



Class period 2

บทที่ 2 ตัวแปรและโครงสร้างข้อมูลของภาษาไพธอน (ต่อ)

Function_Loop_Condition 1

ทบทวนคาบที่แล้ว



- `name = 'ภัทรวรรณ'`
- `surname = 'ใจเที่ยง'`
- `ID = '603021866-7'`
- `b= f'ชื่อ{name} นามสกุล{surname} รหัส{ID}'`
- `print(b)`



Function

- ทำหน้าที่รับ input มาประมวลผลออกมาเป็น output
- ยกตัวอย่างเปรียบเทียบฟังก์ชันในคณิตศาสตร์ $f(x) = y$ เช่น input ตัวแปร x เข้าไป f คือ function เพื่อให้ function ประมวลผลผลลัพธ์ output ออกมาคือ y

สัญลักษณ์ที่จำเป็นต้องใช้ในการเขียน Program



- backtick (`) ==> กด ~ ค้าง, alt - 9>6 (full keyboard with number)
- tilde (~)
- curly bracket ({ })
- square bracket ([])



Function template

- `def` คือการกำหนดฟังก์ชัน
- `def function_name(_Input_)` :
- `do something with _Input_ to get _Output_`
- `return _Output_`
- function มีส่วนสำคัญทั้งหมด 4 ส่วน
- บอก python ว่าเราจะเขียนฟังก์ชัน ชื่ออะไร `def function_name():` (ขาดไม่ได้)
- กำหนดตัวแปรที่จะเป็น input `_Input_` (ขาดได้)
- ส่วนประมวลผล `do something with _Input_ to get _Output_` (ขาดไม่ได้)
- ส่วน output `return _Output_` (ขาดได้)

การเว้นวรรค (indent) หรือการกด tab ก่อนพิมพ์บรรทัดถัดไปจาก `def` เพื่อบอกขอบเขตของโปรแกรม



ตัวอย่างการเขียน normal function

- `def print_name(surname, ID, name) :`
- `st = f'ชื่อ{name} นามสกุล{surname} รหัส{ID} '`
- `return st`
- ฟังก์ชันชื่อ `print_name`
- มี input 3 ตัวแปร คือ `surname, ID, name`
- ส่วนประมวลผล `st = f'ชื่อ{name} นามสกุล{surname} รหัส{ID} '` คือ ให้เขียน string โดยใช้ค่าในตัวแปรที่ input และเก็บไว้ในตัวแปร `st`
- Output ให้ `return st`



ตัวอย่างการเขียน normal function

- การใช้งานให้เขียนชื่อฟังก์ชันและค่า input ตามที่ def ไว้
- แบบที่ 1 `print_name('อินทระ', '64xxxxxx', 'ธนพงศ์')`
- ผลลัพธ์จะได้ 'ชื่อ ธนพงศ์ นามสกุล อินทระ รหัส 64xxxxxx'
- แบบที่ 2 `print(print_name(name='กาญจนา', surname='ประสาคุณ', ID='603021855-2'))`
- ผลลัพธ์จะได้ ชื่อ กาญจนา นามสกุล ประสาคุณ รหัส 603021855-2



ฟังก์ชันไม่จำเป็นต้องมี output หรือการ return

- `def print_name2(surname, ID, name) :`
- `st = f'ชื่อ{name} นามสกุล{surname} รหัส{ID} '`
- `print(st)` (สามารถใส่ `print` แทน `return` ได้)

- `print_name2(name='กาญจนา', surname='ประสาคุณ', ID='603021855-2')`

- ผลลัพธ์จะได้
- ชื่อ กาญจนา นามสกุล ประสาคุณ รหัส 603021855-2



ฟังก์ชันไม่จำเป็นต้องมี input

- `def Pi():`
- `return 3.14159265359`
- `Pi()*(2**2)` # หาพื้นที่ของวงกลมที่มีรัศมีเท่ากับ $Pi * r^2$
- ผลลัพธ์จะได้
- 12.56637061436
- ฟังก์ชันจำเป็นต้องมี 2 อย่าง คือ
- 1. ชื่อฟังก์ชัน():
- 2. เว้นวรรค (indent) ตามด้วยส่วนประมวลผล



การกำหนดค่า default ให้กับฟังก์ชัน

- input ของ function ใน python มี 2 แบบ
 - 1. input ที่จำเป็นต้องใส่
 - 2. input ที่ไม่จำเป็นต้องใส่ (มีค่า default)
 - เรียง input ที่ต้องการใส่ขึ้นก่อน
-
- `def print_2lines_default(name,surname,ID,grade='F') :`
 - `st = f'ชื่อ{name} นามสกุล{surname} รหัส{ID} '`
 - `print (st)`
 - `st2=f'เกรดวิชา Data Viz >>> {grade} '`
 - `print(st2)`
-
- `grade= 'F'` เป็นการกำหนดค่า default ให้ตัวแปร grade เป็น F



ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 1

- `print_2lines_default('ฉันทยาการต์', 'พวงมลัย', '613020551-8')`
- ผลลัพธ์จะได้
- ชื่อ ฉันทยาการต์ นามสกุล พวงมลัย รหัส 613020551-8
- เกรดวิชา Data Viz >>> F
- จะเห็นว่า ไม่มีการใส่ค่าตัวแปร grade ใน input แต่ผลลัพธ์ที่ได้ เกรดวิชา Data Viz >>> F
- เพราะในฟังก์ชันมีการกำหนดค่า default ให้กับตัวแปร grade เป็น F



ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 2

- `print_2lines_default('ฉันทยาการต์', 'พวงมลัย', '613020551-8', 'A')`
- ผลลัพธ์จะได้
- ชื่อ ฉันทยาการต์ นามสกุล พวงมลัย รหัส 613020551-8
- เกรดวิชา Data Viz >>> A
- สามารถ input ค่าตัวแปร grade แบบปกติได้

งานในห้อง กลับไปสร้างฟังก์ชันใน HW python101



- ให้สร้างฟังก์ชันคำนวณเวลาเป็นวินาทีของเวลาใดๆ และ print ออกมาให้สวยงาม

LOOP การวนซ้ำ



- `for each_member in listA :`
- `do_something`
- for เป็นคำที่ใช้บอก python ว่าเรากำลังเขียน loop โดย for จะวนถึงสมาชิกจาก listA มาทำ process `do_something`

ตัวอย่าง LOOP การวนซ้ำ



- `for i in [1,2,3] :`
- `o = i**2`
- `print (f'this member = {i} after process = {o}')`
- หมายความว่า ให้วนลูปอ่านค่าสมาชิกใน list [1,2,3] โดยแทนค่าสมาชิกที่อ่านด้วยตัวแปร i
- ภายในลูป นำตัวแปร i ยกกำลัง 2 เก็บค่าไว้ในตัวแปร o และ print string
- ผลลัพธ์จะได้
- this member = 1 after process = 1
- this member = 2 after process = 4
- this member = 3 after process = 9

Homework class period 2 กลับไป python101



3. ให้ สร้าง list ของ เวลา ['12:30:15','13:41:07',...] แล้ววนลูปเรียกฟังก์ชันคำนวณเวลาเป็น วินาที



Patterns for writing clean code in Python

- 1. Use long descriptive names that are easy to read.
- 2. Use descriptive intention revealing names.
- 3. Avoid using ambiguous shorthand.
- 4. Always use the same vocabulary.
- 5. Start tracking codebase issues in your editor.
- 6. Don't use magic numbers.
- 7. Be consistent with your function naming convention.
- 8. Functions should do one thing and do it well.
- 9. Do not use flags or Boolean flags.
- 10. Do not add redundant context.
- <https://www.freecodecamp.org/news/how-to-write-clean-code/>
- <https://dev.to/alexomeyer/10-must-know-patterns-for-writing-clean-code-with-python-56bf>