต่อไปนี้

- ตั้งค่าตัวแปร `dog, cat`
- แสดงข้อความ `I have 3 dogs and 2 cats.`

### 4. จงรับค่าจากผู้ใช้และแสดงผลต่อไปนี้

- ตั้งตัวแปร `name`
- รับค่าด้วยข้อความว่า กรุณาใส่ชื่อของคุณ
- แสดงข้อความ สวัสดีค่ะคุณ

### 5. จงคำนวณหาค่าตัวเลขต่อไปนี้

- หาค่าพื้นที่สี่เหลี่ยม กว้าง 5 เมตร ยาว 3 เมตร
- หาค่าพื้นที่สามเหลี่ยม สูง 5 เมตร ฐาน 3 เมตร

### 6. ให้ `a = 3`, `b = 4`, `c = 5` จงหาค่าต่อไปนี้

- `a == a*1`
- `a != b`
- `a > b`
- `b < c`
- `a+1 >= c`
- `c <= a+b`

## บรรณานุกรม

- Barry, P. (2016). *Head first python: A brain-friendly guide*. Sebastopol, CA, USA: O'Reilly Media.
- Beazley, D., & Jones, B. K. (2013). *Python cookbook*. Sebastopol, CA, USA: O'Reilly Media.
- Bouras, A. S. (2019). *Python and algorithmic thinking for the complete beginner (2nd edition): Learn to think like a programmer*. Independently published.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. Cambridge, MA, USA: The MIT Press.
- Downey, A. B. (2015). *Think python: How to think like a computer scientist*. Sebastopol, CA, USA: O'Reilly Media.
- Foundation, P. S. (2019, January). *Python*. Retrieved from <https://www.python.org/>
- Guido, V. R. (2019, January). *Guido van rossum - personal home page*. Retrieved from <https://gvanrossum.github.io/help.html>
- Lubanovic, B. (2015). *Introducing python: Modern computing in simple packages*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2011). *Programming python: Powerful object-oriented programming*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2013). *Learning python*. Sebastopol, CA, USA: O'Reilly Media.
- Lutz, M. (2014). *Python pocket reference: Python in your pocket*. Sebastopol, CA, USA: O'Reilly Media.
- Ramalho, L. (2015). *Fluent python: Clear, concise, and effective programming*. Sebastopol, CA, USA: O'Reilly Media.
- Shuup. (2019, April). *25 of the most popular python and django websites*. Retrieved from <https://www.shuup.com/django/25-of-the-most-popular-python-and-django-websites/>

## References

---

TIOBE. (2019, August). *The python programming language*. Retrieved from [\*\*https://www.tiobe.com/tiobe-index/python/\*\*](https://www.tiobe.com/tiobe-index/python/)