



CSS 112 Computer Programming

Lecture II: Variables, Data Types, and Expressions

Suvil Chomchaiya (Ph.D.)



Contents



- Variables and Reserved Word in Python
- Python's Data Types
- Python's Expression





Variables and Reserved Word in Python



What is Python Variables



- ในการเขียนโปรแกรมในภาษา Python ไม่จำเป็นต้องประกาศตัวแปรก่อนใช้งาน
- สามารถกำหนดค่าขึ้นมา และเรียกใช้ได้ทันที
- การตั้งชื่อตัวแปรของ ต้องเป็นไปตามกฎที่ถูกต้อง มีจะนั่นจะมีผลต่อการทำงานของโปรแกรม



Python Variables Naming Rules



- การตั้งชื่อตัวแปรใน Python ต้องเป็นไปตามกฎดังนี้
 - ขึ้นต้นด้วยตัวอักษรตัวแรกแต่ 1 ตัวอักษรขึ้นไป
 - ห้ามมีช่องว่าง
 - ห้ามมีสัญลักษณ์พิเศษ เช่น #, ?, \$, ...
 - ต้องไม่ซ้ำกับคำส่วน (Reserved Words)
 - ตัวพิมพ์เล็กพิมพ์ใหญ่ มีความหมายต่างกัน (เป็น case-sensitive)



Python's Data Types





Python Data Types

ชนิดข้อมูลพื้นฐานที่มากับ Python

1. **Numbers** : ข้อมูลตัวเลข
2. **Boolean** : ข้อมูลค่าความจริง
3. **Sequence** : ข้อมูลเรียงลำดับ
 - **String** เก็บข้อมูลตัวอักษร
 - **List** เก็บข้อมูลได้มากกว่า 1 ค่าใน 1 ตัวแปร หรือที่เรียกว่า **compound type**
 - **Tuple** อ่านว่า "ทูเพิล" เก็บข้อมูลได้มากกว่า 1 ค่าใน 1 ตัวแปร ใช้สำหรับเก็บลำดับ หรือที่เรียกว่า
4. **Set** : ข้อมูลที่เรียงต่อ กันคั่นด้วยเครื่องหมายคอมมา (,) ภายในเครื่องหมายวงเล็บปีกกา { }
5. **Dictionary** : เก็บข้อมูลได้มากกว่า 1 ค่าใน 1 ตัวแปร เช่น กัน หรือที่เรียกว่า **table type** ที่มี **KM** บน **UN**
ได้กับตัวแปร **array** ใน **php**

Python Data Types: Numbers



- ข้อมูลประเภท **Numbers** จะแบ่งออกเป็น
 - Integer
 - Float
 - Decimal
 - Complex
- ในการเขียนโปรแกรมจะเน้นย้ำใน **Integer** ซึ่งเป็นการเก็บข้อมูลแบบจำนวนเต็ม และ **Float** เป็นข้อมูลแบบจำนวนจริง
- ข้อมูลแบบ **Decimal** นั้นแตกต่างไปจาก **Float** คือสามารถเก็บความละเอียดของจุดทศนิยมได้มากกว่า

Python Data Types: Numbers



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Number ในภาษา Python

```
# Integer
```

```
a = 7
```

```
b = 3
```

```
c = a + b
```

```
d = a / b
```

```
print ('a = %d' % a)
```

```
print ('b = %d' % b)
```

```
print ('c = %d' % c)
```

```
print ('d = %d')
```

```
a = 7
```

```
b = 3
```

```
c = 10
```

```
d = 2.3333333333333335
```

ผลลัพธ์การทำงาน

*ในกรณีแสดงผลลัพธ์ในรูปแบบของ **String format** กับพังก์ชัน **print()** นั้นจะใช้ specifier เป็น %d



Python Data Types: Numbers



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Number ในภาษา Python

```
# Floating point number
```

```
speed = 34.12
```

```
pi = 22 / 7
```

```
height = 2.31E5
```

```
length = 1.3E-3
```

```
print ('speed = %f' % speed)
```

```
print ('pi = %f' % pi)
```

```
print ('height = %f' % height)
```

```
print ('length = %f' % length)
```

```
print (pi)
```

```
speed = 34.120000
pi = 3.142857
height = 231000.000000
length = 0.001300
3.142857142857143
```



ผลลัพธ์การทำงาน

ในการแสดงผลของข้อมูลประเภท **Float** กับการจัดรูปแบบของตัวเลขนั้นจะใช้ **%f**

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Numbers



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Number ในภาษา Python

```
# Floating point number
```

```
speed = 34.12
```

```
pi = 22 / 7
```

```
height = 2.31E5
```

```
length = 1.3E-3
```

```
print ('speed = %f' % speed)
```

```
print ('pi = %f' % pi)
```

```
print ('height = %f' % height)
```

```
print ('length = %f' % length)
```

```
print (pi)
```

```
speed = 34.120000
pi = 3.142857
height = 231000.000000
length = 0.001300
3.142857142857143
```



ผลลัพธ์การทำงาน

ในการแสดงผลของข้อมูลประเภท **Float** กับการจัดรูปแบบของตัวเลขนั้นจะใช้ **%f**

Python Data Types: Boolean



- เป็นข้อมูลที่มีค่าเป็นจริงหรือเท็จเท่านั้น โดยจะแทนด้วย `True` และ `False`
- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Boolean ในภาษา Python
 - `A=9>5`
 - `B=9-5>9+5`
 - `C=true`
 - `D=False`

```
A=9>5
B=9-5>9+5
C=True
D=False

print(A)
print(B)
print(C)
print(D)

→ True
    False
    True
    False
```

ผลลัพธ์การพิมพ์

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: String



- **Strings** เป็นประเภทข้อมูลที่สำคัญและใช้งานทั่วไปในการเขียนโปรแกรม
- ในภาษา Python เช่นกัน **String** เป็นลำดับของตัวอักษรหลายตัวเรียงต่อกัน
- ในภาษา Python นั้น **String** จะอยู่ในเครื่องหมาย Double quote หรือ Single quote เท่านั้น



Python Data Types: Sequence: String



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Number ในภาษา Python

```
name = "Mateo"  
country = "Ukrain"  
language = 'Python'  
interest = 'Mountain Everest'
```



- การประกาศตัวแปรประเภท String สองตัวแปรเราเป็นการประดิษฐ์ใช้ Double quote และสองตัวแปรต่อมาเป็นการใช้ Single quote ซึ่งสามารถใช้แบบไหนก็ได้

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: String



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ String ในภาษา Python

```
sentent1 = "What's your name?"
```

```
sentent2 = 'I\'m Mateo.'
```

```
sentent3 = "He said \"I would learn Python first\"."
```

```
sentent4 = 'His teach replied "Oh well!"'
```

```
print (sentent1)
```

```
print (sentent2)
```

```
print (sentent3)
```

```
print (sentent4)
```



Python Data Types: Sequence: String

String

ในตัวอย่างใน slide ที่ผ่านมา การประกาศ String ทั้งสองแบบกับ Escape character ตัวอักขระ ' และ "

- การใช้งานตัวอักขระเหล่านี้ เราจะต้องทำการใส่เครื่องหมาย \ ลงไปข้างหน้าเสมอ
- ในภาษา Python จะมีตัวอักขระพิเศษ หรือที่เรียกว่า Escape Character ซึ่งตัวอักขระเหล่านี้ได้แก่ \ (Backslash), ' (Single quote) หรือ " (Double quote) เป็นต้น ซึ่งใช้กับ Whitespace characters เช่น
 - \t สำหรับการย่อหน้า
 - \n สำหรับการขึ้นบรรทัดใหม่



DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: String



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ String ในภาษา Python

```
sentent1 = "What's your name?"  
sentent2 = 'I\'m Mateo.'  
sentent3 = "He said \"I would learn Python first\"."  
sentent4 = 'His teach replied "Oh well!"'  
print (sentent1)  
print (sentent2)  
print (sentent3)  
print (sentent4)
```

```
What's your name?  
I'm Mateo.  
He said "I would learn Python first".  
His teach replied "Oh well!"
```



ผลลัพธ์การทำงาน

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: Lists



- List เป็นประเภทข้อมูลที่เก็บข้อมูลแบบเป็นชุดและลำดับ
- List สามารถเก็บข้อมูลได้หลายค่าในตัวแปรเดียว และมี Index สำหรับเข้าถึงข้อมูล ในภาษา Python นั้น
- List จะเป็นเหมือนอารเรย์ (Array) ในภาษา C



DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: Lists



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ List ในภาษา Python

```
# Declare lists
```

```
numbers = [1, 2, 4, 6, 8, 19]
```

```
names = ["Mateo", "Danny", "James", "Thomas", "Luke"]
```

```
mixed = [-2, 5, 84.2, "Mountain", "Python"]
```

```
# Display lists
```

```
print(numbers)
```

```
print(names)
```

```
print(mixed)
```

```
[1, 2, 4, 6, 8, 19]  
['Mateo', 'Danny', 'James', 'Thomas', 'Luke']  
[-2, 5, 84.2, 'Mountain', 'Python']
```

ผลลัพธ์การทำงาน

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: List



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ List ในภาษา Python

```
languages = ["C", "C++", "Java", "Python", "PHP"]
```

```
print("Index 0 = ", languages[0])
print("Index 3 = ", languages[3])
languages[0] = "Scalar"
print("Index 0 = ", languages[0])
```

```
Index 0 = C
Index 3 = Python
Index 0 = Scalar
```

ผลลัพธ์การทำงาน



DJ1

Dr. Jard, 8/16/2022

Python Data Types: Sequence: Tuple



- เป็นข้อมูลที่เรียงต่อกันภายในใช้เครื่องหมายวงเล็บ หรือ ไม่ต้องมีวงเล็บครอบได้ และ คั่นด้วยเครื่องหมายคอมมา (,)
- ลักษณะจะเหมือนกับ List แต่จะไม่สามารถ เพิ่มเปลี่ยนแปลงแก้ไข หรือลบข้อมูล ใน Tuple ได้



Python Data Types: Sequence: Tuple



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Tuple ในภาษา Python

#ตัวอย่างข้อมูลประเภท Tuple

```
tuple_a = (1, 2, 3, 4, 5)
```

```
tuple_b = 'a', 'b', 'c', 'd', 'e'
```

```
tuple_c = ('python', True, 10, 4, 1.5)
```

```
tuple_d = () #empty tuple
```

```
#ตัวอย่างข้อมูลประเภท Tuple
tuple_a = (1, 2, 3, 4, 5)
tuple_b = 'a', 'b', 'c', 'd', 'e'
tuple_c = ('python', True, 10, 4, 1.5)
tuple_d = () #empty tuple

print(tuple_a)
print(tuple_a[2])
print(tuple_c[0])
print(tuple_d)

(1, 2, 3, 4, 5)
3
python
()
```

ผลลัพธ์การทำงาน

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Set



- เป็นข้อมูลที่เรียงต่อกันคั่นด้วยเครื่องหมายคอมมา (,) ภายในเครื่องหมายวงเล็บปิด กก { }
- สามารถเพิ่ม แก้ไข ลบข้อมูลและเป็นค่าว่างได้
- การจัดเก็บข้อมูลจะไม่มีลำดับ และจะไม่เก็บข้อมูลที่ซ้ำซ้อนกัน
- ไม่สามารถเข้าถึงข้อมูลด้วยตำแหน่งข้อมูล (index)



Python Data Types: Sequence: Tuple



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Set ในภาษา Python

```
# ตัวอย่างข้อมูลประเภท Set
```

```
set_a = {1,2.5,3,'Thailand'}
```

```
set_b = {1,2.5,3,True,'Thailand'}
```

```
set_c = {1,2.5,3,'Thailand','Thailand'}
```

```
set_d = {}
```

```
## ตัวอย่างข้อมูลประเภท Set
set_a = {1,2.5,3,'Thailand'}
set_b = {1,2.5,3,True,'Thailand'}
set_c = {1,2.5,3,'Thailand','Thailand'}
set_d = {}
```

```
print(set_a)
print(set_b)
print(set_c)
print(set_d)
```

```
{1, 2.5, 3, 'Thailand'}
{1, 2.5, 3, 'Thailand'}
{1, 2.5, 3, 'Thailand'}
{}}
```

ผลลัพธ์การทำงาน

DJ1

Dr. Jard, 8/16/2022

Python Data Types: Dictionary



- เป็นข้อมูลที่เรียงต่อ กัน คัน ด้วย เครื่องหมาย คอมมา (,) ภายใน เครื่องหมาย วง เล็บ ปีก กา { }
- อยู่ในรูปแบบ ของ **key:value**



DJ1

Dr. Jard, 8/16/2022

Python Data Types: Dictionary



- ตัวอย่างในการประกาศและใช้งานตัวแปรแบบ Dictionary ในภาษา Python



```
# ตัวอย่างข้อมูลประเภท Dictionary
```

```
dict_salary = {'Peter': 150000, 'Stark':200000, 'Thor':0}  
dict_salary['Peter']  
dict_salary['Stark']  
dict_salary['Thor']
```



DJ1

Dr. Jard, 8/16/2022

Python's Expression



Expressions in Python



- **Expression** หรือ นิพจน์ คือการทำงานร่วมกันระหว่างตัวแปร (หรือค่าคงที่) และตัวดำเนินการ ภายใต้ตัวดำเนินการ (**Operators**) สำหรับควบคุมการทำงาน
- ในภาษา Python นั้นมี **Expression** อよງ 2 แบบ คือ
 - **Boolean expression** เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการเปรียบเทียบค่าหรือตัวดำเนินการตรวจสอบตรี และจะได้ผลลัพธ์เป็น **Boolean**
 - **Mathematical (หรือ Non-Boolean) Expression** เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการคณิตศาสตร์ และจะได้รับค่าใหม่เป็นตัวเลขหรือค่าที่ไม่ใช่ **Boolean**



DJ1

Dr. Jard, 8/16/2022

Expressions in Python

- ตัวอย่างของ Expressions ในภาษา Python

```
a = 4
```

```
b = 5
```

```
# Boolean expressions
```

```
print(a == 4)
```

```
print(a == 5)
```

```
print(a == 4 and b == 5)
```

```
print(a == 4 and b == 8)
```

```
# Non-boolean expressions
```

```
print(a + b)
```

```
print(a + 2)
```

```
print(a * b)
```

```
print((a * a + b * b) / 2)
```

```
print("Python" + "Language")
```

```
True
```

```
False
```

```
True
```

```
False
```

```
9
```

```
6
```

```
20
```

```
20.5
```

```
Python Language
```

ผลลัพธ์การทำงาน

Conclusion



- ในการเขียนโปรแกรมในภาษา Python ไม่จำเป็นต้องประกาศตัวแปรก่อนใช้งาน สามารถกำหนดค่าขึ้นและเรียกใช้ได้ทันที แต่การตั้งชื่อตัวแปรของ ต้องเป็นไปตามกฎที่
KMUTT กำหนด
- ชนิดข้อมูลพื้นฐานที่มากับ Python 5 ประเภท คือ Numbers Boolean Sequence Set และ Dictionary
- ลำดับความสำคัญของตัวดำเนินการใน Python Operators Precedence เป็นการจัดลำดับความสำคัญให้กับตัวดำเนินการในนิพจน์ (Expression)



Thanks!

Any questions?

