

# **Lecture 1**

# **Python Basic**

**Jaeyun Kang**

# Python Introduction



python

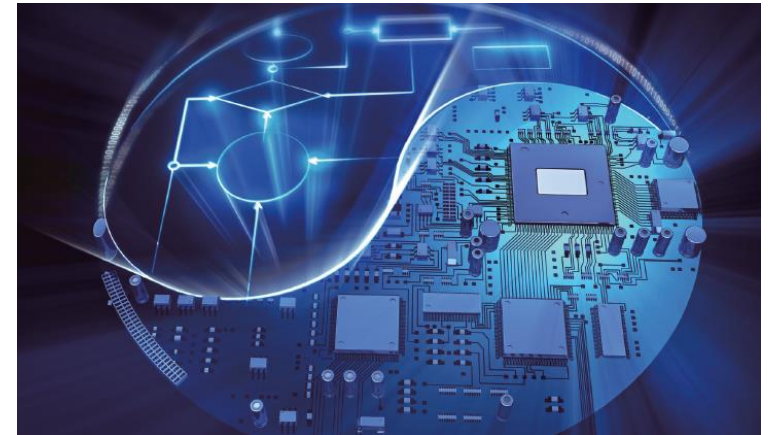
# Programming Language

- **Common expression & grammar**
- **Distinct philosophy**



C

C



# C++



# HARD



# JAVA



Java™

Normal



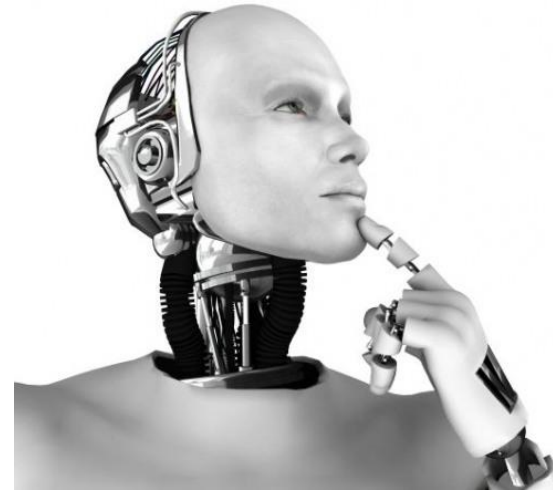
spring

Java



Java

# Python



# Python Basic - Print

```
print ("Hello, World")
```

```
print (5.3)
```

```
print (1 + 3)
```

```
print ('a')
```

```
print ('b')
```

```
print ('a', 'b')
```

```
# print ('a') // tip: comment
```



# Python Basic - Variables and Assigning

```
a = 1  
print (a)
```

```
var0 = "Hello"
```

```
var1 = 10  
var2 = 5
```

```
a = b = "Python"  
print (a)  
print (b)
```

```
a = "Python"  
b = a  
print (b)
```

# Python Basic - Variables and Assigning

**a, b** = ('python', 'life')

**(a, b)** = 'python', 'life'

**[a, b]** = ['python', 'life']

**a** = 3

**b** = 5

**a, b** = **b, a** // **a = 5, b = 3**

# Python Basic - Variables

- What can be variable names?

- Combination of

**alphabets** a, b, c, A, B, C,...

**digits** 1, 2, 3,...

**underbar** \_

- Starts with lower case
- Should contain meaning

**Var**(x), **var**(o)

**a**(x), **b**(x), **numOfStudents**(o), **num\_of\_students**(o)

# Python Basic - Data Type Number

- Integer

**a = 123**

**a = -178**

**a = 0**

- Floating-point

**a = 1.2**

**a = -3.45**

# Python Basic - Data Type Number

- Fundamental operations
- Plus : **+** , Minus : **-** , Multiply : **\*** , Divide : **/**

```
a = 7
```

```
b = 2
```

```
print (a + b) // 9
```

```
print (a - b) // 5
```

```
print (a * b) // 14 ( print (a ** b) // 49 )
```

```
print (a / b) // 3.5 ( print (a // b) // 3 )
```

```
print (a % b) // 1
```

# Python Basic - Data Type Number

```
a = 2  
b = a * 2  
print(b) // 4
```

```
a = 5  
a = a + 1  
print(a) // 6
```

```
a = 5  
a += 1 ( a = a $ 3 -> a $= 3)  
print(a) // 6
```

```
c = 10  
c *= 2  
print(c) // 20
```

# Python Basic - Data Type String

**a** = "KUSITMS is alcohol"

**a** = 'KUSITMS is alcohol'

**a** = """KUSITMS is alcohol"""

**a** = '''KUSITMS is alcohol'''

**a** = 'Python's favorite food is perl!' (x)

**a** = "Python's favorite food is perl!" (o)

**a** = 'Python~~w~~'s favorite food is perl!' (o)

# Python Basic - Data Type String

Life is too short  
You need python

```
a = "Life is too short\nYou need python"  
print (a)
```

```
a = """  
    Life is too short  
    You need python  
    """  
print (a)
```



# Python Basic - Data Type String

- String operations

```
head = "Python"
```

```
tail = " is fun"
```

```
print (head + tail) // Python is fun
```

```
a = "Python"
```

```
print (a * 2) // PythonPython
```

```
print ("=" * 10) // =====
```

```
print ("Python" + 2) // error
```

```
print ("Python" + str(2)) // Python2
```

# Python Basic - Data Type String

- String Indexing

**a** = "Life is too short. You need Python"

```
print (a[0]) // L
```

```
print (a[1]) // i
```

```
print (a[2]) // f
```

```
print (a[3]) // e
```

```
print (a[4]) // ' '
```

```
print (a[5]) // i
```

```
...
```

```
print (a[-1]) // n
```

```
print (a[-2]) // o
```

```
print (a[-0]) // L
```

# Python Basic - Data Type String

- String Slicing

**a** = "Life is too short, You need Python"

- How to print Life?

```
print (a[0]+a[1]+a[2]+a[3]) // Life
```

```
print (a[0:4]) // Life
```

```
print (a[0:3]) // Lif
```

```
print (a[5:7]) // is
```

```
print (a[19:]) // You need Python
```

```
print (a[:17]) // Life is too short
```

```
print (a[:]) // Life is too short, You need Python
```

```
print (a[19:-7]) // You need
```

# Python Basic - Data Type String

- String Slicing

```
a = "20160930Rainy"
```

```
data = a[:8]
```

```
weather = a[8:]
```

```
print(data) // 20160930
```

```
print(weather) // Rainy
```

```
word= "Pithon"
```

```
word[1] = 'y' // error
```

```
newWord = word[:1] + 'y' + word[2:]
```

```
print(newWord) // Python
```

# Python Basic - Data Type String

- String Formatting

```
a = "I eat %d apples." % 3  
print (a) // I eat 3 apples.
```

```
a = "I eat %s apples." % "five"  
print (a) // I eat five apples.
```

```
number = 3  
a = "I eat %d apples." % number  
print (a) // I eat 3 apples.
```

# Python Basic - Data Type String

- String Formatting

```
number = 10
```

```
day = "three"
```

```
a = "I ate %d apples. So I was sick for %s  
days." %(number, day)
```

```
// I ate 10 apples. So I was sick for three days.
```

**%s: String, %c: character, %d: Integer, %f: floating-point**

# Python Basic - Data Type String (Functions)

```
a = "hobby"  
print (a.count('b')) // 2
```

```
a = "Python is best choice"  
print (a.find('b')) // 10  
print (a.find('k')) // -1
```

```
a = ","  
a.join('abcd') // a = "a,b,c,d"
```

# Python Basic - Data Type String (Functions)

```
a = "hi"
```

```
a.upper() // a = "HI"
```

```
a = "HI"
```

```
a.lower() // a = "hi"
```

```
a = " hi "
```

```
a.strip() // a = "hi"
```



# Python Basic - Data Type List

- `List_name=[a1, a2, ...]`

`odd = [1, 3, 5, 7, 9]`

`a = []`

`b = [1, 2, 3]`

`c = ['Life', 'is', 'too', 'short']`

`d = [1, 2, 'Life', 'is']`

`e = [1, 2, ['Life', 'is']]`

# Python Basic - Data Type List

- List indexing

```
a = [1, 2, 3]
```

```
print (a) // [1, 2, 3]
```

```
print (a[0]) // 1
```

```
print (a[1]) // 2
```

```
print (a[2]) // 3
```

```
print (a[0] + a[2]) // 4
```

```
print (a[-1]) // 3
```

# Python Basic - Data Type List

- List indexing

```
a=[1, 2, 3, ['a', 'b', 'c']]
```

```
print (a[0]) // 1
```

```
print (a[-1]) // ['a', 'b', 'c']
```

```
print (a[-1][0]) // a
```

```
print (a[-1][1]) // b
```

# Python Basic - Data Type List

- List Slicing

```
a=[1, 2, 3, 4, 5]
```

```
print (a[0:2]) // [1, 2]
```

```
print (a[:2]) // [1, 2]
```

```
print (a[2:]) // [3, 4, 5]
```

```
a = [1, 2, 3, ['a', 'b', 'c'], 4, 5]
```

```
print (a[2:5]) // [3, ['a', 'b', 'c'], 4]
```

```
print (a[3][:2]) // ['a', 'b']
```

# Python Basic - Data Type List

- List Operation

```
a = [1, 2, 3]
```

```
b = [4, 5, 6]
```

```
print (a + b) // [1, 2, 3, 4, 5, 6]
```

```
a = [1, 2, 3]
```

```
print (a * 3) // [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

# Python Basic - Data Type List

- List Modification

```
a = [1, 2, 3]
```

```
a[2] = 4
```

```
print(a) // [1, 2, 4]
```

```
a = [1, 2, 3]
```

```
a[1:2] = ['a', 'b', 'c']
```

```
print(a) // [1, 'a', 'b', 'c', 3]
```

```
a = [1, 2, 3]
```

```
a[1] = ['a', 'b', 'c']
```

```
print(a) // [1, ['a', 'b', 'c'], 3]
```

# Python Basic - Data Type List

- List Deletion

```
a = [1, 'a', 'b', 'c', 3]
a[1:3] = []
print (a) // [1, 'c', 3]
```

```
a = [1, 'c', 3]
del a[1]
print (a) // [1, 3]
```

```
a = [1, 2, 3]
del a[1:3]
print (a) // [1]
```

# Python Basic - Data Type List (Functions)

```
a = [1, 2, 3]
```

```
a.append(4) // a = [1, 2, 3, 4]
```

```
a.append([5, 6]) // a = [1, 2, 3, 4, [5, 6]]
```

```
a = [1, 4, 3, 2]
```

```
a.sort() // a = [1, 2, 3, 4]
```

```
a = ['a', 'c', 'b']
```

```
a.sort() // a = ['a', 'b', 'c']
```



# Python Basic - Data Type List (Functions)

```
a = [1, 2, 3, 4]
```

```
a.reverse() // a = [4, 3, 2, 1]
```

```
a = [1, 2, 3]
```

```
print (a.index(3)) // 2
```

```
print (a.index(1)) // 0
```

```
print (a.index(0)) // error
```

# Python Basic - Data Type List (Functions)

```
a = [1, 2, 3]
```

```
a.insert(0, 4) // a = [4, 1, 2, 3]
```

```
a.insert(3, 5) // a = [4, 1, 2, 5, 3]
```

```
a = [1, 2, 3, 1, 2, 3]
```

```
a.remove(3) // a = [1, 2, 1, 2, 3]
```

```
a.remove(3) // a = [1, 2, 1, 2]
```

# Python Basic - Data Type List (Functions)

```
a = [1, 2, 3]
```

```
a.pop() // a.pop() = 3, a = [1, 2]
```

```
a = [1, 2, 3]
```

```
a.pop(1) // a.pop(1) = 2, a = [1, 3]
```

```
a = [1, 2, 3, 1]
```

```
print (a.count(1)) // 2
```

# Python Basic - Data Type Tuple

```
t1 = ()
```

```
t2 = (1,)
```

```
t3 = (1, 2, 3)
```

```
t4 = 1, 2, 3
```

```
t5 = ('a', 'b', ('ab', 'cd'))
```

```
t1 = (1, 2, 'a', 'b')
```

```
del t1[0] // error
```

```
t2 = (1, 2, 'a', 'b')
```

```
t2[0] = 'c' // error
```

Indexing / Slicing / Operations are supported like list

# Python Basic - Data Type Dictionary

**{Key1: Value1, Key2: Value2, Key3: Value3 ...}**

**dic = {'name': 'pey', 'phone': '01068631342', 'birth': '1118'}**

**a = {1: 'hi'}**

**a = {'a': [1, 2, 3]}**

# Python Basic - Data Type Dictionary

```
a = {1: 'a'}
```

```
a[2] = 'b' // a = {2: 'b', 1: 'a'}
```

```
a['name'] = 'pey' // a = {'name': 'pey', 2: 'b', 1: 'a'}
```

```
a[3] = [1, 2, 3] // a = {'name': 'pey', 3: [1, 2, 3], 2: 'b', 1: 'a'}
```

```
del a[1] // a = {'name': 'pey', 3: [1, 2, 3], 2: 'b'}
```

# Python Basic - Data Type Dictionary

```
a = {1: 'a', 2: 'b'}
```

```
print (a[1]) // a
```

```
print (a[2]) // b
```

```
grade = {'pey': 10, 'Juliet': 99}
```

```
print (grade[pey]) // error
```

```
print (grade['pey']) // 10
```

```
print (grade['Juliet']) // 99
```

# Python Basic - Data Type Dictionary

Key should be distinct value

```
a = {1: 'a', 1: 'b'}  
print (a) // {1: 'b'}
```

Cannot use lists for keys

```
a = {[1, 2]: 'hi'} // error
```



# Python Basic - Data Type

## True & False

- There is 'Boolean' data type

```
a = True
```

```
b = False (Capital)
```

```
print a
```

- Boolean operator : and, or, not

```
not True // False
```

```
not False // True
```

```
True and True // True
```

```
True and False // False
```

```
False and False // False
```

```
True or True // True
```

```
True or False // True
```

```
False or False // False
```

# Python Basic - Data Type

## True & False

**a** = False

**b** = True

```
print (not (a and b)) // True
```

- Boolean operator priority : ( ) > not > and > or

**c** = not False or True and False

```
print (c) // True
```

# Python Basic - Data Type True & False

- Comparing Operator : >, <, >=, <=, ==, !=

```
a = 2 ; b = 3  
print (a > b) // False  
print (a < b) // True
```

```
a = 2 ; b = 2  
print (a == b)  
print (a != b)
```

```
a = 5  
print (a >= 3) // True  
print (a == 5) // True  
c = (a > 6)  
print (c) // False
```

# Python Basic - Data Type True & False

- Other Data type also can be True or False

## String

True: "python"

False: ""

## List

True: [1,2,3]

False: []

## Number

True: not 0

False: 0

# Python Basic - Control Statement

## if, else, elif

```
if (Boolean condition):  
    statement1  
    ...
```

```
else:  
    statement2  
    ...
```

```
money = 9000  
if money > 10000:  
    print ("taxi")  
else:  
    print ("bus") // bus
```

# Python Basic - Control Statement

## if, else, elif

```
a = 17
```

```
if a % 2 == 0:
```

```
    print ("Even")
```

```
else:
```

```
    print ("Odd") // Odd
```

```
a = 20
```

```
if a > 10 and a != 30:
```

```
    print ("yes") // yes
```

```
else:
```

```
    print ("no")
```

# Python Basic - Control Statement

## if, else, elif

```
if 1 in [1, 2, 3]:  
    print ("yes") // yes  
else:  
    print ("no")
```

```
if 'j' not in 'python':  
    print ("yes") // yes  
else:  
    print ("no")
```

# Python Basic - Control Statement

## if, else, elif

```
a = 20
```

```
if a > 10 and a != 30:
```

```
    pass // Nothing happened
```

```
else:
```

```
    print ("no")
```

```
a = 17
```

```
if a % 2 == 0: print ("Even")
```

```
else: print ("Odd") // Odd
```



# Python Basic - Control Statement

## if, else, elif

```
a = 10
if a > 5
    print ("yes")
else
    print ("no") // Error
```

```
a = 10
if a > 5:
    print ("yes")
else:
    print ("no") // Error
```

Notice : Don't forget ' :' & Tap

# Python Basic - Control Statement

## if, else, elif

```
if (Boolean condition):  
    statement  
    ...
```

```
elif (Boolean condition):  
    statement  
    ...
```

```
elif (Boolean condition):  
    statement  
    ...
```

```
else:  
    statement  
    ...
```

# Python Basic - Control Statement

## if, else, elif

```
grade = 76
if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C") // C
elif grade >= 60:
    print("D")
else:
    print("F")
```

# Python Basic - Control Statement

## if, else, elif

```
grade = 86
if grade >= 90:
    if grade >= 95:
        print("A+")
    else:
        print("A0")
elif grade >= 80:
    if grade >= 85:
        print("B+") // B+
    else:
        print("B0")
else:
    print("F")
```

# Python Basic - Control Statement

## while

```
while (Boolean condition):  
    statement1  
    statement2  
    ...
```

```
i = 1  
while i <= 10:  
    print (i)  
    i = i + 1 // 1 2 3 4 5 6 7 8 9 10
```

Also don't forget `:` & Tab

# Python Basic - Control Statement

## while

```
a = 1
while a < 10:
    a = a * 2
print (a) // 16
```

```
a = ['x', 'y', 'z', 'w']
i = 0
while i < 4:
    print (a[i])
    i = i + 1 // x y z w
```

# Python Basic - Control Statement while (Continue & Break)

```
a = 1
```

```
while True:
```

```
    a = a + 1
```

```
    if a >= 100:
```

```
        break
```

```
print (a) // 100
```

```
lis = ['a', 'b', 'c', 'd']
```

```
i = -1
```

```
while i < 3:
```

```
    i = i + 1
```

```
    if lis[i] == 'c':
```

```
        continue
```

```
    print (lis[i]) // a b d
```

# Python Basic - Control Statement for

```
for (variable) in (list_name):  
    statement1  
    statement2  
    ...
```

```
test_list = ['one', 'two', 'three']  
for i in test_list:  
    print (i) // one two three
```



# Python Basic - Control Statement for

```
marks = [90, 25, 67, 45, 80]
```

```
number = 0
```

```
for mark in marks:
```

```
    number = number + 1
```

```
    if mark >= 60:
```

```
        print (str(number) + " student is passed")
```

```
    else:
```

```
        print (str(number) + " student is failed")
```

```
// 1 student is passed
```

```
// 2 student is failed
```

```
// 3 student is passed ...
```

# Python Basic - Control Statement for

```
sum = 0
```

```
for i in range(1, 11):
```

```
    sum = sum + i
```

```
print (sum) // 55
```

```
for i in range(5):
```

```
    print (i) // 0 1 2 3 4
```

# Python Basic - Control Statement for

```
marks = [90, 25, 67, 45, 80]
for number in range(len(marks)):
    if marks[number] < 60: continue
    print (str(number+1) + " student is passed")
// 1 student is passed
// 3 student is passed
// 5 student is passed
```

```
for i in range(2, 10):
    for j in range(1, 10):
        print (i * j, end = " ")
    print ("") // ?
```

# Exercise 1

Print below pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

# Exercise 2

Calculate average of student's score

[70, 60, 55, 75, 95, 90, 80, 80, 85, 100]

Answer: 79.0

# Exercise 3

Print all prime number  $p \leq 100$

Prime number: 2, 3, 5, 7, 11, 13, 17...

```
C:\Python27\python.exe C:/Users/Jaeyun/PycharmProjects/myFirstProject/test.py
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

```
Process finished with exit code 0
```

# Exercise 4

**Print all twin-prime number pair (p1, p2)  $p2 \leq 100$**

**twin-prime number (p1, p2):**

**both p1, p2 are prime numbers and  $p2 = p1 + 2$**

**twin-prime number: (3, 5), (5, 7), (11, 13), (17, 19) ...**

(3, 5)

(5, 7)

(11, 13)

(17, 19)

(29, 31)

(41, 43)

(59, 61)

(71, 73)

# Exercise 5

Make program which print binary notation of number 925  
(925 is in decimal notation)

Ex) binary notation of number 3 : 11

Ex) binary notation of number 5 : 101

Ex) binary notation of number 7 : 111

Ex) binary notation of number 17 : 10001

Hint:

```
a = []
```

```
a.append(3) // a = [3]
```

```
a.append(4) // a = [3, 4]
```

```
a.reverse() // a = [4, 3]
```



**Question**