

CS

พื้นฐานการโปรแกรมภาษาไพทอนเบื้องต้นด้วย Colab

# บทที่ 7

ตอนที่ 1 เครื่องมือสำหรับการเขียนโปรแกรมภาษาไพทอน  
และหลักการโปรแกรมภาษาไพทอนเบื้องต้น



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร์  
Asst.Prof.Dr.Nutthapat Kaewrattanapat  
Suan Sunandha Rajabhat University

รายวิชาวิทยาการคำนวณ (Computational Science) 3(2-2-5) หน่วยกิต  
บรรยายนักศึกษาสาขาวิชาเทคโนโลยีดิจิทัลเพื่อการศึกษา ภาคเรียนที่ 2 ปีการศึกษา 2566  
ห้องบรรยาย 1121 คณะครุศาสตร์ มหาวิทยาลัยราชภัฏสวนสุนันทา



# อาจารย์บรรยาย



## ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

### การศึกษา

- 2565 ปริญญาเอก ปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตร์มหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

### คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขารุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCN-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตร์ข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชซิกแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

# Course Description

## DTI1306 วิทยาการคำนวณ (Computational Science)

3(2-2-5) บรรยาย 2 ชม ปฏิบัติ 2 ชม ศึกษาด้วยตนเอง 5 ชม

วิเคราะห์ เทคนิค วิธีการขั้นตอนการแก้ปัญหา ทักษะการคิดเชิงคำนวณ เชิงนามธรรม ฝึกทักษะในการแก้ปัญหาโดยใช้ขั้นตอนการแก้ปัญหา การย่อยปัญหา การแสดงขั้นตอน การแก้ปัญหา โดยการเขียน บอกเล่า วาดภาพ หรือใช้สัญลักษณ์ ออกแบบและเขียนโปรแกรม โดยใช้ซอฟต์แวร์หรืออุปกรณ์ 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.

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

System Theory

Computational Thinking

Decomposition

Abstraction

Pattern Recognition

Algorithm Design

Design Thinking

Standard of Flowchart Design

Flowgorithm

Computer Programming Language

# Measurement and Evaluation

## การวัดและประเมินผล

### 1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%  
24%  
12%  
4%

### 2. การสอบกลางภาค (Midterm Examination)

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

20%

### 3. โครงการประจำภาคเรียน (Term Project)

- บทความ และการนำเสนอ

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                  |

# การมีส่วนร่วมในชั้นเรียน - ส่ง Lecture Note

ครั้งที่ 1 วันพุธที่ 6 ธันวาคม 2566

ชื่อ-สกุล:  
อีเมล:

รหัสนักศึกษา:

## Pretest

- |    |     |
|----|-----|
| 1. | 6.  |
| 2. | 7.  |
| 3. | 8.  |
| 4. | 9.  |
| 5. | 10. |

## Post-test

- |    |     |
|----|-----|
| 1. | 6.  |
| 2. | 7.  |
| 3. | 8.  |
| 4. | 9.  |
| 5. | 10. |

สรุปเนื้อหาบรรยาย

# Course Outline

- บทที่ 1 – พื้นฐานทางวิทยาการคำนวณ
- บทที่ 2 – พื้นฐานทางด้านเทคโนโลยีดิจิทัล
- บทที่ 3 – พื้นฐานทางด้านการรู้เท่าทันสื่อและดิจิทัล
- บทที่ 4 – พื้นฐานการวิเคราะห์และออกแบบอัลกอริทึม
- บทที่ 5 – พื้นฐานการเขียนโปรแกรมด้วยบล็อกคำสั่ง (Scratch)
- บทที่ 6 – พื้นฐานการโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้นด้วย Microbit
- บทที่ 7 – พื้นฐานการโปรแกรมภาษาไพทอนเบื้องต้นด้วย Colab
- บทที่ 8 – การพัฒนาโครงงานทางเทคโนโลยีดิจิทัลเพื่อการศึกษา

# Measurement and Evaluation

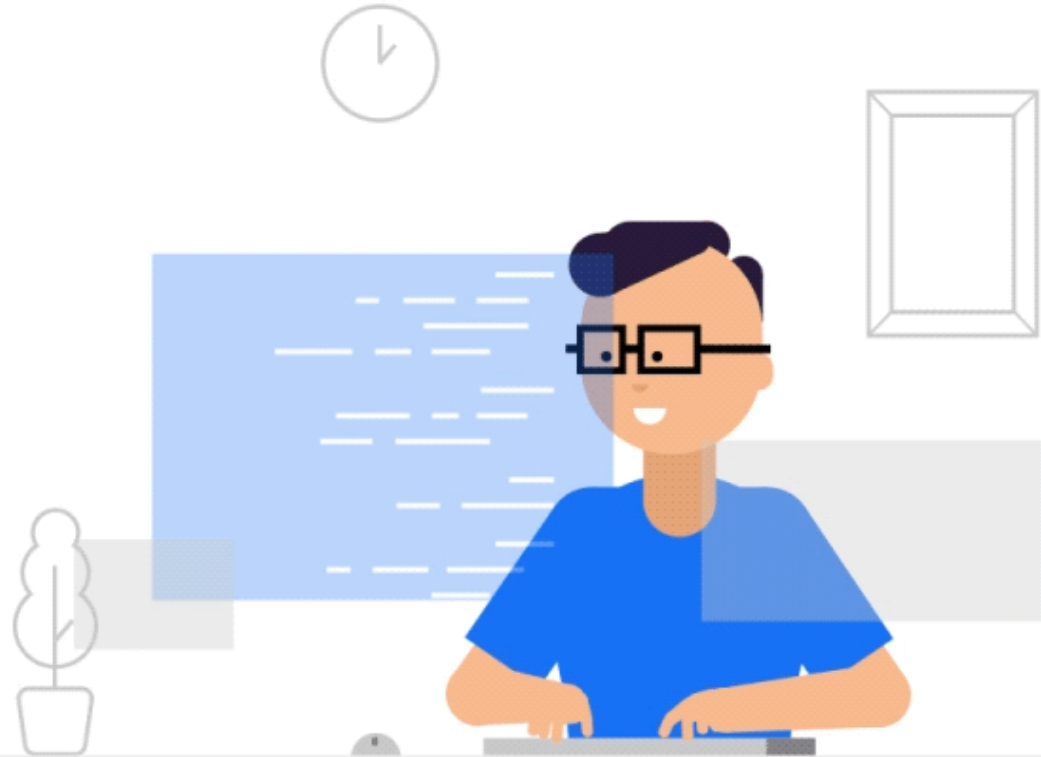
| ครั้งที่/วันที่             | บทเรียน/หัวข้อ   | มอบหมายงาน (24%)            | สอบ Post-test (12%)  | การมีส่วนร่วมในชั้นเรียน (4%) |
|-----------------------------|--|-----------------------------|----------------------|-------------------------------|
| 1 พุท 6 ธค 66<br>Onsite     | แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 พื้นฐานทางวิทยาการคำนวณ  | -                           | -                    | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 2 พุท 13 ธค 66<br>Onsite    | บทที่ 2 พื้นฐานทางด้านเทคโนโลยีดิจิทัล   | 2 คะแนน<br>Minute Paper     | 2 คะแนน              | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 3 พุท 20 ธค 66<br>Onsite    | บทที่ 3 พื้นฐานทางด้านความรู้เท่าทันสื่อและดิจิทัล   | 2 คะแนน<br>Certificate TPQI | 1 คะแนน              | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 4 พุท 27 ธค 66<br>Hybrid    | บทที่ 4 พื้นฐานการวิเคราะห์และออกแบบอัลกอริทึม ตอนที่ 1 (Flowchart, Flowgorithm และโครงสร้างการควบคุมโปรแกรมแบบเรียงลำดับและโครงสร้างการควบคุมโปรแกรมแบบตัดสั้นใจ) | 3 คะแนน<br>โปรแกรม bmi      | 1 คะแนน<br>สอบในระบบ | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 5 พุท 3 มค 67<br>Onsite     | บทที่ 4 พื้นฐานการวิเคราะห์และออกแบบอัลกอริทึม ตอนที่ 2 (โครงสร้างการควบคุมโปรแกรมแบบทำซ้ำ)  | 3 คะแนน                     | 1 คะแนน              | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 6 พุท 10 มค 67<br>Onsite    | บทที่ 5 พื้นฐานการเขียนโปรแกรมด้วยบล็อกคำสั่ง ตอนที่ 1 (Block-based Programming ด้วย Scratch)  | 2 คะแนน                     | 1 คะแนน              | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 7 พุท 17 มค 67<br>On-Demand | บทที่ 5 พื้นฐานการเขียนโปรแกรมด้วยบล็อกคำสั่ง ตอนที่ 2 (Block-based Programming และการประยุกต์)  | 2 คะแนน                     | 1 คะแนน<br>สอบในระบบ | เช็คชื่อในระบบ                |
| 8 พุท 24 มค 67              | สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%   |                             |                      |                               |

# Measurement and Evaluation

| ครั้งที่/วันที่                  | บทเรียน/หัวข้อ  | มอบหมายงาน (24%) | สอบ Post-test (12%) | การมีส่วนร่วมในชั้นเรียน (4%) |
|----------------------------------|---|------------------|---------------------|-------------------------------|
| 9 พุท 7 กพ 67<br><b>Onsite</b>   | บทที่ 6 การโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้นด้วย Microbit                                       | 2 คะแนน          | 1 คะแนน             | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 10 พุท 14 กพ 67<br><b>Onsite</b> | บทที่ 7 การโปรแกรมภาษาไพทอนเบื้องต้นด้วย Colab (Basic Input/Output, Variables, Operation)       | 2 คะแนน          | 1 คะแนน             | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 11 พุท 21 กพ 67<br><b>Onsite</b> | บทที่ 7 การโปรแกรมภาษาไพทอนเบื้องต้นด้วย Colab (Decision and Iteration Statement)               | 2 คะแนน          | 1 คะแนน             | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 12 พุท 28 กพ 67<br><b>Online</b> | บทที่ 8 การพัฒนาโครงการทางเทคโนโลยีดิจิทัลเพื่อการศึกษา (การคิดเชิงออกแบบ, กระบวนการทางโครงการ) | 2 คะแนน          | 1 คะแนน             | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 13 พุท 6 มีค 67<br><b>Online</b> | บทที่ 8 การพัฒนาโครงการทางเทคโนโลยีดิจิทัลเพื่อการศึกษา (ปฏิบัติ)                               | 2 คะแนน          | 1 คะแนน             | ขาด/ไม่ทันเช็คชื่อ<br>-1%     |
| 14 พุท 13 มีค 67                 | สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%                                |                  |                     |                               |
| 15 พุท 20 มีค 67                 | ส่งบทความ (โครงการ) และนำเสนอ 20%<br>แผนการจัดการเรียนรู้และสื่อการเรียนรู้ด้านวิทยาการคำนวณ    |                  |                     |                               |

# Pretest

# Post-test



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

# การสอน Coding สำหรับผู้เริ่มต้น

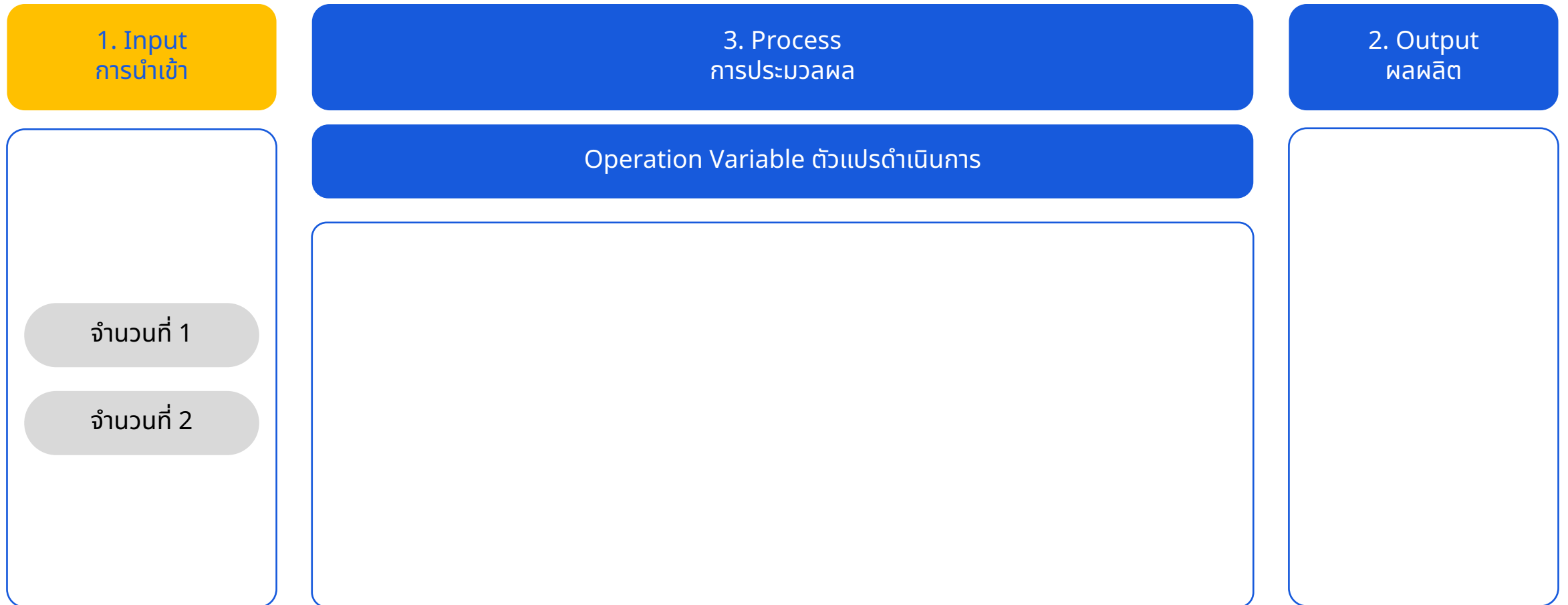
1. Input  
การนำเข้า

3. Process  
การประมวลผล

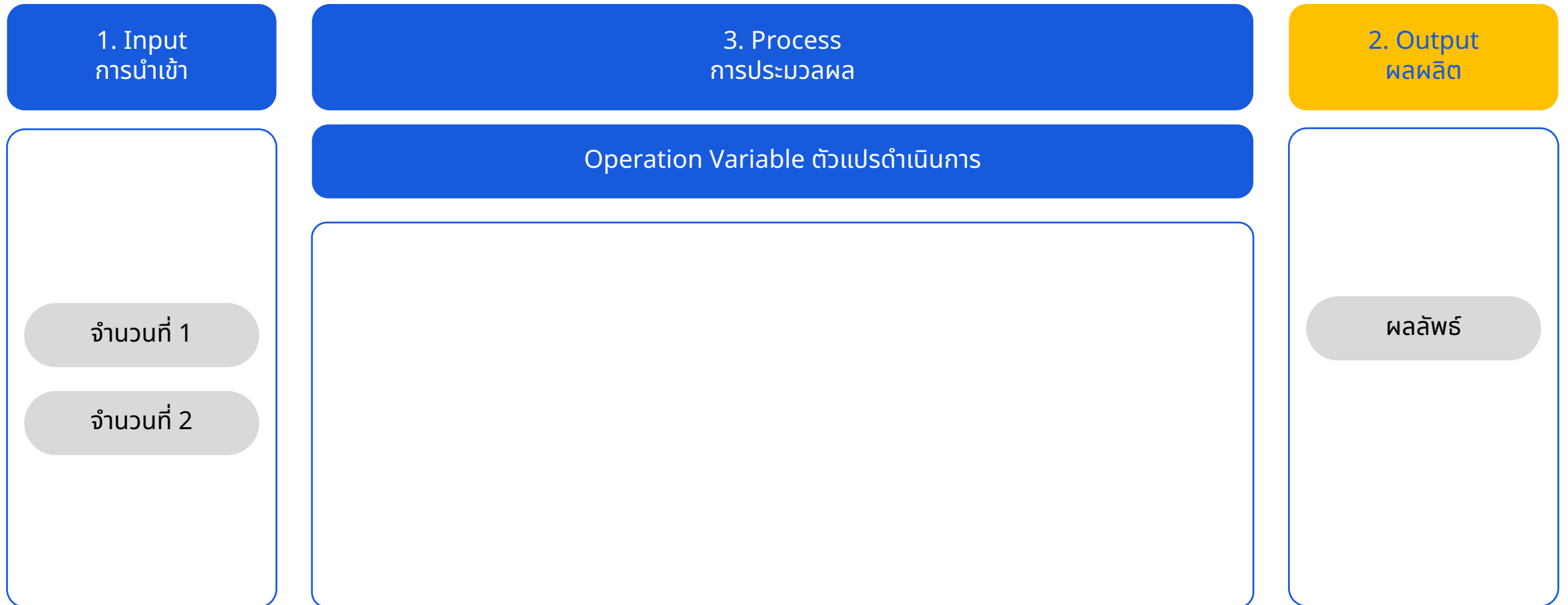
2. Output  
ผลผลิต

Operation Variable ตัวแปรดำเนินการ

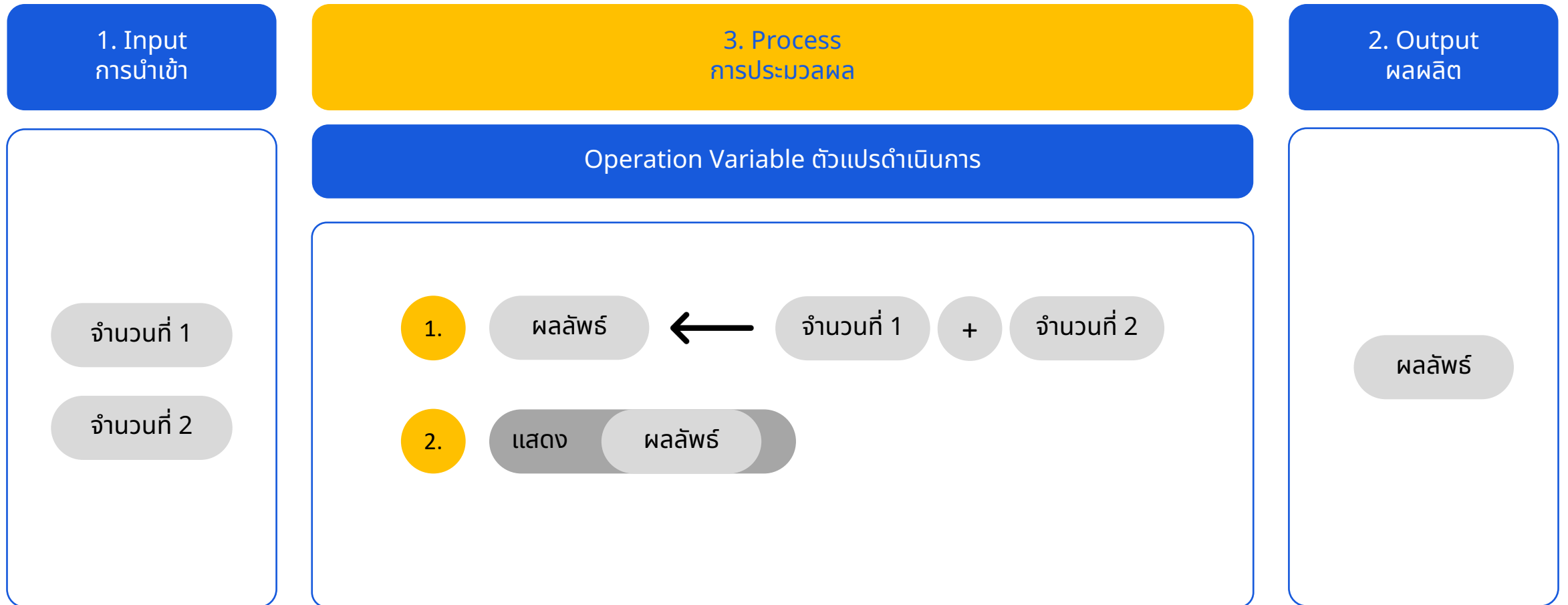
# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการผลรวมของค่า 2 จำนวน



# ต้องการคำนวณส่วนลดราคาสินค้า ราคาสินค้า 200 บาท ส่วนลด 10% ต้องจ่ายกี่บาท

1. Input  
การนำเข้า

ราคาสินค้า

ส่วนลด

3. Process  
การประมวลผล

Operation Variable ตัวแปรดำเนินการ

1. ส่วนลด 20 บาท ← ราคาสินค้า 200 × ส่วนลด 10 / 100

2. ราคาที่ต้องจ่าย 180 บาท ← ราคาสินค้า 200 บาท - ส่วนลด 20 บาท

3. แสดง ราคาที่ต้องจ่าย 180 บาท

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

2. Output  
ผลผลิต

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 ถือเป็นเวอร์ชันหลักในช่วงเวลาหลายปีก่อนที่จะมีการเปลี่ยนแปลงไปสู่ Python 3 ในปี 2008



# What is Python

- Python มีความนิยมอย่างแพร่หลายในวงการโปรแกรมเมอร์เนื่องจากมีความอ่านง่ายและเข้าใจได้ง่าย มีไวยากรณ์ที่เข้าใจง่าย และมีไลบรารีมากมายที่สามารถนำมาใช้งานได้ในงานต่างๆ เช่น
  - เว็บไซต์ (Website)
  - การวิเคราะห์ข้อมูล และงานด้าน AI
  - เครื่องมือที่ช่วยในการพัฒนาซอฟต์แวร์
- Python ยังมีความนิยมเนื่องจากมีชุมชนขนาดใหญ่ที่พร้อมแบ่งปันความรู้เกี่ยวกับ Python มีการพัฒนาและอัปเดตอย่างต่อเนื่องที่ช่วยให้ภาษานี้เติบโตและก้าวหน้าไปอย่างต่อเนื่องตามความต้องการของผู้ใช้งาน



# 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 มีชุมชนขนาดใหญ่และพร้อมแบ่งปันความรู้เกี่ยวกับ Python มีการสนับสนุนและแก้ไขปัญหามาจากผู้ใช้งานคนอื่น 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 เรียนรู้ง่ายและมีทรัพยากรการเรียนรู้ออนไลน์ที่มากมายที่สามารถใช้ในการศึกษาและการพัฒนาทักษะได้อย่างมีประสิทธิภาพ โดยเฉพาะ **STEM Education**

# Why Python

แทบจะทุกภาษาโปรแกรมมิ่งมีรูปแบบการเขียนโค้ดที่แตกต่างกันออกไปเล็กน้อย แต่ Python มีลักษณะที่แตกต่างอย่างมากจากภาษาโปรแกรมอื่น ดังนี้

- 1. โค้ดอ่านง่าย** Python ออกแบบมาให้เข้าใจง่าย คล้ายกับภาษาอังกฤษ โค้ดสามารถถูกอ่านและเข้าใจได้ง่าย ทำให้เป็นที่นิยมสำหรับผู้เพิ่งเริ่มต้นเรียนรู้การเขียนโปรแกรม
- 2. จบคำสั่งด้วยบรรทัดใหม่** Python ใช้บรรทัดใหม่เพื่อสิ้นสุดคำสั่ง ในขณะที่ภาษาโปรแกรมอื่นๆ บางครั้งอาจใช้เครื่องหมายเซมิโคลอน (;) หรือวงเล็บ () เพื่อแสดงถึงสิ้นสุดของคำสั่ง

- 3. การจัดระดับโดยการเยื้อง (Indentation)** Python ใช้การเยื้อง (indentation) เพื่อกำหนดขอบเขตของโค้ด เช่น ขอบเขตของลูป (loops), ฟังก์ชัน (functions), และคลาส (classes) ในขณะที่ภาษาโปรแกรมอื่น 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)
```

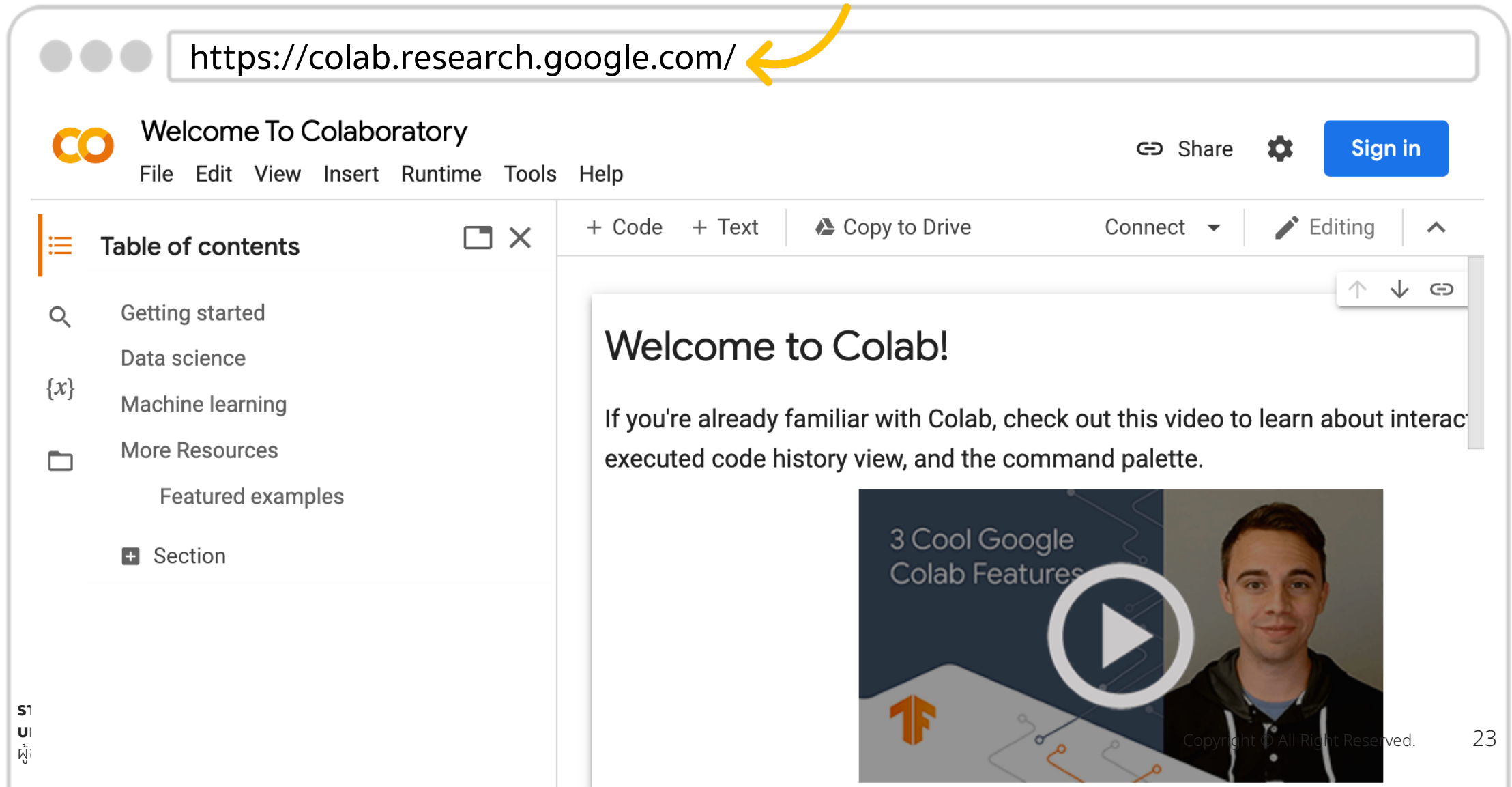
# Who is using Python today?

- 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, การสร้างโปรตอกายได้อย่างรวดเร็ว, การผสานระบบ, วิทยาศาสตร์ข้อมูล, แอปพลิเคชันฐานข้อมูล, การสื่อสารแบบเรียลไทม์ และอื่น ๆ อีกมากมาย
- มหาวิทยาลัยชั้นนำก็ได้้นำ Python เข้ามาใช้เป็นภาษาหลักในรายวิชาด้านวิทยาการคอมพิวเตอร์ด้วย

# Python Programming Steps Using Google Colab

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

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



The screenshot shows the Google Colaboratory website. The address bar at the top contains the URL <https://colab.research.google.com/>, with a yellow arrow pointing to it. The page header includes the Colab logo, the text "Welcome To Colaboratory", and navigation links for "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the header, there are links for "Share", a settings gear icon, and a blue "Sign in" button. Below the header, there is a sidebar on the left titled "Table of contents" with a search icon and a list of categories: "Getting started", "Data science", "Machine learning" (marked with a red 'x'), and "More Resources" (with a folder icon). Under "More Resources", there is a sub-section for "Featured examples" and a "+ Section" button. The main content area on the right has a toolbar with "+ Code", "+ Text", "Copy to Drive", "Connect", "Editing", and an up/down arrow. The main heading is "Welcome to Colab!". Below this, there is a paragraph of text: "If you're already familiar with Colab, check out this video to learn about interactive code history view, and the command palette." Below the text is a video player thumbnail with a play button icon. The video title is "3 Cool Google Colab Features" and it features a man's face. At the bottom right of the video player, there is a copyright notice: "Copyright © All Right Reserved." The page number "23" is visible in the bottom right corner.

# 2. Log in with your gmail

The screenshot shows the Google Colaboratory interface. At the top, the browser address bar displays `https://colab.research.google.com/`. Below the address bar, the page title is "Welcome To Colaboratory". To the right of the title are links for "Share" and "Settings", and a prominent blue "Sign in" button. A yellow arrow points from the top right towards the "Sign in" button. Below the title bar is a menu with options: "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the left side, there is a "Table of contents" sidebar with a search icon and a list of categories: "Getting started", "Data science", "Machine learning", "More Resources", and "Featured examples". The main content area features a header with "+ Code", "+ Text", "Copy to Drive", "Connect", and "Editing" options. The main heading is "Welcome to Colab!". Below this, a paragraph reads: "If you're already familiar with Colab, check out this video to learn about interactive code history view, and the command palette." A video thumbnail is shown with the text "3 Cool Google Colab Features" and a play button icon. The video thumbnail includes a copyright notice: "Copyright © All Right Reserved." The bottom left corner of the page has a vertical logo with the letters "S", "U", and "N". The page number "24" is located in the bottom right corner.

# 3. File -> New Notebook

The image shows a browser window at <https://colab.research.google.com/>. The page title is "Welcome To Colaboratory". The top navigation bar includes "Share", a settings gear, and a user profile icon. Below the navigation bar is a toolbar with "Code", "+ Text", "Copy to Drive", "Connect", "Editing", and a scroll indicator. The main content area displays "Welcome to Colab!" and a video thumbnail titled "3 Cool Google Colab Features".

Annotations include:

- A yellow circle with the number "1" and an arrow pointing to the "File" menu in the top-left corner.
- A yellow circle with the number "2" and an arrow pointing to the "New notebook" option in the "File" dropdown menu.

The "File" menu is open, showing the following options:

- New notebook
- Open notebook ⌘/Ctrl+O
- Upload notebook
- Rename
- Save a copy in Drive
- Save a copy as a GitHub Gist
- Save a copy in GitHub
- Save ⌘/Ctrl+S
- Revision history
- Download ▶
- Print ⌘/Ctrl+P

# 4. Coding with python commands on cell

The screenshot shows a Google Colab notebook interface. The browser address bar at the top displays `https://colab.research.google.com/`. The notebook title is `chapter1_example1.ipynb`, and the status indicates `All changes saved`. The main workspace contains a single code cell with the following Python code: `1 print("Hello World!")`. A yellow arrow points to the closing parenthesis of the print statement. The interface includes a left sidebar with navigation icons, a top toolbar with options like `+ Code`, `+ Text`, `Connect`, and `Editing`, and a right sidebar with `Comment`, `Share`, and a user profile icon.

# 5. Run cell

The screenshot shows a web browser window at <https://colab.research.google.com/>. The notebook is titled "chapter1\_example1.ipynb". The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", with a status "All changes saved". On the right, there are "Comment", "Share", and a user profile icon. Below the menu, there are controls for "+ Code", "+ Text", RAM and Disk usage (both with green checkmarks), and "Editing" mode. The main area shows a code cell with the code `1 print("Hello World!")`. A yellow arrow points to the play button (run icon) on the left of the code cell. Below the code, the output "Hello World!" is displayed with a green checkmark underneath it. A toolbar with various icons is visible to the right of the code cell.

# 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!

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

2 RUN

1 CODE



# Syntax – วากยสัมพันธ์ - กฎไวยากรณ์ภาษาคอมพิวเตอร์

- **Syntax ในบริบทของการเขียนโปรแกรม**  
หมายถึง กฎและรูปแบบที่ต้องปฏิบัติตาม  
เมื่อเขียนโค้ดในภาษาโปรแกรมนั้น ๆ ซึ่ง  
คล้ายกับการใช้ภาษาที่มีกฎไวยากรณ์ เช่น  
การใช้ภาษาอังกฤษ "I am a teacher" และ  
เราไม่สามารถใช้ "I is a teacher" ได้เพราะ  
ผิดกฎไวยากรณ์ของภาษาอังกฤษ
- ดังนั้น คำว่า "syntax" จึงหมายถึงกฎเกณฑ์  
ที่ใช้ในการเขียนโค้ด เพื่อให้โปรแกรมทำงาน  
ได้อย่างถูกต้อง

ภาษาโปรแกรมแต่ละภาษามี syntax ที่แตกต่างกันไป  
เช่น Python, JavaScript, C++, Java จะมี syntax ที่ไม่  
เหมือนกัน การเขียนโค้ดต้องเป็นไปตาม syntax ของ  
ภาษาโปรแกรมนั้น ๆ โดยถ้ามี syntax error หรือ bug  
(ข้อผิดพลาดในรูปแบบ) โปรแกรมจะไม่สามารถรันหรือ  
ทำงานได้อย่างถูกต้อง

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

ในตัวอย่างนี้มี syntax error เนื่องจากไม่มีเครื่องหมาย "  
(double quote) ปิดที่สมบูรณ์ ซึ่งจะทำให้โปรแกรมไม่  
สามารถรันได้ เนื่องจากไม่เป็น syntax ที่ถูกต้องใน Python

# Reserved Words – คำสงวน

## Reserved Words หรือ คำสงวน

ในภาษา Python คือ คำที่ถูกสงวนไว้เพื่อใช้ในบางไวยากรณ์ของภาษา Python เท่านั้น โดยคำเหล่านี้มีประมาณ 35 คำ และมีหน้าที่เฉพาะในการควบคุมโครงสร้างและการทำงานของโปรแกรม ซึ่งไม่สามารถใช้เป็นชื่อตัวแปร หรือ ตั้งชื่อฟังก์ชันได้

- and
- as
- assert
- async [note 1]
- await [note 1]
- break
- class
- continue
- def

- del
- elif
- else
- except
- False [note 2]
- finally
- for
- from
- global

- if
- import
- in
- is
- lambda
- None
- nonlocal [note 3]
- not
- or

- pass
- raise
- return
- True [note 2]
- try
- while
- with
- yield

# Reserved Words - คำสงวน

## 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()` เป็นการแสดงข้อความหรือค่าใดๆ ผ่านทางหน้าจอ
- โดยทำการส่งข้อความหรือค่าที่ต้องการแสดงเป็นอาร์กิวเมนต์เข้าไปในฟังก์ชัน `print()` ซึ่งจะแสดงผลออกทางหน้าจอตามที่ระบุ

## Syntax:

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

โครงสร้างของฟังก์ชัน `print()` ใน Python มีรายละเอียดดังนี้:

**\*objects** เป็น parameter ที่รับค่าที่ต้องการแสดงผลบนหน้าจอ สามารถรับค่าหลาย ๆ ตัว พร้อมกันได้ ซึ่งทำให้ `print()` สามารถรับค่าจากหลายตัวแปรหรือค่าที่ต่างกันก็ได้

**sep=' '** parameter ที่กำหนดวิธีการคั่นระหว่าง objects ที่ถูกแสดงผล หากไม่ได้กำหนดค่า parameter จะใช้ช่องว่างเป็นค่าเริ่มต้น

**end='\n'** parameter ที่กำหนดวิธีการจบบรรทัดหรือการลงบรรทัดใหม่หลังจากที่แสดงผลข้อความ ค่าเริ่มต้นคือ การลงบรรทัดใหม่ (`\n`)

## Code 2

```
1. # Code 2
2. print('My name is')
3. print('Nutthapat')
4. print('Kaewrattanapat')
```

```
My name is
Nutthapat
Kaewrattanapat
```

## Code 3

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

My name isNutthapatKaewrattanapat

## Code 4

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

```
My name is Nutthapat Kaewrattanapat
```

## 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")
```

```
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) หรือเครื่องหมาย underscore (\_) เท่านั้น ไม่สามารถขึ้นต้นด้วยตัวเลขได้
2. **ไม่สามารถขึ้นต้นด้วยตัวเลข** ชื่อตัวแปรไม่สามารถขึ้นต้นด้วยตัวเลขได้ แต่สามารถประกอบไปด้วยตัวอักษรหรือตัวเลขหลังตัวอักษรแรกได้
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

## ตัวอย่างการตั้งชื่อตัวแปรที่ถูก

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 !)

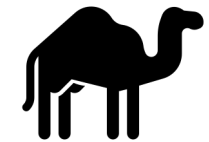
break (เป็น keyword ของ Python)

# 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'
```



Ref: [https://www.w3schools.com/python/python\\_variables\\_names.asp](https://www.w3schools.com/python/python_variables_names.asp)

## 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

Nutthapat 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=' ')
```

```
Nutthapat Score: 80
```

# Numbers - จำนวน

1. จำนวนเต็ม (Integers) เป็นจำนวนที่ค่าต่างๆ ไม่มีทศนิยม  
เช่น -5, 0, 100
2. จำนวนทศนิยม (Floating-point numbers) เป็นจำนวนที่ค่าต่างๆ 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))
```

**ฟังก์ชัน type()** เป็นฟังก์ชันที่ใช้ตรวจสอบประเภท (type) ของวัตถุหรือค่าที่รับเข้ามาในพารามิเตอร์ของฟังก์ชันนี้ ฟังก์ชัน type() จะคืนค่าประเภทของวัตถุนั้นออกมาในรูปของ object type หรือ ชนิดของข้อมูลที่ได้รับ

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

**<class 'int'>** หมายถึง ประเภทข้อมูลนี้จัดกลุ่มเป็น integer หรือ ค่าจำนวนเต็ม  
**<class 'float'>** หมายถึง ประเภทข้อมูลนี้จัดกลุ่มเป็น 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 การควบคุมการทำงานของลูป หรือเงื่อนไขในการทำงานของโค้ด

```
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" และแสดงผลจริงหรือเท็จ
4. ตรวจสอบว่า password มีค่าเท่ากับ "99999" และแสดงผลจริงหรือเท็จ
5. ตรวจสอบว่า password มีค่าเท่ากับ "90009" และแสดงผลจริงหรือเท็จ

### 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: Nutthapat
Hello, Nutthapat
```

## 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

```
Please enter Firstname: Nutthapat
Please enter Lastname: Kaewrattanapat
Please enter Birth Year: 1983
Nutthapat Kaewrattanapat 40
```

## Practice 5 - Error

```
Please enter Firstname: Nutthapat
Please enter Lastname: Kaewrattanapat
Please enter Birth Year: 1983
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-22-548c4cf3a793> in <cell line: 5>()
      3 lastname = input("Please enter Lastname: ")
      4 y_birth = input("Please enter Birth Year: ")
----> 5 print(firstname, lastname, 2023-y_birth)

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

**Error ที่เกิดขึ้นเป็น TypeError** ซึ่งบอกว่ามีประเภทข้อมูลไม่สามารถทำการดำเนินการบวกลบได้ระหว่างตัวแปรข้อมูลประเภทต่าง ๆ ในกรณีนี้ โค้ดพยายามที่จะคำนวณอายุโดยการลบปีปัจจุบันกับปีที่เกิด แต่มีปัญหาเกิดขึ้นกับข้อมูลที่รับจากผู้ใช้ที่ป้อนผ่านฟังก์ชัน `input()` เนื่องจาก `input()` จะรับข้อมูลเป็นข้อความ (String) ไม่ใช่ตัวเลข (Integer)

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

## Practice 5

```
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, 2023 - y_birth)
```

```
Please enter Firstname: Nutthapat
Please enter Lastname: Kaewrattanapat
Please enter Birth Year: 1983
Nutthapat Kaewrattanapat 40
```

# 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

**Operators หรือ ตัวดำเนินการ** ใน Python คือ สัญลักษณ์ที่ใช้ในการดำเนินการทางคณิตศาสตร์ หรือ เปรียบเทียบข้อมูล เพื่อให้เกิดการกระทำที่ต้องการในโปรแกรม ซึ่งสามารถทำได้กับ Operand (ตัวถูกดำเนินการ) ตัวอื่น ๆ

- ตัวดำเนินการทางคณิตศาสตร์ เช่น + (บวก), - (ลบ), \* (คูณ), / (หาร)
- ตัวดำเนินการเปรียบเทียบ เช่น == (เท่ากับ), != (ไม่เท่ากับ), > (มากกว่า), < (น้อยกว่า)
- ตัวดำเนินการทางตรรกศาสตร์ เช่น and (และ), or (หรือ), not (ไม่)

# 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

## Comparison Operators หรือ ตัวดำเนินการเปรียบเทียบ

== (เท่ากับ)

!= (ไม่เท่ากับ)

> (มากกว่า)

< (น้อยกว่า)

>= (มากกว่าหรือเท่ากับ)

<= (น้อยกว่าหรือเท่ากับ)

```
1. x = 10
2. y = 5
3. print(x == y) # Output: False
4. print(x != y) # Output: True
5. print(x > y) # Output: True
```

# 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 1 (2 Points)

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