



CSS 112 Computer Programming

Lecture I: Introduction to Python Programming

Suvil Chomchaiya (Ph.D.)





Welcome Abroad.....

Hello!



I am Suvil Chomchaiya

Your course instructor for the 1st half of the semester.



You can contact me at @
E-mail: suvil.cho@kmutt.ac.th

Contents

- All about Python
- Programming with Python





All about Python



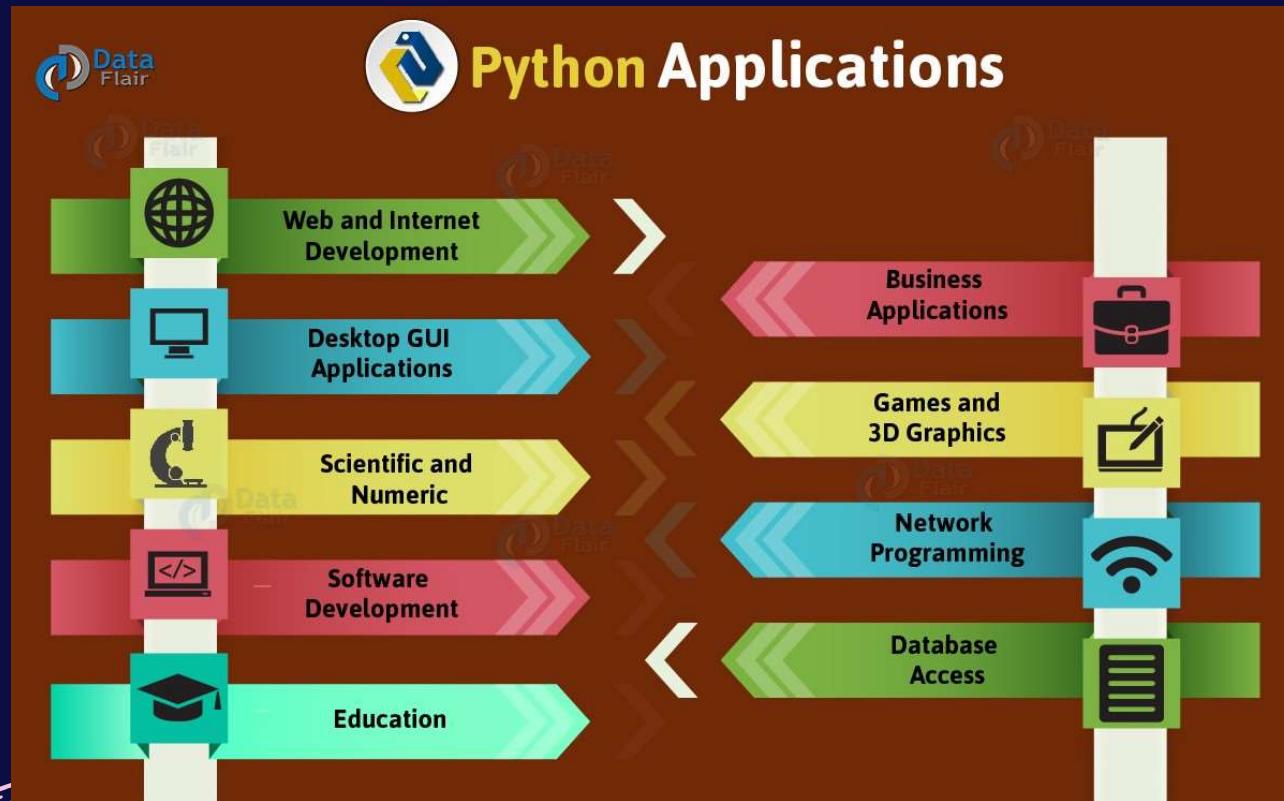


What is Python

- Python เป็นภาษา Dynamic Object-Oriented Programming ที่ถูกพัฒนาขึ้นโดย Guido von Rossum ในปี ค.ศ. 1990
- ปัจจุบันถูกดูแลโดย Python Software Foundation (PSF) โดย Version ล่าสุดตอนนี้คือ Python 3.1 หรือเรียกอีกอย่างว่า Python 3000 หรือ Py3k



Python Application



Source: <https://www.ultimatepython.co/post/study-plan>

KM
UTT



What's Good about Python

- โครงสร้างภาษาไม่มีความซับซ้อน เข้าใจง่าย
- จัดการหน่วยความจำได้แบบอัตโนมัติและมีประสิทธิภาพ (Automatic Memory Management)
- ทำงานร่วมกันได้กับทุกระบบ (Platform Independent) Windows, Mac, Linux, Raspberry Pi และอื่น ๆ
- Python มีไวยากรณ์การเขียนภาษาโปรแกรมที่เข้าใจง่าย เหมือนเขียนภาษาอังกฤษทั่วไป





What's Good about Python

- รองรับการเขียนโปรแกรมแบบแนวคิดหลากหลาย (**Multi-Paradigm Programming**) เช่น
 - การเขียนโปรแกรมเชิงโครงสร้าง (**Structural Programming**)
 - การเขียนโปรแกรมเชิงฟังก์ชัน (**Functional Programming**)
 - การเขียนโปรแกรมเชิงวัตถุ (**Object-Oriented Programming**)
 - การเขียนโปรแกรมจัดการฐานข้อมูล (**Database Programming**)
 - การเขียนโปรแกรมเพื่อการทำข้อมูลให้เป็นภาพ (**Data Visualization Programming**)





Inside the Python

- โครงสร้างของ Python ที่มีการแบ่งข้อมูลเป็นหลายประเภทที่จะถูกเรียกว่า **OBJECT**
- การทำงานกับ Python จะอาศัยความเข้าใจ **OBJECT** ต่างๆ นี้เป็นหลัก
- OBJECT** แต่ละประเภทจะมีเครื่องมือ และการใช้งานที่แตกต่างกัน โดยการกระทำต่างๆ คำสั่งหลักๆ ที่เราจะใช้จะเป็นคำสั่งที่ใช้งานกับ **OBJECT**
- เมื่อสั่งคำสั่งได้แล้ว ขึ้นตัดไป คือการสร้างเงื่อนไข หรือการกำหนดการทำงานของแต่ละคำสั่ง



Inside the Python

- การกำหนดเงื่อนไขเหล่านี้จะใช้สิ่งที่เรียกว่า **STATEMENT** หรืออาจเรียกว่ารวมๆ ได้ว่าเป็น เงื่อนไข ที่จะมาช่วยให้การกระทำ คำสั่งทำงานได้ชัดเจน และครอบคลุมมากขึ้นมากกว่าการเรียกคำสั่งที่ละเอียด
- เครื่องมือต่างๆ ใน Python จะถูกจัดเก็บไว้เป็นกลุ่มคำสั่งที่เรียกว่า **LIBRARY** ซึ่ง ในแต่ละ **LIBRARY** จะรวบรวมคำสั่งที่ใช้ร่วมกันเข้าไว้ภายในตัวข้อได้หัวข้อหนึ่ง



Learning Python



Source: <https://www.ultimatepython.co/post/study-plan>

KM
UTT

Python Programming Tools



- **Kodomo** – มีลิขสิทธิ์ มีค่าใช้จ่าย รองรับหลายภาษา เช่น Python, PHP, C, Java Script...
- **PyScripter** – ใช้งานได้ฟรีสำหรับระบบปฏิบัติการ Windows
- **NetBeans** – ใช้พัฒนาโปรแกรม application ด้วยภาษาต่างๆ รวมถึง Python ใช้งานฟรี
- **Pycharm** – เวอร์ชัน Edu สามารถใช้งานได้ฟรี





How simple the Python V.S. Java...

- โปรแกรม “Hello World” (Java)

```
public class
{
    public static void main(String[]args)
    {
        System.out.println("Hello, world!");
    }
}
```



How simple the Python V.S. Java...



- โปรแกรม “Hello World” (Python)

```
print "Hello, world!"
```



Python Installation: via www.python.org



Python 3: Fibonacci series up to n

```
>>> def fib(n):
...     a, b = 0, 1
...     while a < n:
...         print(a, end=' ')
...         a, b = b, a+b
...     print()
... fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)

47684398_102185...jpg Quintus - SlidesC....pptx Moonstone Free P....zip intro22 (2).ppt intro22 (1).ppt Removed Show all



Python Installation: via www.python.org

All releases

Source code

Windows

macOS

Other Platforms

License

Alternative Implementations

Download for Windows

Python 3.10.6

Note that Python 3.9+ cannot be used on Windows 7 or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

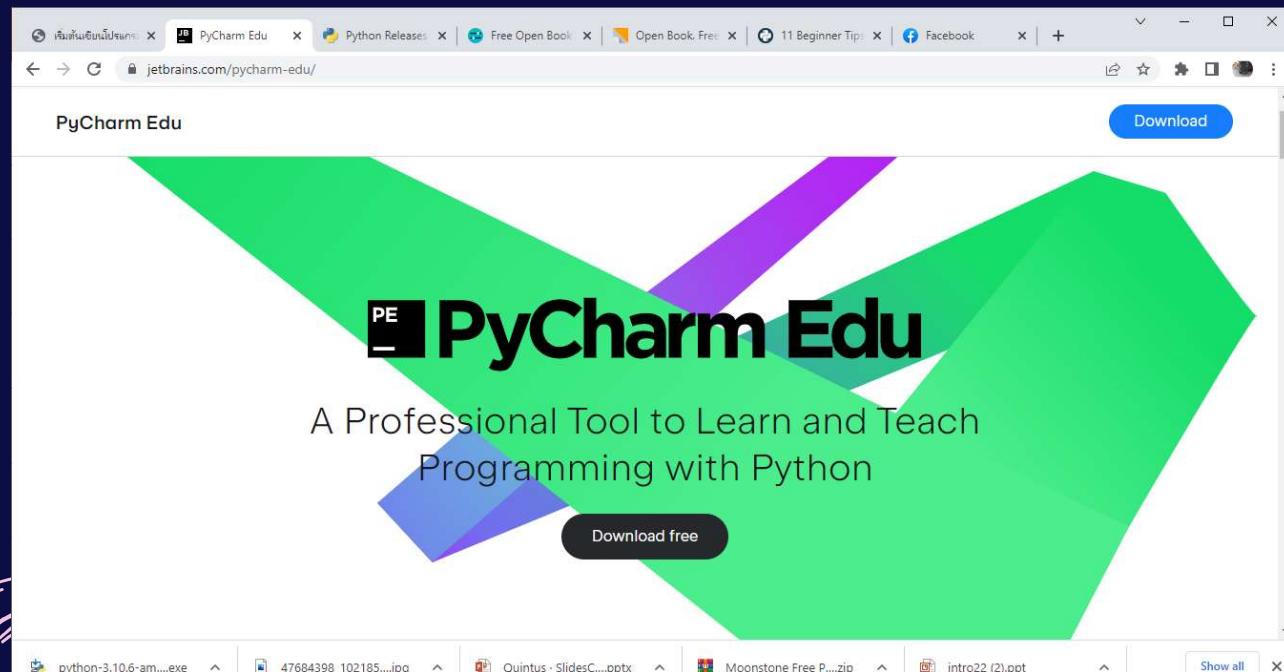
[View the full list of downloads.](#)

ดาวน์โหลดไฟล์ **python-3.10.6-amd64.exe**



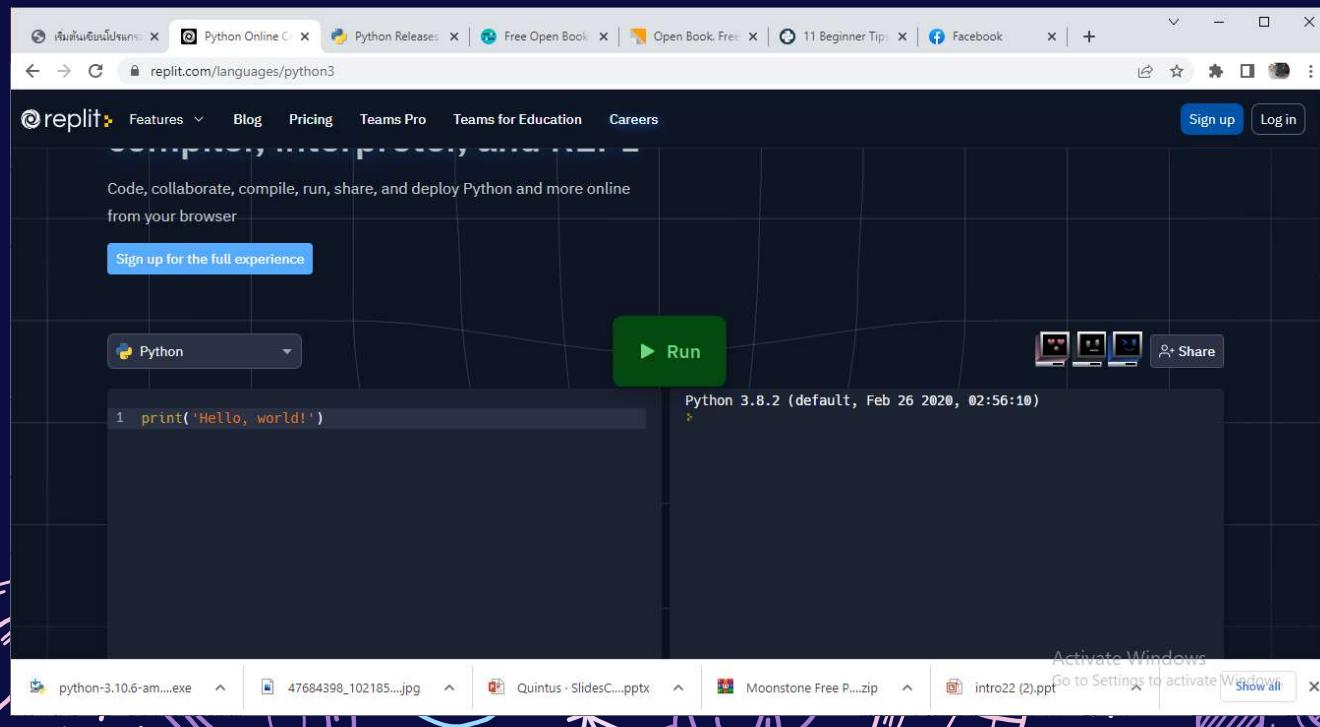
Python Installation: PyCharm Edu

- ເວີບໄຊຕົ້ນ <http://www.jetbrains.com/pycharm-edu/>



Python on the Web: Web-base Python Editor

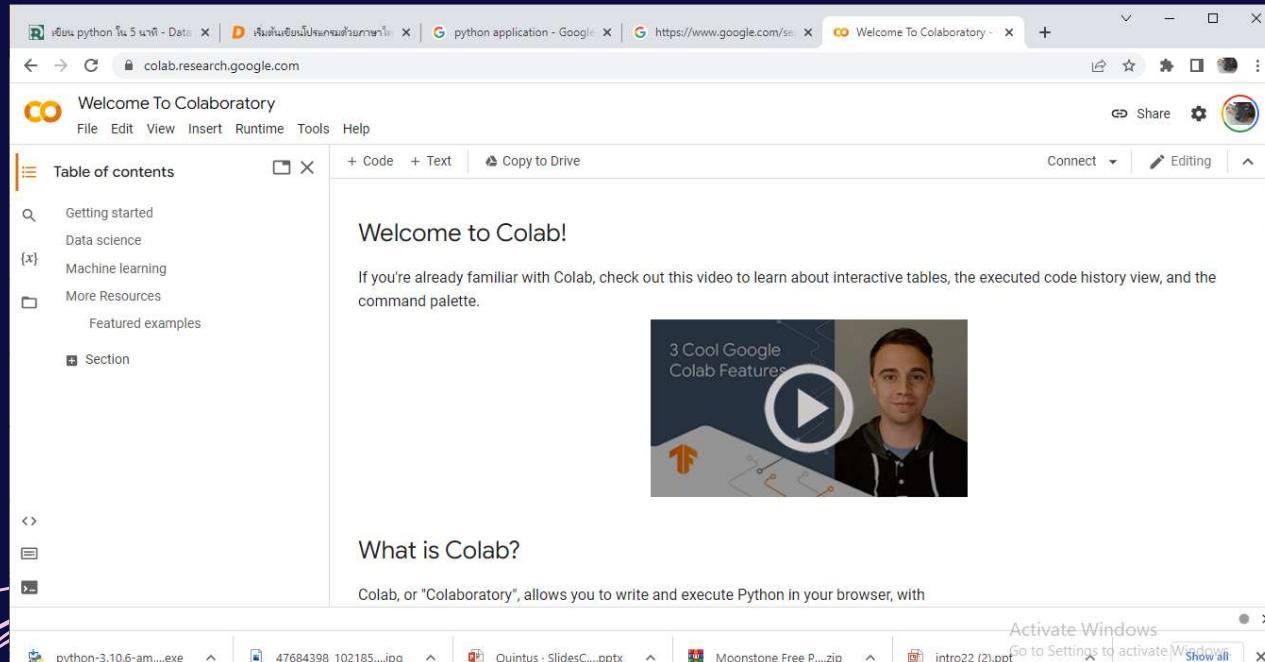
- <https://repl.it/languages/python3>





Python on the Web: Google Colab

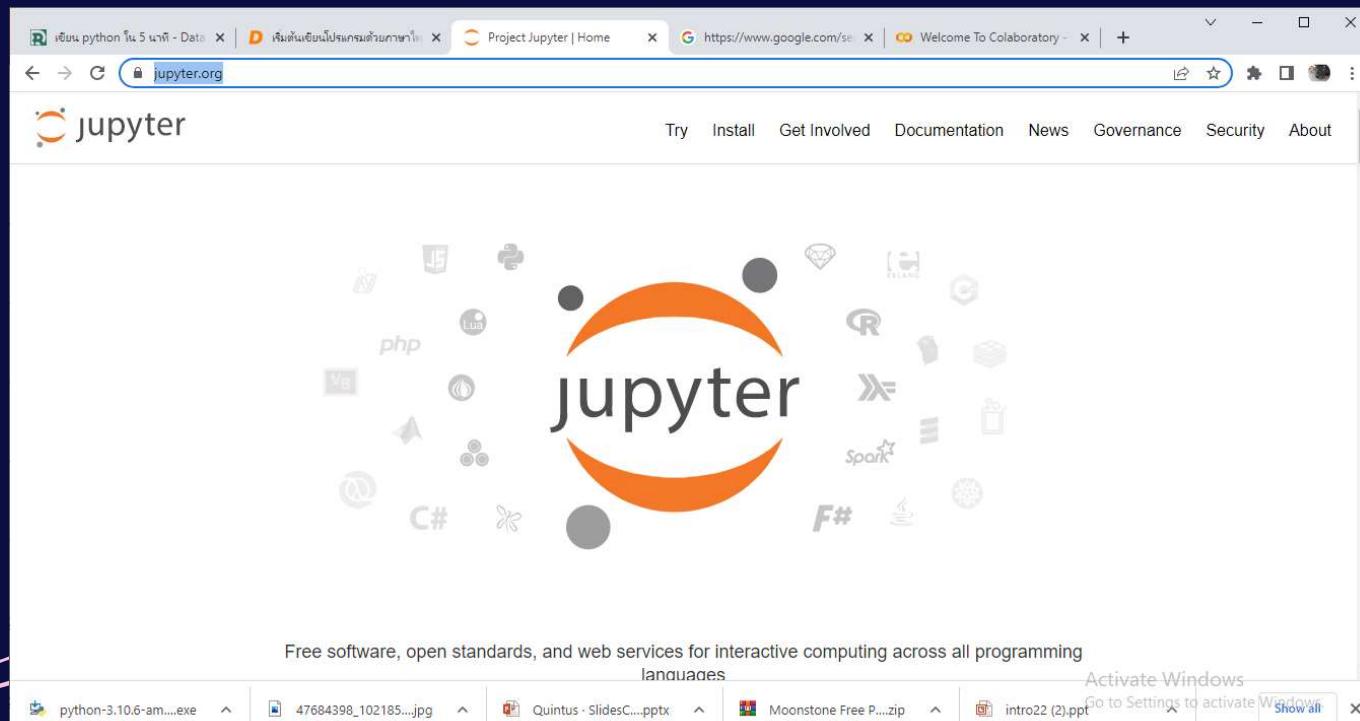
- <https://colab.research.google.com/>





Python on the Web: JupyterLab

- <https://jupyter.org/>





Programming with Python



Python Syntax



- Python Syntax (รูปแบบการเขียนโปรแกรมภาษา Python)
 - ภาษา Python มีรูปแบบที่แตกต่างกับการเขียนโปรแกรมในภาษาอื่น ๆ อยู่พอสมควร เช่น
 - ไม่ต้องมีเซมิโคลอน (;) ปิดท้ายคำสั่ง
 - ให้ความสำคัญกับการจัดย่อหน้า (Indentation)



Python Indentation



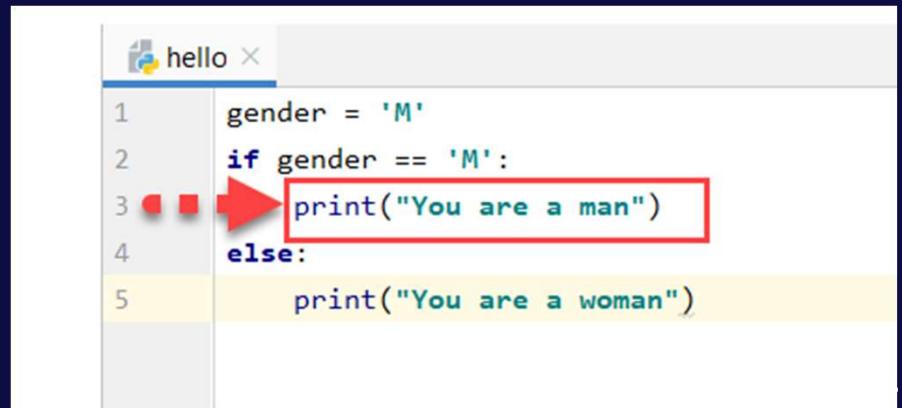
- **Python Syntax** รูปแบบการเขียนโปรแกรมภาษา Python
 - ในภาษาเขียนโปรแกรมอื่น ๆ นั้น การจัดย่อหน้าเพียงเพื่อให้เกิดความสวยงาม อ่านโค้ดง่าย โดยไม่มีผลต่อการทำงานของโปรแกรมเลย
 - แต่สำหรับภาษา Python นั้นไม่ใช่ เพราะ Python ใช้การจัดย่อหน้าเพื่อปั่งบอกขอบเขตของโค้ด (Block of code)



Python Indentation

- จากโค้ดตัวอย่าง บรรทัดที่ 2 เป็นการเขียนคำสั่งตรวจสอบเงื่อนไขด้วย if
- โค้ดบรรทัดถัดมาที่เป็นคำสั่งทำงานตามเงื่อนไข ต้องย่อหน้าเข้ามา 1 แท็บ เพื่อป้องกันการบอกให้รู้ว่าโค้ดบรรทัดนี้อยู่ภายในบล็อกของคำสั่ง if

```
Python
1 gender = 'M'
2 if gender == 'M':
3     print("You are a man")
4 else:
5     print("You are a woman")
```



```
hello x
1 gender = 'M'
2 if gender == 'M':
3     print("You are a man")
4 else:
5     print("You are a woman")
```

Source: www.dcrub.com



Python Indentation

- ถ้าไม่มีการจัดย่อหน้าดังกล่าว ตัว Editor จะแจ้งข้อผิดพลาดว่าต้องจัดย่อหน้า
- ตัวอย่าง (ในตัวอย่างใช้ PyCharm) โปรแกรมจะแจ้งข้อผิดพลาดในโค้ดบรรทัดที่ 3 ว่า **Indent expected** คือบรรทัดนี้ต้องมีการจัดย่อหน้า เพราะเป็นคำสั่งที่อยู่ภายใต้คำสั่ง if อีกที

```
hello x
1 gender = 'M'
2 if gender == 'M':
3     print("You are a man")
4
5     print("You are a woman")
```

Indent expected

Source: www.dcrub.com

Python Indentation



- ในแต่ละบรรทัดคำสั่ง สามารถจัดย่อหน้าโดยเคาะแท็บเกิน 1 แท็บก็ได้ โดยจะไม่มีผลกระทบต่อการทำงานของโปรแกรม
- สรุปว่า จะจัดย่อหน้าแท็บกี่แท็บก็ได้ แต่อย่างน้อยที่สุดต้องมี 1 แท็บ
- เพียงแต่โค้ดทุกบรรทัดที่อยู่ในบรรทัดเดียวกัน ต้องจัดแท็บให้เท่ากัน
- การจัดย่อหน้าใน Python ไม่ได้จำกัดว่าต้องจัดด้วยปุ่ม Tab เท่านั้น แต่ยังสามารถจัดย่อหน้าด้วยปุ่ม Space bar คือจัดย่อหน้าด้วยช่องว่างก็ได้ เช่นกัน
- แต่ก็ต้อง space bar ให้เท่ากันด้วย



Python Indentation

- ตัวอย่างโค้ดที่ถูกต้อง
- จัดแท็บไม่เท่ากันจะเกิด Error ขึ้น

```
Python
1 gender = 'M'
2 if gender == 'M':
3     print("You are a man")
4     print("You are handsome")
5 else:
6     print("You are a woman")
```

```
hello
1 gender = 'M'
2 if gender == 'M':
3     print("You are a man")
4     print("You are a boy")
5 else:
6     print("You")
```

แบบนี้ผิด

Statement expected, found Py:DEDENT

Source: www.dcrub.com



Code Comments in Python

- Comments คือการเขียนคำอธิบายต่างๆ ใน code ของตัวโปรแกรมโดยไม่มีผลต่อการทำงานของโปรแกรม
- การเขียนคอมเม้นต์ Comments ในภาษา Python จะใช้เครื่องหมาย # ขึ้นต้นบรรทัดหรือโดยคทีต้องการคอมเม้นต์ ตามด้วยประโยคทีต้องการทำเป็นคอมเม้นต์
- ตัวแปลงภาษา (compiler) จะไม่ประมวลผลคำสั่งใด ๆ ทีอยู่หลังเครื่องหมาย #





Code Comments in Python

- ตัวอย่างการเขียนคอมเม้นต์ในภาษา Python
- จาก code ตัวอย่างด้านขวา ในบรรทัดที่ 3 และบรรทัดที่ 6 เป็น comments ตัวภาษาจะไม่นำข้อความดังต่อไปนี้เป็นต้นไป ไปประมวลผล

Python

```
1 gender = 'M'  
2 if gender == 'M':  
3     # Print if gender is M  
4     print("You are a man")  
5 else:  
6     # Print if gender is not M  
7     print("You are a woman")
```

Source: www.dcrub.com



Output Function in Python

- ในภาษา Python เมื่อต้องการแสดงข้อมูลออกทางหน้าจอ จะใช้ฟังก์ชัน `print()` พร้อมระบุข้อมูลที่ต้องการแสดง เช่น `print(ข้อมูลที่ต้องการแสดง)` หรือ `print(ข้อมูล, ข้อมูล, ข้อมูล, ...)`
- ข้อมูลที่จะแสดงด้วยฟังก์ชัน `print()` ถ้าเป็น `string` (ข้อความ) ให้เขียนไว้ในเครื่องหมาย `""` หรือ `''`
- ข้อมูลที่จะแสดงด้วยฟังก์ชัน `print()` ถ้าเป็นตัวเลข สามารถเขียนลงไปได้โดยตรง
- ถ้าจะแสดงค่าจากตัวแปร ไม่ว่าตัวแปรนั้นจะเก็บข้อมูลชนิดใดก็ตาม สามารถเขียนตัวปั๊ปรลงไปได้โดยตรง



Output Function in Python

- เมื่อใช้ฟังก์ชัน `print()` หลังจากแสดงข้อความแล้วจะขึ้นบรรทัดใหม่โดยอัตโนมัติ
- การใช้ฟังก์ชัน `print()` แสดงข้อมูลหลายค่าในครั้งเดียว จะเป็นการแสดงค่าข้อมูลหล่านั้นในบรรทัดเดียวกัน
- `print` ไม่ใช่ `Print` ถ้าสะกดใช้ตัว `P` เป็นตัวพิมพ์ใหญ่จะเกิด `error`
- ตัวอย่างการใช้งานฟังก์ชัน `print()`

```
Python
1 name = 'Lucy'
2 sirname = 'Johnson'
3 print('Hello', 'My name is', name, sirname)
4 #ผลลัพธ์ : Hello My name is Lucy Johnson
```

Source: www.dcrub.com



Output Function in Python

- กฎใช้งาน print() ยังสามารถใช้ในรูปแบบที่ซับซ้อนมากขึ้น มีรูปแบบดังนี้

```
print(data, data, data, ..., sep=...)
print(data, data, data, ..., end=...)
print(data, data, data, ..., sep=..., end=...)
```

Source: www.dcrub.com

- data คือข้อมูล หรือตัวแปรที่เก็บค่าข้อมูลที่ต้องการนำมาแสดง ไม่จำกัดจำนวน

- `sep` และ `end` คือ **Keyword Argument** เป็นตัวเลือกเพิ่มเติม จะระบุหรือไม่ก็ได้



Output Function in Python

- `sep` มาจากคำว่า `separator` ใช้ระบุว่าต้องการคั่นระหว่างข้อมูลแต่ละตัวด้วยอะไร
- ค่าที่จะกำหนดให้ `sep` ต้องอยู่ในรูปแบบของ `string` ที่ครอบด้วยเครื่องหมาย `''` หรือ `''''`
- `sep` จะระบุหรือไม่ก็ได้ ถ้าหากไม่ระบุ ข้อมูลจะถูกคั่นด้วยช่องว่าง





Output Function in Python

- ตัวอย่างการใช้ฟังก์ชัน print แบบระบุ sep และไม่ระบุ sep

```
Python
1 a  = 'Lucy'
2 b  = 'Johnson'
3 # ไม่ระบุ sep
4 print('Wanchai', 'Wanida', a, b)
5 # ผลลัพธ์ : Wanchai Wanida Lucy Johnson
6
7 # ระบุ sep
8 print('Wanchai', 'Wanida', a, b, sep=",")
9 # ผลลัพธ์ : Wanchai,Wanida,Lucy,Johnson
```

Source: www.dcrub.com



Output Function in Python

- end คือสิ่งที่จะเขียนต่อท้ายข้อมูลที่นำมาแสดงในฟังก์ชัน print()
- ค่าของ end ต้องอยู่ในรูปแบบของ string ครอบด้วยเครื่องหมาย ‘ ’ หรือ “ ”
- โดยปกติ เมื่อใช้ฟังก์ชัน print() ถ้าไม่ระบุคีย์เวิร์ด end ข้อมูลที่แสดงจะถูกต่อท้ายด้วย ‘\n’ ซึ่งเป็นสัญลักษณ์สำหรับการขึ้นบรรทัดใหม่





Output Function in Python

- ตัวอย่างการใช้ฟังก์ชัน `print()` แบบระบุคีย์เวิร์ด `end` และไม่ระบุ

```
Python
1 a  = 'Lucy'
2 b  = 'Johnson'
3 print('Wanchai', 'Wanida', a, b)
4 print('Wanchai', 'Wanida', a, b, end="**")
5 print('Wanchai', 'Wanida', a, b, end="")
```

- ผลลัพธ์

```
Wanchai Wanida Lucy Johnson
Wanchai Wanida Lucy Johnson**Wanchai Wanida Lucy Johnson
```

Source: www.dtcub.com



Output Function in Python

ผลลัพธ์ของโค้ดบรรทัดที่ 3 (**Wanchai Wanida Lucy Johnson**) แสดงข้อมูลแล้วขึ้นบรรทัดใหม่ เพราะไม่ได้ระบุคีย์เวิร์ด **end** ตัวแปลภาษาจึงใส่ค่าตั้งต้นให้เป็น **end='\\n'** โดยอัตโนมัติ ทำให้มีการขึ้นบรรทัดใหม่

ผลลัพธ์ของโค้ดบรรทัดที่ 4 และ 5 (**Wanchai Wanida Lucy Johnson**Wanchai Wanida Lucy Johnson**) ถูกแสดงผลในบรรทัดเดียวกัน เพราะโค้ดบรรทัดที่ 4 ระบุคีย์เวิร์ด **end** เป็น ******* ทำให้มีเครื่องหมาย ****** ต่อท้ายข้อมูลและไม่ขึ้นบรรทัดใหม่

Wanchai Wanida Lucy Johnson
Wanchai Wanida Lucy Johnson**Wanchai Wanida Lucy Johnson

Source: www.dcrub.com

ผลลัพธ์ของโค้ดบรรทัดที่ 5 จึงถูกแสดงออกมาในบรรทัดเดียวกัน กับผลลัพธ์ของโค้ดในบรรทัดที่ 4 และไม่ขึ้นบรรทัดใหม่ เพราะระบุคีย์เวิร์ด **end** แต่ไม่ได้ใส่ค่าใดๆ

เทคนิคการเรียน เขียนโปรแกรมภาษา Python



เทคนิค 6 ประการในการเริ่มเรียนเขียน โปรแกรมภาษา Python



Tips#1 – เข้าใจในพื้นฐานของภาษา Python

- ความเข้าใจในพื้นฐานของภาษา Python จะช่วยให้ก้าวไปสู่วิชาโครงสร้างข้อมูล (Data Structures) และการวิเคราะห์ขั้นตอนวิธี (Algorithms Analysis) ได้อย่างมั่นใจ

- สำหรับคนที่ไม่เคยเขียนโปรแกรมมาก่อนอาจจะรู้สึกว่ายาก แต่ทุกคนที่ชำนุญภาษา Python ตัววันนี้เคยเริ่มต้นที่จุดเดียวกัน



เทคนิค 6 ประการในการเริ่มเรียนเขียน โปรแกรมภาษา Python

Tips#2 – ทดลองเขียน Code ภาษา Python ลงบนกระดาษ

- การใช้ Code Editor ที่แก้ไขข้อผิดพลาด (Error) ให้โดยอัตโนมัติ อาจทำให้มองข้ามหรือไม่ได้ตระหนักรถึงจุดผิดพลาดเล็กน้อยได้ตลอดเวลา
- การทดลองเขียน code ด้วยมือลงบนกระดาษจะช่วยให้คุณเดยและจดจำได้ดีกว่า



เทคนิค 6 ประการในการเริ่มเรียนเขียน โปรแกรมภาษา Python



Tips#3 – การฝึกฝน

- ฝึกการเขียน code วันละนิดอย่างสม่ำเสมอเป็นประจำทุกวันเพื่อเป็นการทบทวน และสร้างความคุ้นเคยกับการเขียนโปรแกรม
- การฝึกเขียนโปรแกรมอย่างสม่ำเสมอจะช่วยให้มีโอกาสได้พบเจอบัญหาที่หลากหลาย และพัฒนาเทคนิคการแก้ไขบัญหาต่างๆ ในการเขียน





เทคนิค 6 ในการเริ่มเรียนเขียน โปรแกรมภาษา Python

Tips#4 – มีส่วนร่วมใน Open Source

- การแลกเปลี่ยนการเรียนรู้เป็นวิธีในการพัฒนาทักษะกับนักเขียนโปรแกรมที่มีความสนใจแบบเดียวกัน.
- การมีส่วนร่วมใน shared platform จะสามารถเป็นทางลัดในการเรียนรู้เทคนิคและหลักการใหม่ ๆ จากนักเขียนโปรแกรมรายอื่น

เทคนิค 6 ประการในการเริ่มเรียนเขียน โปรแกรมภาษา Python



Tips#5 – แสวงหาแหล่งความรู้ใหม่ ๆ อย่างสม่ำเสมอ

- ในปัจจุบันมีแหล่งความรู้ในการเขียนโปรแกรมทุกวิชาที่มีการอัปเดตตลอดเวลา
- ไม่จำเป็นต้องเรียนรู้ให้ครบจากทุกแหล่ง แต่เลือกจากแหล่งที่ตรงกับความสนใจหรือสามารถทำความเข้าใจได้



เทคนิค 6 ประการในการเริ่มเรียนเขียน โปรแกรมภาษา Python



Tips#6 – พั้กสมองบ้าง

- เป็นธรรมดaicในการพับเจอปัญหาในการเขียนโปรแกรมที่มีความซับซ้อนที่ไม่สามารถแก้ไขหรือหาคำตอบได้ในทันที
- พั้กสมอง เติมพลัง และเริ่มต้นทบทวนปัญหา/ความผิดพลาดการเขียนโปรแกรมใหม่





“A person who never made a
mistake never tried anything
new.”

End of Chapter Practice



- ทดลองเขียนโปรแกรมโดยใช้ Editor ที่ชอบ เช่น PyCharm Google Colab Jupyter Notebook หรืออื่นๆ
- Output ที่ต้องการ:

• 1 .
I am your father Star Wars 1980

• 2 .
I am your father-Star Wars-1980

• 3 .
I am your father
Star Wars
1980

Thanks!

Any questions?

