

Computational Science

Chapter 7: Introduction to Python Programming

Act 1: Sequential Statement

DTI1306 Computational Science

Department of Digital Technology for Education

Faculty of Education, Suan Sunandha Rajabhat University

Content Credit By: Asst.Prof.Nutthapat Kaewrattanapat, PhD.



Pasawut Cheerapakorn

Suan Sunandha Rajabhat University

Course Description:

วิเคราะห์ เทคนิค วิธีการขั้นตอนการแก้ปัญหา ทักษะการคิดเชิงคำนวณ เชิงนามธรรม ฝึกทักษะในการแก้ปัญหาโดยใช้ขั้นตอนการแก้ปัญหา การย่อยปัญหา การแสดงขั้นตอน การแก้ปัญหา โดยการเขียน บอกเล่า วาดภาพ หรือใช้สัญลักษณ์ ออกแบบและเขียนโปรแกรมโดยใช้ซอฟต์แวร์หรืออุปกรณ์ เทคโนโลยีเบื้องต้น เพื่อไปประยุกต์ใช้ในการแก้ปัญหาในชีวิตประจำวัน ในการตัดสินใจได้อย่างมีประสิทธิภาพและตระหนักถึงการใช้งานสารสนเทศอย่างปลอดภัย พัฒนาโครงงานทางเทคโนโลยีดิจิทัล เพื่อการศึกษาที่มีการบูรณาการกับสาขาอื่น ๆ อย่างสร้างสรรค์และเชื่อมโยงกับชีวิตจริง

The study analyzed how the process solutions, abstract thinking skills, computational skills to solve problems by using the steps to solve the problem of small steps to solve the problem by writing a story or painting the symbol, designers and programmers using software or technology introduction, to use the solution on a daily basis, decisions efficiently and realize the information securely, technological development project.

System Theory

Computational Thinking

Decomposition

Abstraction

Pattern Recognition

Algorithm Design

Design Thinking

Flowchart Design Standard

Flowgorithm

Computer Programming

Course Outline:

- Chapter 1 - Fundamental of Computational Science
- Chapter 2 - Digital Technology
- Chapter 3 - Digital Literacy
- Chapter 4 - Algorithm Design and Analysis
- Chapter 5 - Block-Based Programming
- Chapter 6 - Microbit for Learning
- **Chapter 7 - Introduction to Computer Programming**
- Chapter 8 - Project Design

Measurement and Evaluation:

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test 0%
- การมอบหมายงาน 20%
- สอบ Post-test 15%
- การมีส่วนร่วมในชั้นเรียน 5%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ 20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 – 81	B+	ดีมาก
74 – 77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 – 45	F	ตก

Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
1	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ (Introduction to Course)
2	บทที่ 1 พื้นฐานวิทยาการคำนวณ (Fundamental of Computational Science)
3	บทที่ 2 พื้นฐานเทคโนโลยีดิจิทัล (Digital Technology)
4	บทที่ 3 พื้นฐานการรู้เท่าทันดิจิทัล (Digital Literacy)
5	บทที่ 4 พื้นฐานการวิเคราะห์และออกแบบอัลกอริทึม (Algorithm Design and Analysis)
6	บทที่ 4 พื้นฐานการวิเคราะห์และออกแบบอัลกอริทึม (Algorithm Design and Analysis) [ต่อ]
7	บทที่ 5 การโปรแกรมแบบ Block-Based ด้วย Scratch (Block-Based Programming)
8	บทที่ 5 การโปรแกรมแบบ Block-Based ด้วย Scratch (Block-Based Programming) [ต่อ]

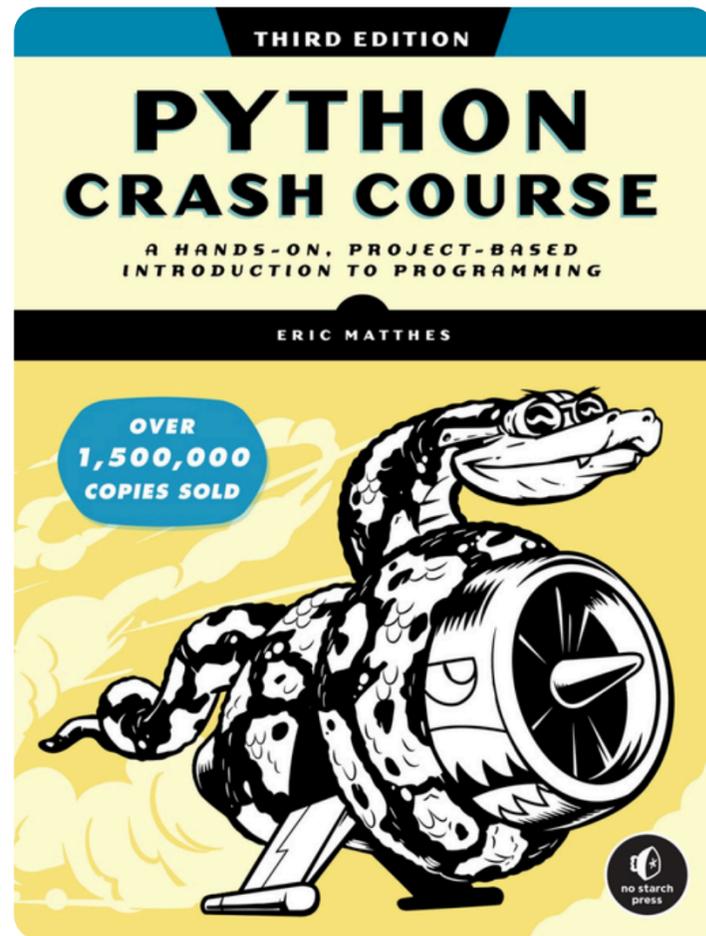
Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
9	สอบกลางภาค (Midterm Examination)
10	บทที่ 6 การโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้นด้วย Microbit (Microbit for Learning)
11	บทที่ 6 การโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้นด้วย Microbit (Microbit for Learning) [ต่อ]
12	บทที่ 7 การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น (Introduction to Computer Programming)
13	บทที่ 8 การออกแบบโครงงานทางเทคโนโลยีเพื่อการศึกษา (Project Design)
14	สอบปลายภาค (Final Examination)
15	นำเสนอและส่งโครงงาน (Project Pitching and Presentation)
16	

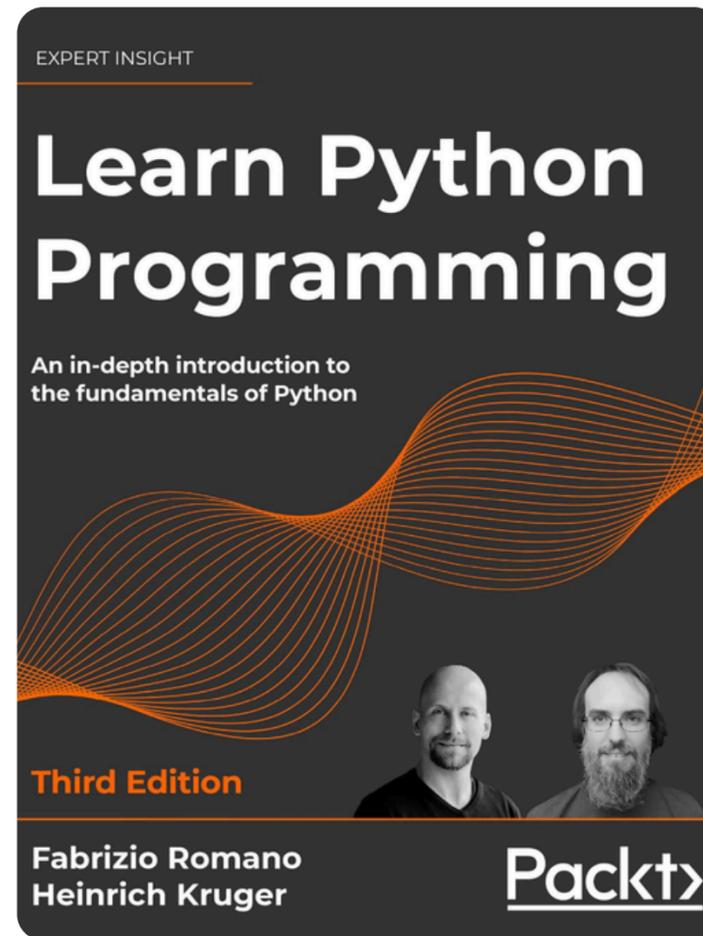
Chapter 7: Introduction to Python Programming

1. "Hello, World"
2. Function print() = ฟังก์ชันแสดงผลข้อมูล
3. Variable = ตัวแปร
4. Build-In Data Types = ประเภทข้อมูลในภาษา Python
5. Numbers = จำนวน
6. Integers = ค่าจำนวนเต็ม
7. Booleans = ค่าบูลีน
8. Real Numbers = ค่าจำนวนจริง
9. Function input()
10. Operand and Operators

Learning Materials Suggestion:



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- <https://www.w3schools.com/python/>
- <https://realpython.com/>
- <https://docs.python.org/3/tutorial/index.html>
- <https://pythontutor.com/>

Pre Test

Question:

1. ต้องการแสดงผลคำว่า Hello, World ต้องเติมคำสั่งใด

A Print("Hello, World")

B Print(Hello, World)

C print("Hello, World")

D print(Hello, World)

E ถูกทุกข้อ

F ไม่มีข้อถูก

Question:

2. ខ្នែងខ្នែង Reserved Word

A True

B None

C break

D class

E Control

Question:

3. ข้อใดให้ผลลัพธ์ Date:11/DEC/2023

A

```
print('Date:')  
print('11', 'DEC', '2023', sep='/')
```

C

```
print('Date:')  
print('11/', 'DEC/', '2023')
```

B

```
print('Date:', end='/n')  
print('11/ 'DEC/'2023', sep='/')
```

D

```
print('Date:', end="")  
print('11', 'DEC', '2023', sep='/')
```

Question:

4. ข้อใด คือ การตั้งชื่อตัวแปรแบบ Pascal Case

A My_Product_Price

D my_product_price

B MyProductPrice

E MYPRODUCTPRICE

C myProductPrice

Question:

5. จาก Code แสดงผลตามข้อใด

```
1. data1 = "50"  
2. data1 = float(data1)  
3. print(data1, 'is', type(data1))
```

A 50.0 is <class 'float'>

C "50" is <class 'str'>

B 50 is <class 'float'>

D "50.0" is <class 'float'>

Question:

6. จาก Code ข้อใดให้ค่า Boolean เป็น True

```
1. student1 = 50
2. student2 = 100
```

A print(student1>student2)

C print(student1<student2)

B print(student1=student2)

D print(student1!student2)

Question:

7. จาก Code ประเภทข้อมูลที่ได้รับเข้ามาเป็นประเภทใด

```
1. year = input("Birth Year:")  
Birth Year: 1983
```

A int

C string

B float

D real

Question:

8. តើខ្ញុំត្រូវប្រើ Keywords ណាខ្លះ ក្នុង Conditional Statement ក្នុងភាសា Python

A if

D if else

B if elif

E if elif else

C else

F if elseif else

Question:

9. จากคำสั่ง ไม่สามารถ แสดงผลข้อใด

```
import random
num = random.randrange(5, 10)
print(num)
```

A 5

C 10

E 8

B 7

D 9

F 6

Question:

10. จากคำสั่ง ไม่สามารถ แสดงผลข้อใด

```
1. count = 1
2. while count<=5:
3.     print(count)
4.     count+=1
```

A 1

B 2

C 3

D 4

E 5

F 6

Learning Tools:



<https://colab.research.google.com/>

การสอน Coding สำหรับผู้เริ่มต้น

1. Input
การนำเข้า

3. Process
การประมวลผล

2. Output
ผลผลิต/ผลลัพธ์

Operation Variable ตัวแปรดำเนินการ

ต้องการผลรวมของค่า 2 จำนวน

1. Input
การนำเข้า

จำนวนที่ 1

จำนวนที่ 2

3. Process
การประมวลผล

Operation Variable ตัวแปรดำเนินการ

2. Output
ผลผลิต

ต้องการผลรวมของค่า 2 จำนวน

1. Input
การนำเข้า

จำนวนที่ 1

จำนวนที่ 2

3. Process
การประมวลผล

Operation Variable ตัวแปรดำเนินการ

2. Output
ผลลัพธ์

ผลลัพธ์

ต้องการผลรวมของค่า 2 จำนวน

1. Input
การนำเข้า

จำนวนที่ 1

จำนวนที่ 2

3. Process
การประมวลผล

Operation Variable ตัวแปรดำเนินการ

1.

ผลลัพธ์



จำนวนที่ 1

+

จำนวนที่ 2

2.

แสดง

ผลลัพธ์

2. Output
ผลลัพธ์

ผลลัพธ์

ต้องการคำนวณส่วนลดราคาสินค้า

ราคาสินค้า 200 บาท ส่วนลด 10% ต้องจ่ายที่บาท

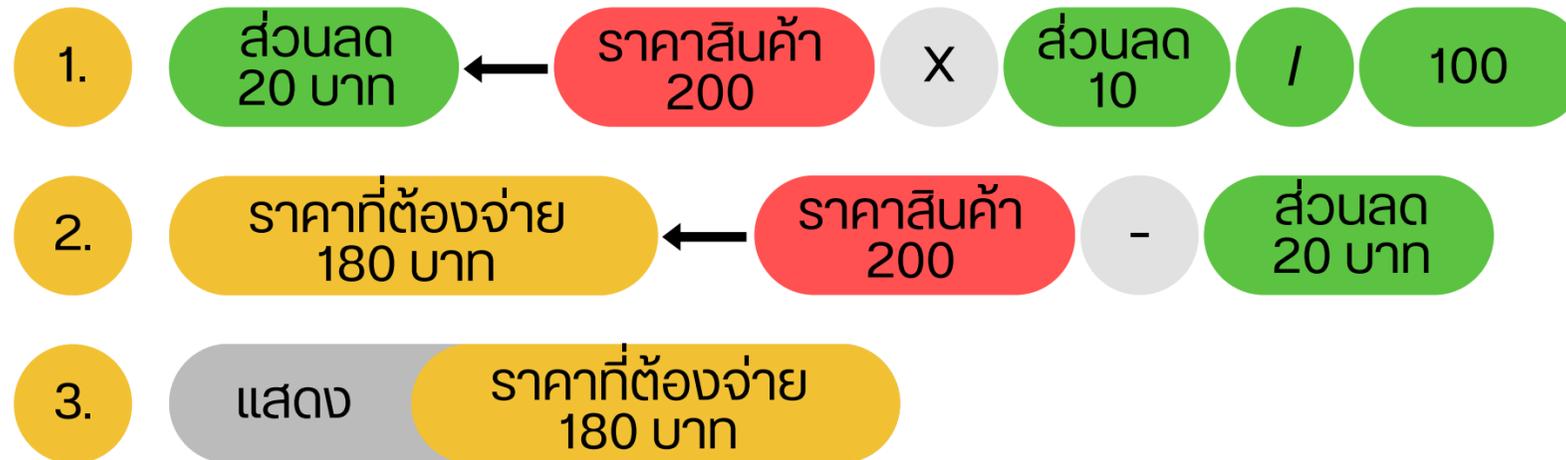
1. Input
การนำเข้า

ราคาสินค้า

ส่วนลด

3. Process
การประมวลผล

Operation Variable ตัวแปรดำเนินการ



ตัวแปร (Variable) เช่น ราคาสินค้า ส่วนลด ส่วน 50, 100 เป็นค่าของตัวแปร (Value)

2. Output
ผลลัพธ์

ราคาที่ต้องจ่าย

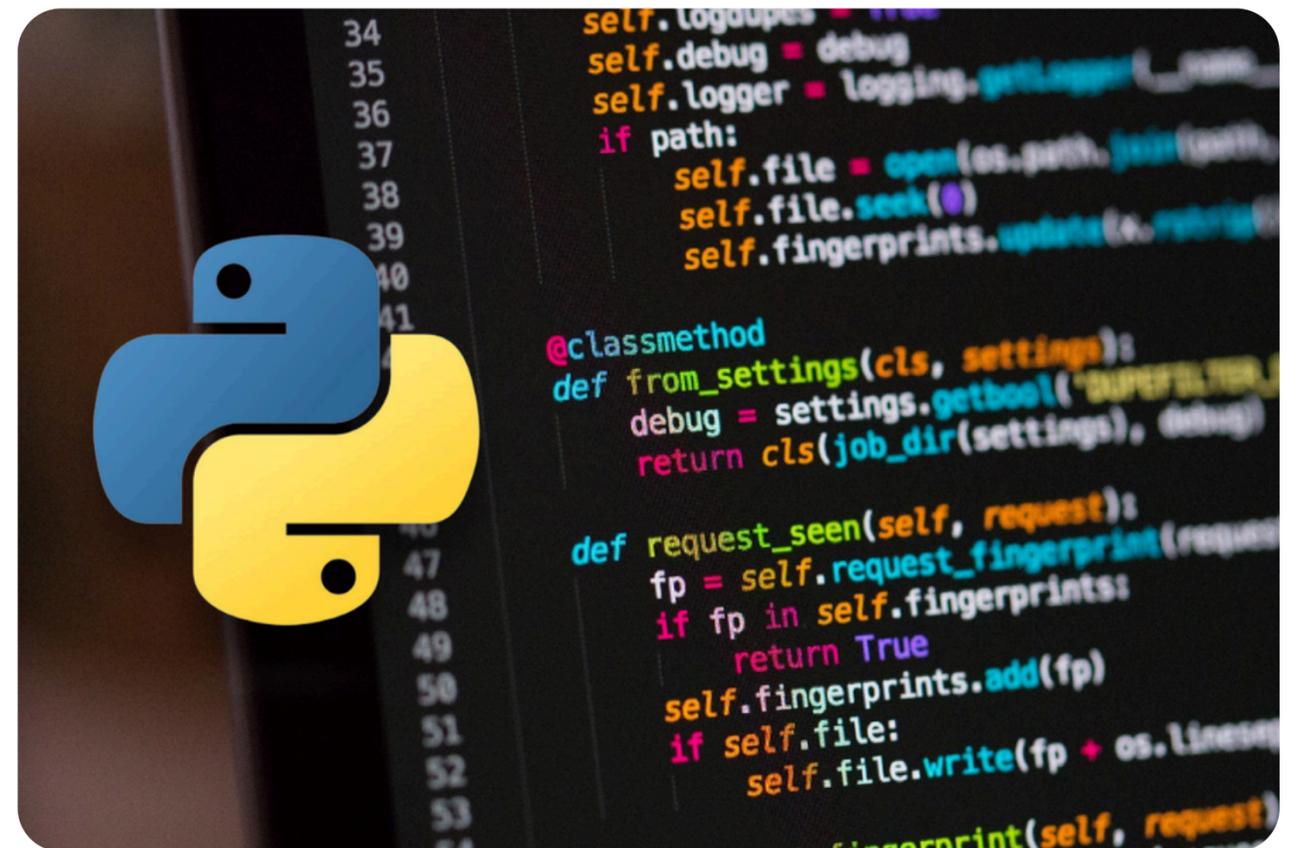
What is Python

- Python เป็นภาษาโปรแกรมที่ถูกสร้างขึ้นโดย Guido van Rossum ในปี ค.ศ. 1989
- Python เป็นภาษาระดับสูง หรือ High-Level Language แบบ Interpreter
- Python ได้รับการเปิดเผยตัวแรกในปี 1991 โดยมีเวอร์ชันแรกที่ถูกเปิดเผยไว้สาธารณะคือ Python 0.9.0 ซึ่งมีการพัฒนาต่อเนื่องจนได้มีการออกเวอร์ชันปัจจุบันที่เรียกว่า Python 3.x โดย Python 2.x ถือเป็นเวอร์ชันหลักในช่วงเวลาหลายปีก่อนที่จะมีการเปลี่ยนแปลง ไปสู่ Python 3 ในปี 2008



What is Python

- Python มีความนิยมอย่างแพร่หลายในวงการโปรแกรมเมอร์ เนื่องจากมีความอ่านง่ายและเข้าใจได้ง่าย มีไวยากรณ์ที่เข้าใจง่าย และมีไลบรารีมากมาย ที่สามารถนำมาใช้งานได้ในงานต่างๆ เช่น
 - เว็บไซต์ (Website)
 - การวิเคราะห์ข้อมูล และงานด้าน AI
 - เครื่องมือที่ช่วยในการพัฒนาซอฟต์แวร์
- Python ยังมีความนิยม เนื่องจากมีชุมชนขนาดใหญ่ที่พร้อมแบ่งปันความรู้เกี่ยวกับ Python มีการพัฒนา และอัปเดตอย่างต่อเนื่องที่ช่วยให้ภาษานี้เติบโตและก้าวหน้าไปอย่างต่อเนื่องตามความต้องการของผู้ใช้งาน



Why Python

Python มีข้อดีมากมายที่ทำให้เป็นภาษาโปรแกรมที่นิยมและถูกนำมาใช้งานอย่างแพร่หลาย

- **1. อ่านและเขียนโปรแกรมได้ง่าย** มีไวยากรณ์ (Syntax) ที่อ่านง่าย และโค้ดมักจะมีที่สวยงามและเข้าใจง่าย เช่น การใช้การวนลูปและโครงสร้างที่ชัดเจน
- **2. มีความหลากหลายในการใช้งาน** Python มีไลบรารีและโมดูลที่หลากหลาย เหมาะสำหรับงานต่างๆ เช่น การพัฒนาเว็บไซต์, การวิเคราะห์ข้อมูล, การเรียนรู้เชิงลึก (Deep Learning), และการประมวลผลภาพ
- **3. ชุมชนและความร่วมมือ** Python มีชุมชนขนาดใหญ่ และพร้อมแบ่งปันความรู้เกี่ยวกับ Python มีการสนับสนุนและแก้ไขปัญหาจากผู้ใช้งานคนอื่น ๆ

- **4. โปรแกรมเชิงวัตถุ (Object-Oriented Programming)** ภาษา Python สนับสนุนการเขียนโปรแกรมเชิงวัตถุได้อย่างสมบูรณ์ ซึ่งช่วยให้การพัฒนาและการบำรุงรักษาโค้ดเป็นไปอย่างมีประสิทธิภาพ
- **5. สามารถนำไปใช้ร่วมกับภาษาอื่นได้ง่าย** Python สามารถใช้งานร่วมกับภาษาโปรแกรมอื่น ๆ ได้อย่างมีประสิทธิภาพ เช่น การใช้งานฐานข้อมูล, การเรียกใช้งานโค้ดภาษาอื่นผ่านหน้า Python
- **6. สนับสนุนในการเรียนรู้และการสอน** Python เป็นภาษาที่น่าสนใจสำหรับผู้เริ่มต้นเนื่องจากภาษา Python เรียนรู้ง่ายและมีทรัพยากรการเรียนรู้ออนไลน์ที่มากมายที่สามารถใช้ในการศึกษาและการพัฒนาทักษะ ได้อย่างมีประสิทธิภาพ โดยเฉพาะ STEM Education

Why Python

แทบจะทุกภาษาโปรแกรมมิ่งมีรูปแบบการเขียนโค้ดที่แตกต่างกันออกไปเล็กน้อย แต่ Python มีลักษณะที่แตกต่างอย่างมากจากภาษาโปรแกรมมิ่งอื่น ดังนี้

- **1. โค้ดอ่านง่าย** Python ออกแบบมาให้เข้าใจง่าย คล้ายกับภาษาอังกฤษ โค้ดสามารถถูก อ่านและเข้าใจได้ง่าย ทำให้เป็นที่นิยมสำหรับผู้เพิ่งเริ่มต้นเรียนรู้การเขียนโปรแกรม
- **2. จบคำสั่งด้วยบรรทัดใหม่** Python ใช้บรรทัดใหม่เพื่อสิ้นสุดคำสั่ง ในขณะที่ภาษาโปรแกรมมิ่งอื่นๆ บางครั้งอาจใช้เครื่องหมายเซมิโคลอน (;) หรือ วงเล็บ () เพื่อแสดงถึงสิ้นสุดของคำสั่ง

- **3. การจัดระดับโดยการเยื้อง (Indentation)** Python ใช้การเยื้อง (indentation) เพื่อกำหนดขอบเขตของโค้ด เช่น ขอบเขตของลูป (loops), ฟังก์ชัน (functions), และคลาส (classes) ในขณะที่ภาษาโปรแกรมมิ่งอื่น อาจใช้วงเล็บปีกกา {} เพื่อจำกัดขอบเขตนั้นๆ

python

Copy code

```
# โปรแกรมหาผลรวมของตัวเลขทั้งหมดในลิสต์
numbers = [1, 2, 3, 4, 5]

# ใช้ลูป for เพื่อวนรอบข้างในลิสต์และหาผลรวม
sum_result = 0
for num in numbers:
    sum_result += num

# พิมพ์ผลรวมออกทางหน้าจอ
print("ผลรวมของตัวเลขทั้งหมดคือ:", sum_result)
```

Who is using Python today?

Python เป็นภาษาโปรแกรมที่ใช้กันอย่างแพร่หลายในหลายสาขางาน โดยบริษัทชั้นนำหลายแห่งเลือกใช้ Python เช่น

- Google,
- YouTube,
- Dropbox,
- Pixar,
- NASA,
- IBM,
- Netflix,
- Yelp,
- Intel,
- Cisco,
- HP,
- Qualcomm, และ JPMorgan Chase

- โดยยังมีการใช้งาน Python ในสายงานต่าง ๆ เช่น การเขียนระบบ, เว็บและการเขียนโปรแกรม API, การพัฒนาแอปพลิเคชัน GUI, เกมและหุ่นยนต์, การสร้างโปรตอทายได้อย่างรวดเร็ว, การผสานระบบ, วิทยาศาสตร์ข้อมูล, แอปพลิเคชัน ฐานข้อมูล, การสื่อสารแบบเรียลไทม์ และอื่น ๆ อีกมากมาย
- มหาวิทยาลัยชั้นนำก็ได้นำ Python เข้ามาใช้เป็นภาษาหลักในรายวิชาด้านวิทยาการคอมพิวเตอร์ด้วย



Python Programming Steps Using Google Colab

1. Go to **<https://colab.research.google.com/>**
2. Log in with your **SSRU gmail**
3. File -> New Notebook
4. Coding with python commands on cell
5. Run cell

1. Go to <https://colab.research.google.com/>

https://colab.research.google.com/

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share Settings Sign in

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Featured examples
- Section

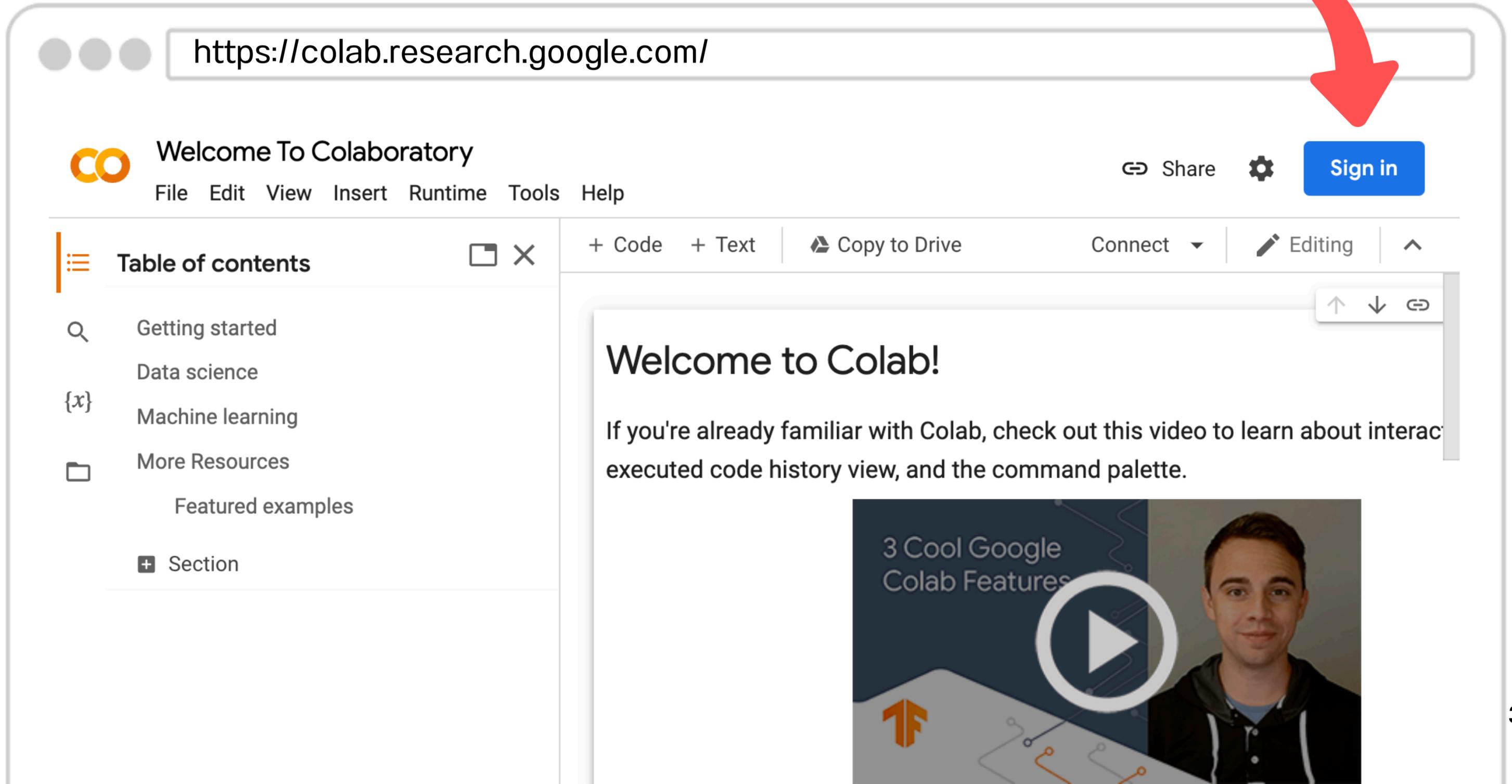
Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interacting with the Colab interface, the executed code history view, and the command palette.

3 Cool Google Colab Features

2. Log in with your SSRU gmail

Click Here



The screenshot shows the Google Colaboratory interface. At the top, the address bar displays `https://colab.research.google.com/`. Below the address bar, the Google Colaboratory logo is on the left, followed by the text "Welcome To Colaboratory". A menu bar contains "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the header, there are "Share" and "Settings" icons, and a prominent blue "Sign in" button. A red arrow points from the text "Click Here" to this "Sign in" button. Below the header, there is a toolbar with options like "+ Code", "+ Text", "Copy to Drive", "Connect", and "Editing". On the left, a "Table of contents" sidebar lists categories like "Getting started", "Data science", "Machine learning", and "More Resources". The main content area displays "Welcome to Colab!" and a paragraph of introductory text. At the bottom, there is a video thumbnail titled "3 Cool Google Colab Features" featuring a man's face and a play button icon.

3. File -> New Notebook

The image shows a browser window at <https://colab.research.google.com/>. The page title is "Welcome To Colaboratory". The "File" menu is open, showing options: "New notebook", "Open notebook", "Upload notebook", "Rename", "Save a copy in Drive", "Save a copy as a GitHub Gist", "Save a copy in GitHub", "Save", "Revision history", "Download", and "Print". A red circle with the number "1" points to the File menu icon in the top left. A red circle with the number "2" points to the "New notebook" option in the File menu. A red arrow points from the "New notebook" option to the "Code" tab in the top right. Below the menu, the text "Welcome to Colab!" is visible, followed by a paragraph: "If you're already familiar with Colab, check out this video to learn about interactive tables, executed code history view, and the command palette." Below the text is a video thumbnail titled "3 Cool Google Colab Features" featuring a man's face and a play button icon.

4. Coding with python commands on cell

The image shows a Google Colab notebook interface. The browser address bar displays `https://colab.research.google.com/`. The notebook title is `chapter1_example1.ipynb`. The top navigation bar includes `File`, `Edit`, `View`, `Insert`, `Runtime`, `Tools`, `Help`, `All changes saved`, `Comment`, `Share`, and a user profile icon. The main workspace shows a code cell with the following Python code:

```
1 print("Hello World!")
```

A red arrow points to the closing parenthesis of the `print` function, with the text **Rigth Here** written in red below it. The code cell also features a play button on the left and a toolbar on the right with icons for undo, redo, link, comment, settings, copy, delete, and a menu.

5. Run cell

The screenshot shows a Google Colab notebook interface. The browser address bar displays `https://colab.research.google.com/`. The notebook title is `chapter1_example1.ipynb`. The top navigation bar includes `File`, `Edit`, `View`, `Insert`, `Runtime`, `Tools`, and `Help`, along with `Comment`, `Share`, and a user profile icon. The `Runtime` section shows `RAM` and `Disk` usage bars, and the `Editing` mode is active. The main workspace contains a code cell with the following content:

```
1 print("Hello World!")
```

The cell has a play button icon on the left, a green checkmark, and a `0s` execution time. Below the code, the output `Hello World!` is displayed, accompanied by a green checkmark icon. A red arrow points to the play button icon, with the text `Click Here` written below it.

Activity: Type the Python command as shown in the picture and observe the results.

https://colab.research.google.com/

chapter1_example1.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[1] ✓ 0s 1 print("Hello World!")

{x} Hello World!

✓ 0s 1 a = "Python"
2 b = "course"
3 c = "is awesome"
4 print(a,b,c)

1 CODE

2 RUN



Syntax - วากยสัมพันธ์ - กฎไวยากรณ์ภาษาคอมพิวเตอร์

- Syntax ในบริบทของการเขียนโปรแกรม หมายถึง กฎและรูปแบบที่ต้องปฏิบัติตามเมื่อเขียนโค้ดในภาษาโปรแกรมนั้น ๆ ซึ่งคล้ายกับการใช้ภาษาที่มีกฎไวยากรณ์ เช่น การใช้ภาษาอังกฤษ "I am a teacher" และเราไม่สามารถใช้ "I is a teacher" ได้เพราะผิดกฎไวยากรณ์ของภาษาอังกฤษ
- ดังนั้น คำว่า "syntax" จึงหมายถึงกฎเกณฑ์ที่ใช้ในการเขียนโค้ด เพื่อให้โปรแกรมทำงานได้อย่างถูกต้อง

- ภาษาโปรแกรมแต่ละภาษามี syntax ที่แตกต่างกันไป เช่น Python, JavaScript, C++, Java จะมี syntax ที่ไม่เหมือนกัน การเขียนโค้ดต้องเป็นไปตาม syntax ของภาษาโปรแกรมนั้น ๆ โดยถ้ามี syntax error หรือ bug (ข้อผิดพลาดในรูปแบบ) โปรแกรมจะไม่สามารถรันหรือทำงานได้อย่างถูกต้อง

```
python
# ตัวอย่าง syntax error
print("Hello world)
```

- ในตัวอย่างนี้มี syntax error เนื่องจากไม่มีเครื่องหมาย " (double quote) ปิดที่สมบูรณ์ ซึ่งจะทำให้โปรแกรมไม่สามารถรันได้ เนื่องจากไม่เป็น syntax ที่ถูกต้องใน Python

Reserved Words or Keywords - คำสงวน หรือ คำสำคัญ

- Reserved Words หรือ คำสงวนในภาษา Python คือ คำที่ถูกสงวนไว้เพื่อใช้ในบางไวยากรณ์ของภาษา Python เท่านั้น โดยคำเหล่านี้มีประมาณ 35 คำ และมีหน้าที่เฉพาะในการควบคุมโครงสร้างและการทำงานของโปรแกรม ซึ่งไม่สามารถใช้เป็นชื่อตัวแปร หรือ ตั้งชื่อฟังก์ชันได้

• and	• del	• if	• pass
• as	• elif	• import	• raise
• assert	• else	• in	• return
• async [note 1]	• except	• is	• True [note 2]
• await [note 1]	• False [note 2]	• lambda	• try
• break	• finally	• None	• while
• class	• for	• nonlocal [note 3]	• with
• continue	• from	• not	• yield
• def	• global	• or	

Reserved Words or Keywords - คำสงวน หรือ คำสำคัญ

คำสงวนที่เป็นค่าคงที่ (Built-in Constants)

False, True, None

คำสงวนที่เป็นตัวดำเนินการทางตรรกะ (Logical Operator)

and, or, not

คำสงวนที่ควบคุมเงื่อนไข (Conditional Control)

if, else, elif

คำสงวนที่ควบคุมการวนทำงานซ้ำ (Iteration Control)

while, for, break, continue

คำสงวนที่เกี่ยวกับการสร้างฟังก์ชันและใช้งานฟังก์ชัน

def, return, lambda

คำสงวนที่เกี่ยวกับการสร้างคลาสและใช้งานคลาส

class, pass

คำสงวนที่ใช้นำเข้าโมดูลและใช้งานโมดูล

import, from, as

คำสงวนที่ใช้จัดการข้อผิดพลาด (Exceptions)

try, except, finally, with

คำสงวนที่เกี่ยวกับการทำงานกับข้อมูล (Data Operation)

in, is

คำสงวนที่เกี่ยวกับขอบเขตการทำงาน

global, nonlocal

คำสงวนที่เกี่ยวกับการลบตัวแปร หรือ ลบอ็อบเจกต์

del

คำสงวนที่เกี่ยวกับการสร้าง Generator Function

yield

Code 1

```
1. # Code 1  
2. print("Hello, World")
```

Hello, World

การเขียนโปรแกรมเพื่อแสดงผลใน Python

- การใช้ print() เป็นการแสดงข้อความ หรือ ค่าใด ๆ ผ่านทางหน้าจอ
- โดยทำการส่งข้อความหรือค่าที่ต้องการแสดงเป็นอาร์กิวเมนต์เข้าไปในฟังก์ชัน print() ซึ่งจะแสดงผลออกทางหน้าจอตามที่ระบุ
- โครงสร้างของฟังก์ชัน print() ใน Python มีรายละเอียดดังนี้:
- *objects เป็น parameter ที่รับค่าที่ต้องการแสดงผลบนหน้าจอ สามารถรับค่าหลาย ๆ ตัวพร้อมกันได้ ซึ่งทำให้ print() สามารถรับค่าจากหลายตัวแปรหรือค่าที่ต่างกันได้
- sep=' ' parameter ที่กำหนดวิธีการคั่นระหว่าง objects ที่ถูกแสดงผล หากไม่ได้กำหนดค่า parameter จะใช้ช่องว่างเป็นค่าเริ่มต้น
- end='\n' parameter ที่กำหนดวิธีการจบบรรทัด หรือการลงบรรทัดใหม่หลังจากที่แสดงผลข้อความ ค่าเริ่มต้น คือ การลงบรรทัดใหม่ (\n)

Syntax:

```
print(*objects, sep=' ', end='\n')
```

Code 2

```
1. # Code 2
2. print('My name is')
3. print('Pasawut')
4. print('Cheerapakorn')
```

```
My name is
Pasawut
Cheerapakorn
```

Code 3

```
1. # Code 3
2. print('My name is', end='')
3. print('Pasawut', end='')
4. print('Cheerapakorn')
```

My name isPasawutCheerapakorn

Code 4

```
1. # Code 4
2. print('My name is', end=' ')
3. print('Pasawut', end=' ')
4. print('Cheerapakorn')
```

My name is Pasawut Cheerapakorn

Code 5

```
1. # Code 5  
2. print('Age', '40', sep=':')
```

Age:40

Practice 1

ให้ปรับปรุงโค้ดนี้ เพื่อแสดงผลตามภาพ

1. `# Practice 1`
2. `print("Student ID", "6622449288")`
3. `print("Student Name", "Mana Ruklean")`
4. `print("email", "manaruklean@gmail.com")`

ตัวอย่างผลลัพธ์

```
Student ID:6622449288  
Student Name:Mana Ruklean  
email manaruklean@gmail.com
```

Practice 1 - Key

```
1. # Practice 1
2. print("Student ID", "6622449288", sep=':')
3. print("Student Name", "Mana Ruklean", sep=':')
4. print("email", "manaruklean@gmail.com", sep=' ')
```

```
Student ID:6622449288
Student Name:Mana Ruklean
email manaruklean@gmail.com
```

Variables- ตัวแปรในภาษา Python

```
x = 10 # สร้างตัวแปรชื่อ x และกำหนดค่าเป็น 10
name = "Alice" # สร้างตัวแปรชื่อ name และกำหนดค่าเป็น "Alice"
my_list = [1, 2, 3] # สร้างตัวแปรชื่อ my_list และกำหนดค่าเป็นรายการข้อมูล [1, 2, 3]
```

ตัวแปร (Variable) ในโปรแกรมคอมพิวเตอร์ คือ

- ชื่อที่ใช้สำหรับอ้างอิงข้อมูลหรือค่าที่เปลี่ยนแปลงได้ในหน่วยความจำ
- ตัวแปรเปรียบเสมือนช่องที่เก็บข้อมูลที่สามารถเปลี่ยนแปลงค่าได้ตลอดระยะเวลาที่โปรแกรมทำงาน
- ตัวแปรมักจะถูกใช้เพื่อเก็บข้อมูลที่ต้องการนำมาใช้ในโปรแกรม เช่น ตัวเลข, ข้อความ, ค่าทางคณิตศาสตร์, รายการข้อมูล เป็นต้น

Variables- ตัวแปรในภาษา Python

การตั้งชื่อตัวแปรใน Python ต้องปฏิบัติตามกฎและหลักเกณฑ์ที่กำหนดไว้ดังนี้

- 1. ชื่อตัวแปรจะต้องขึ้นต้นด้วยตัวอักษรหรือเครื่องหมาย underscore (_)** ตัวแปรต้องขึ้นต้นด้วยตัวอักษร (a-z, A-Z) หรือ เครื่องหมาย underscore (_) เท่านั้น ไม่สามารถขึ้นต้นด้วยตัวเลขได้
- 2. ไม่สามารถขึ้นต้นด้วยตัวเลข** ชื่อตัวแปรไม่สามารถขึ้นต้นด้วยตัวเลขได้ แต่สามารถประกอบไปด้วยตัวอักษรหรือ ตัวเลขหลัง ตัวอักษรแรกได้
- 3. ใช้ตัวอักษร ตัวเลข และเครื่องหมาย underscore เท่านั้น** ชื่อตัวแปรสามารถประกอบไปด้วยตัวอักษร (a-z, A-Z), ตัวเลข (0-9), และเครื่องหมาย underscore (_) เท่านั้น ไม่สามารถใช้เครื่องหมายอื่น ๆ เช่น @, \$, % ได้
- 4. ชื่อตัวแปรเป็นตัวพิมพ์เล็กและตัวพิมพ์ใหญ่แตกต่างกัน** ตัวแปรใน Python เป็น case-sensitive ซึ่งหมายความว่า age, Age, และ AGE จะถือเป็นตัวแปรที่แตกต่างกัน
- 5. ชื่อตัวแปรไม่สามารถเป็น keyword ของ Python** ชื่อตัวแปรไม่สามารถเป็นคำสงวน (keyword) ของ Python เช่น if, else, for, while เป็นต้น

Variables- ตัวแปรในภาษา Python

ตัวอย่างการตั้งชื่อตัวแปรที่ถูกต้อง

```
age
```

```
user_age
```

```
totalAmount
```

```
my_list
```

```
first_name
```

```
last_name
```

```
username2
```

```
_variable
```

```
user_123
```

```
carColor
```

Variables- ตัวแปรในภาษา Python

ตัวอย่างการตั้งชื่อตัวแปรที่ผิด

3apples (ขึ้นต้นด้วยตัวเลข)

my-variable (มีเครื่องหมายขีด -)

@username (มีเครื่องหมายพิเศษ @)

if (เป็น keyword ของ Python)

my variable (มีช่องว่าง)

my+var (มีเครื่องหมายพิเศษ +)

5number (ขึ้นต้นด้วยตัวเลข)

my.variable (มีเครื่องหมายจุด .)

variable-name! (มีเครื่องหมายพิเศษ !)

break (เป็น keyword ของ Python)

Variables- ตัวแปรในภาษา Python

แนวทางการตั้งชื่อตัวแปร

1. Camel Case แบบหลังอูฐ

```
myFirstName = 'Nutthapat'
```



2. Pascal Case แบบคำ

```
MyFirstName = 'Nutthapat'
```

XxxYyyZzz

3. Snake Case แบบยาว

```
my_first_name = 'Nutthapat'
```



Code 6

```
1. # Code 6
2. number1 = 10
3. number2 = 40
4. print(number2)
```

40

Code 7

```
1. # Code 7
2. number1 = 10
3. number2 = 40
4. number3 = 50
5. result = number1+number2+number3
6. print(result)
```

100

Variables - Assign Multiple Values - การกำหนดค่าแบบหลายค่า

1. One Value to Multiple Variables การกำหนดค่าแบบค่าเดียวให้หลายตัวแปร
ตัวอย่าง

```
fruit1 = fruit2 = fruit3 = 'Mango'  
print(fruit1, fruit2, fruit3)
```

2. Many Values to Multiple Variables การกำหนดค่าแบบหลายค่าให้หลายตัวแปร
ตัวอย่าง

```
fruit1, fruit2, fruit3 = 'Mango', 'Banana', 'Orange'  
print(fruit1, fruit2, fruit3)
```

3. Unpack a Collection การกำหนดค่าแบบกระจายค่าจากลิสต์
ตัวอย่าง

```
fruits = ['Mango', 'Banana', 'Orange']  
fruit1, fruit2, fruit3 = fruits  
print(fruit1, fruit2, fruit3)
```

Code 8

```
1. # Code 8
2. # One Value to Multiple Variables
3. fruit1 = fruit2 = fruit3 = 'Mango'
4. print(fruit1, fruit2, fruit3)
5. # Many Values to Multiple Variables
6. fruit1, fruit2, fruit3 = 'Mango', 'Banana', 'Orange'
7. print(fruit1, fruit2, fruit3)
8. # Unpack a Collection
9. fruits = ['Mango', 'Banana', 'Orange']
10. fruit1, fruit2, fruit3 = fruits
11. print(fruit1, fruit2, fruit3)
```

```
Mango Mango Mango
Mango Banana Orange
Mango Banana Orange
```

Practice 2

ให้ทำการ Coding เพื่อตั้งชื่อตัวแปร 3 ตัว คือ

stu_name สำหรับเก็บชื่อนักเรียน

score_test1 สำหรับเก็บคะแนนการทดสอบที่ 1 (40 คะแนน)

score_test2 สำหรับเก็บคะแนนการทดสอบที่ 2 (30 คะแนน)

score_test3 สำหรับเก็บคะแนนการทดสอบที่ 3 (10 คะแนน)

score สำหรับรวมคะแนนการทดสอบที่ 1 ถึง 3

จากนั้นให้ทำการแสดงผลดังภาพ

Pasawut Score: 80

Practice 2 - Key

```
1. # Practice 2
2. name = "Nutthapat"
3. score_test1 = 40
4. score_test2 = 30
5. score_test3 = 10
6. score =
score_test1+score_test2+score_test3
7. print(name, 'Score:', score, sep=' ')
```

Pasawat Score: 80

Numbers - จำนวน

1. **จำนวนเต็ม (Integers)** เป็นจำนวนที่ค่านันๆ ไม่มีทศนิยม

เช่น -5, 0, 100

2. **จำนวนทศนิยม (Floating-point numbers)** เป็นจำนวนที่ค่านันๆ มีทศนิยม

เช่น 3.14, -0.001, 2.0

```
1. # integer
```

```
2. x = 10
```

```
3. y = -5
```

```
4. z = 0
```

```
5. # float
```

```
6. pi = 3.14159
```

```
7. temperature = 25.5
```

Code 9

```
1. # integer
2. score1 = 10
3. print(score1, type(score1))
4. # float
5. score2 = 10.9
6. print(score2, type(score2))
```

```
10 <class 'int'>
10.9 <class 'float'>
```

ฟังก์ชัน type() เป็นฟังก์ชันที่ใช้ตรวจสอบประเภท (type) ของวัตถุหรือค่าที่รับเข้ามาในพารามิเตอร์ของฟังก์ชันนี้ ฟังก์ชัน type() จะคืนค่าประเภทของวัตถุนั้นออกมาในรูปแบบของ object type หรือ ชนิดของข้อมูลที่ได้รับ

<class 'int'> หมายถึง ประเภทข้อมูลนี้จัดกลุ่มเป็น integer หรือ ค่าจำนวนเต็ม
<class 'float'> หมายถึง ประเภทข้อมูลนี้จัดกลุ่ม เป็น float หรือ ค่าจำนวนทศนิยม

Code 10

```
1. # Code 10
2. score1 = '10'
3. score2 = 10
4. print(score1, type(score1))
5. print(score2, type(score2))
```

```
10 <class 'str'>
10 <class 'int'>
```

<class 'str'> หมายถึง ประเภทข้อมูลนี้จัดกลุ่มเป็น string
หรือ สายอักขระ หรือ ข้อความ

<class 'int'> หมายถึง ประเภทข้อมูลนี้จัดกลุ่มเป็น integer
หรือ ค่าจำนวนเต็ม

Casting - การแปลงประเภทข้อมูล

- **Casting หรือ Type Casting** หมายถึง การเปลี่ยนแปลงประเภทของข้อมูลจากประเภทหนึ่งไปยังประเภทอื่น ๆ โดยการ casting นั้นทำให้เราสามารถแปลงประเภทค่าข้อมูลจากประเภทหนึ่งไปยังประเภทอื่นที่เหมาะสมในการใช้งานได้
- ในภาษา Python การ casting ทำได้โดยใช้ฟังก์ชันที่เกี่ยวข้องกับประเภทข้อมูล เช่น **int(), float(), str()**

```
x = "10"  
y = int(x)      # แปลงจากข้อความเป็นจำนวนเต็ม  
print(y)       # Output: 10
```

```
a = 5  
b = float(a)   # แปลงจากจำนวนเต็มเป็นทศนิยม  
print(b)       # Output: 5.0
```

```
m = 3.14  
n = str(m)     # แปลงจากทศนิยมเป็นข้อความ  
print(n)       # Output: '3.14'
```

```
j = 100.99  
k = int(j)     # แปลงจากทศนิยมเป็นจำนวนเต็ม  
print(k)       # Output: 100
```

Code 11

```
1. # Code 11
2. x = "10"
3. y = int(x) # แปลงจากข้อความเป็นจำนวนเต็ม
4. print(x, 'is', type(x), y, 'is', type(y))
5.
6. a = 5
7. b = float(a) # แปลงจากจำนวนเต็มเป็นทศนิยม
8. print(a, 'is', type(a), b, 'is', type(b))
9.
10. m = 3.14
11. n = str(m) # แปลงจากทศนิยมเป็นข้อความ
12. print(m, 'is', type(m), n, 'is', type(n))
13.
14. j = 100.99
15. k = int(j) # แปลงจากทศนิยมเป็นจำนวนเต็ม
16. print(j, 'is', type(j), k, 'is', type(k))
```

```
10 is <class 'str'> 10 is <class 'int'>
5 is <class 'int'> 5.0 is <class 'float'>
3.14 is <class 'float'> 3.14 is <class 'str'>
100.99 is <class 'float'> 100 is <class 'int'>
```



Practice 3

เขียนโปรแกรมตรวจสอบประเภทข้อมูล และแปลงค่าข้อมูล ต่อไปนี้

1. ให้ตั้งตัวแปร data1 = '50' และแสดงผล data1 จากนั้นตรวจสอบว่า data1 เป็นประเภทอะไร?
2. ให้แปลง data1 เป็นประเภทข้อมูล integer และแสดงผล data1 จากนั้นตรวจสอบว่า data1 เป็นประเภทอะไร
3. ให้แปลง data1 เป็นประเภทข้อมูล float และแสดงผล data1 จากนั้นตรวจสอบว่า data1 เป็นประเภทอะไร
4. ให้แปลง data1 เป็นประเภทข้อมูล string และแสดงผล data1 จากนั้นตรวจสอบว่า data1 เป็นประเภทอะไร

ตัวอย่างผลลัพธ์

```
50 is <class 'str'>  
50 is <class 'int'>  
50.0 is <class 'float'>  
50.0 is <class 'str'>
```

Practice 3 - Key

```
1. # Practice 3
2. data1 = '50'
3. print(data1, 'is', type(data1))
4.
5. data1 = int(data1)
6. print(data1, 'is', type(data1))
7.
8. data1 = float(data1)
9. print(data1, 'is', type(data1))
10.
11. data1 = str(data1)
12. print(data1, 'is', type(data1))
```

```
50 is <class 'str'>
50 is <class 'int'>
50.0 is <class 'float'>
50.0 is <class 'str'>
```

Boolean - True or False?

ค่าบูลีน (Boolean) เป็นประเภทของข้อมูลที่มีค่าเพียงสองค่าเท่านั้น คือ

- True แทนค่าที่เป็นจริง และ
- False แทนค่าที่เป็นเท็จ

ค่าบูลีนสามารถใช้ในการควบคุมการทำงานของโปรแกรมด้วยเงื่อนไข เช่น ในการตรวจสอบการเปรียบเทียบข้อมูล การควบคุม การทำงานของลูป หรือ เงื่อนไขในการทำงานของโค้ด

```
1. student1 = 50
2. student2 = 100
3. print(student1 > student2)
```

False

```
1. student1 = 50
2. student2 = 100
3. print(student1 < student2)
```

True

Code 12

```
1. student1 = 50
2. student2 = 100
3. print(student1 > student2)
4. print(student1 < student2)
```

False
True

Practice 4

1. สร้างตัวแปรชื่อ password
2. กำหนดค่าให้ตัวแปร password มีค่าเท่ากับ "90009"
3. ตรวจสอบว่า password มีค่าเท่ากับ "00000" และแสดงผลจริง หรือ เท็จ
4. ตรวจสอบว่า password มีค่าเท่ากับ "99999" และแสดงผลจริง หรือ เท็จ
5. ตรวจสอบว่า password มีค่าเท่ากับ "90009" และแสดงผลจริง หรือ เท็จ

ตัวอย่างผลลัพธ์

```
PASSWORD 00000 False  
PASSWORD 99999 False  
PASSWORD 90009 True
```

Practice 4 - Key

```
1. # Practice 4
2. password = "90009"
3. print("PASSWORD 00000", password == "00000")
4. print("PASSWORD 99999", password == "99999")
5. print("PASSWORD 90009", password == "90009")
```

```
PASSWORD 00000 False
PASSWORD 99999 False
PASSWORD 90009 True
```

Input() - รับค่าข้อมูลผ่านทาง Keyboard

ฟังก์ชัน input() ใช้สำหรับการรับข้อมูลจากผู้ใช้ โดยรับค่าจากการป้อนผ่านคีย์บอร์ด (Keyboard) โดยสามารถแสดงข้อความให้ผู้ใช้เห็นเพื่อแจ้งว่าให้ทำการป้อนค่าอะไร และจะรอรับข้อมูลจากผู้ใช้จนกว่าผู้ใช้จะป้อนข้อมูลและกด Enter เพื่อส่งข้อมูลเข้าสู่โปรแกรม

```
1. #Code 13
2. name = input("Please enter your name: ")
3. print("Hello,", name)
```

```
Please enter your name: Pasawut
Hello, Pasawut
```

Code 13

```
1. #Code 13
2. name = input("Please enter your name: ")
3. print("Hello,", name)
```

```
Please enter your name: 
Hello, Pasawut
```

Practice 5

ให้เขียนโปรแกรม

1. รับค่าชื่อจริง เก็บไว้ในตัวแปร firstname
2. รับค่านามสกุลเก็บไว้ในตัวแปร lastname
3. รับค่าปี ค.ศ. เกิด เก็บไว้ในตัวแปร y_birth
4. จากนั้นให้ทำการแสดงผล ชื่อและนามสกุล และอายุ (นำ ค.ศ.ปัจจุบันลบกับ ค.ศ.เกิด) ออกทางหน้าจอ

ตัวอย่างผลลัพธ์

```
Please enter Firstname: Pasawut
Please enter Lastname: Cheerapakorn
Please enter Birth Year: 1993
Pasawut Cheerapakorn 30
```

Practice 5 - Error

ให้เขียนโปรแกรม

1. รับค่าชื่อจริง เก็บไว้ในตัวแปร firstname
2. รับค่านามสกุลเก็บไว้ในตัวแปร lastname
3. รับค่าปี ค.ศ. เกิด เก็บไว้ในตัวแปร y_birth
4. จากนั้นให้ทำการแสดงผล ชื่อและนามสกุล และอายุ (นำ ค.ศ.ปัจจุบันลบกับ ค.ศ.เกิด) ออกทางหน้าจอ

ตัวอย่างผลลัพธ์

```
Please enter Firstname: Pasawut
Please enter Lastname: Cheerapakorn
Please enter Birth Year: 1993
Pasawut Cheerapakorn 30
```

Practice 5 - Error

```
Please enter Firstname: Pasawut
Please enter Lastname: Cheerapakorn
Please enter Birth Year: 1993

-----
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-1474987834.py in <cell line: 0>()
      2 lastname = input("Please enter Lastname: ")
      3 y_birth = input("Please enter Birth Year: ")
----> 4 print(firstname, lastname, 2023 - y_birth)

TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

Error ที่เกิดขึ้นเป็น TypeError ซึ่งบอกว่ามีประเภทข้อมูลไม่สามารถทำการดำเนินการบวกลบได้ระหว่างตัวแปรข้อมูลประเภทต่าง ๆ ในกรณีนี้ โค้ดพยายามที่จะคำนวณอายุโดยการลบปีปัจจุบันกับปีที่เกิด แต่มี ปัญหาเกิดขึ้นกับข้อมูลที่ได้รับจากผู้ที่ใช้ที่ป้อนผ่านฟังก์ชัน input() เนื่องจาก input() จะรับข้อมูลเป็นข้อความ (String) ไม่ใช่ตัวเลข (Integer)

โดยการที่เราพยายามที่จะทำการลบค่าปีที่ป้อนเข้ามา (string) กับปีปัจจุบัน (integer) จึงเกิด Error ขึ้น เนื่องจากไม่สามารถทำการดำเนินการลบได้ระหว่าง String และ Integer โดยตรงสำหรับการแก้ปัญหานี้ ให้ทำการแปลงปีที่ป้อนเข้ามาจาก String เป็น Integer ก่อนที่จะทำการคำนวณอายุได้โดยใช้ int()

Practice 5 - Key

```
1. # Practice 5
2. firstname = input("Please enter Firstname: ")
3. lastname = input("Please enter Lastname: ")
4. y_birth = int(input("Please enter Birth Year: "))
5. print(firstname, lastname, 2025 - y_birth)
```

```
Please enter Firstname: Pasawut
Please enter Lastname: Cheerapakorn
Please enter Birth Year: 1993
Pasawut Cheerapakorn 30
```

Operand and Operators

Operand หรือ ตัวถูกดำเนินการ คือ ค่าหรือตัวแปรที่ถูกนำมาใช้กับตัวดำเนินการ เพื่อทำให้เกิดการดำเนินการทางคณิตศาสตร์ การเปรียบเทียบ หรือการดำเนินการต่าง ๆ ตามที่ต้องการ

ตัวอย่าง Operand

- ค่าจำนวน (value) เช่น 5, 3.14
- ตัวแปร (variable) เช่น x, y, result

Operand and Operators

Arithmetic Operators หรือ ตัวดำเนินการทางคณิตศาสตร์

+ (บวก): บวกจำนวน

- (ลบ): ลบจำนวน

* (คูณ): คูณจำนวน

/ (หาร): หารจำนวน

% (modulo): หาเศษจากการหาร

** (ยกกำลัง): ยกกำลังจำนวน

```
1. x = 10
2. y = 3
3. print(x + y) # Output: 13
4. print(x - y) # Output: 7
5. print(x * y) # Output: 30
6. print(x / y) # Output: 3.3333333333333335
7. print(x % y) # Output: 1
8. print(x ** y) # Output: 1000
```

Operand and Operators

Assignment Operators หรือ ตัวดำเนินการกำหนดค่า

= (กำหนดค่า): กำหนดค่าให้ตัวแปร

+=, -=, *=, /= เป็นแบบย่อของการดำเนินการทางคณิตศาสตร์พร้อมกับการกำหนดค่า

```
1. x = 5
2. x += 3      # เทียบเท่ากับ x = x + 3
3. print(x)    # Output: 8
```

Operand and Operators

Logical Operators หรือ ตัวดำเนินการตรรกศาสตร์

and (และ): ทำการควบคุมเงื่อนไขที่ทั้งสองข้างเป็นจริง

or (หรือ): ทำการควบคุมเงื่อนไขอย่างน้อยหนึ่งข้างเป็นจริง

not (ไม่): ใช้เพื่อกลับค่าของเงื่อนไข

```
1.     x = True
2.     y = False
3.     print(x and y)    # Output: False
4.     print(x or y)    # Output: True
5.     print(not x)     # Output: False
```

Code 14

```
1. # Code 14
2. height = float(input("Please enter your height (cm.): "))
3. height = height/100
4. print("Height: ", height, "m.")
```

```
Please enter your height (cm.): 199
Height: 1.99 m.
```

Practice 6

ให้เขียนโปรแกรมแปลงอุณหภูมิจาก Celsius เป็น Fahrenheit ตามอัลกอริทึมนี้

1. สร้างตัวแปรชื่อ temperature สำหรับรับค่าอุณหภูมิเป็น Celsius

2. แปลงอุณหภูมิจาก Celsius เป็น Fahrenheit ด้วยสูตร

$$\text{fahrenheit} = (\text{celsius} * 9/5) + 32$$

3. แสดงผลลัพธ์ดังภาพ

```
Temperature (Celsius): 37
Fahrenheit: 98.6
```

Practice 6 - Key

```
1. # Practice 6
2. temperature = float(input("Temperature (Celsius): "))
3. temperature = (temperature * 9/5) + 32
4. print("Fahrenheit:", temperature)
```

```
Temperature (Celsius): 37
Fahrenheit: 98.6
```

Assignment 7.1 - มอบหมายงานครั้งที่ 7.1

ให้เขียนโปรแกรมภาษา Python ใน Google Colab เพื่อแปลงค่าเงินบาทเป็นค่าเงินสกุลต่างๆ โดยมีอัลกอริทึมดังนี้

1. รับค่าเงินสกุลบาท ทางหน้าจอ เก็บค่าไว้ในตัวแปร `thai_baht`
2. แสดงค่าเงินสกุลบาทออกทางหน้าจอ
3. แปลงค่าเงินสกุลบาทเป็นสกุลเงินเก็บลงตัวแปร `usd_dollars`, `jpy_yen`, `eur_euro`
4. แสดงค่าเงินสกุลบาทออกทางหน้าจอ
5. แสดงค่าเงินสกุล `usd` ออกทางหน้าจอ
6. แสดงค่าเงินสกุล `jpy` ออกทางหน้าจอ
7. แสดงค่าเงินสกุล `eur` ออกทางหน้าจอ
8. ตรวจสอบค่าเงินจาก Google เพื่อเทียบเคียงความถูกต้อง

Post Test

Question:

1. ต้องการแสดงผลคำว่า Hello, World ต้องเติมคำสั่งใด

A Print("Hello, World")

B Print(Hello, World)

C print("Hello, World")

D print(Hello, World)

E ถูกทุกข้อ

F ไม่มีข้อถูก

Question:

2. ខ្មែរដែលប្រើ Reserved Word

A True

B None

C break

D class

E Control

Question:

3. ข้อใดให้ผลลัพธ์ Date:11/DEC/2023

A

```
print('Date:')  
print('11', 'DEC', '2023', sep='/')
```

C

```
print('Date:')  
print('11/', 'DEC/', '2023')
```

B

```
print('Date:', end='/n')  
print('11/ 'DEC/'2023', sep='/')
```

D

```
print('Date:', end="")  
print('11', 'DEC', '2023', sep='/')
```

Question:

4. ข้อใด คือ การตั้งชื่อตัวแปรแบบ Pascal Case

A My_Product_Price

D my_product_price

B MyProductPrice

E MYPRODUCTPRICE

C myProductPrice

Question:

5. จาก Code แสดงผลตามข้อใด

```
1. data1 = "50"  
2. data1 = float(data1)  
3. print(data1, 'is', type(data1))
```

A 50.0 is <class 'float'>

C "50" is <class 'str'>

B 50 is <class 'float'>

D "50.0" is <class 'float'>

Question:

6. จาก Code ให้อินให้ค่า Boolean เป็น True

```
1. student1 = 50  
2. student2 = 100
```

A print(student1>student2)

C print(student1<student2)

B print(student1=student2)

D print(student1!student2)

Question:

7. จาก Code ประเภทข้อมูลที่รับเข้ามาเป็นประเภทใด

```
1. year = input("Birth Year:")  
Birth Year: 1983
```

A int

C string

B float

D real

Question:

8. តើពាក្យគន្លឹះ Keywords ណាខ្លះ ដែល ប្រើ ក្នុង Conditional Statement ក្នុងភាសា Python

A if

D if else

B if elif

E if elif else

C else

F if elseif else

Question:

9. จากคำสั่ง ไม่สามารถ แสดงผลข้อใด

```
import random  
num = random.randrange(5, 10)  
print(num)
```

A 5

C 10

E 8

B 7

D 9

F 6

Question:

10. จากคำสั่ง ไม่สามารถ แสดงผลข้อใด

```
1. count = 1
2. while count<=5:
3.     print(count)
4.     count+=1
```

A 1

B 2

C 3

D 4

E 5

F 6

Computational Science

Thank You

DTI1306 Computational Science

Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

Content Credit By: Asst.Prof.Nutthapat Kaewrattanapat, PhD.



Pasawut Cheerapakorn

Suan Sunandha Rajabhat University