Python tutorial #1

본 페이지는 한림대학교 710231(딥러닝이해및응용) 수업에서 학생들의 Python 학습을 위해 만든 페이지입니다.

Hello World !

In [1]:

```python
print('Hello World')
print('Hello World {} + {} = {}'.format(2, 3, 2+3))
```

```
Hello World
Hello World 2 + 3 = 5
```

Basic data types

In [2]:

```python
x = 3
print(type(x)) # Prints "<class 'int'>"
print(x)       # Prints "3"
```

```
<class 'int'>
3
```

In [3]:

```python
print(x + 1)   # Addition; prints "4"
print(x - 1)   # Subtraction; prints "2"
print(x * 2)   # Multiplication; prints "6"
print(x ** 2)  # Exponentiation; prints "9"
```

```
4
2
6
9
```

For statement

range: 영역을 설정

```
A = range(5)
print(A)
```

range(0, 5)

- A의 세번째 요소를 출력

```
print(A[2])
```

2

```
for i in range(5):
    #print(i, A[i])
    print('{} ----- {}'.format(i, A[i]))
```

```
0 ----- 0
1 ----- 1
2 ----- 2
3 ----- 3
4 ----- 4
```

Excersise

구구단을 작성하시오 (아래 코드를 수정하시오)

```
for i in range(9):
    print('{} x {} = {}'.format(2, i, 2*i))
```

```
2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
```

Operators

```python
print((1, 2, 3) * 3)
print([1, 2, 3] * 3)
print("Hello "*3)
```

```
(1, 2, 3, 1, 2, 3, 1, 2, 3)
[1, 2, 3, 1, 2, 3, 1, 2, 3]
Hello Hello Hello
```

Containers

Python includes several built-in container types: lists, dictionaries, sets, and tuples.

Tuple

A simple immutable (변경할 수 없는, 불변의) ordered sequence of items

```python
# -*- coding: utf-8 -*-
# creating a tuple
months = ('January','February','March','April','May','June',₩
'July','August','September','October','November','December')

print(months[0])
print("index of 7 ==> " , months[7])
```

```
January
index of 7 ==>  August
```

하나씩 출력하기

```
# iterate through them:
for item in months:
    print (item)
```

```
January
February
March
April
May
June
July
August
September
October
November
December
```

```
t = ('john', 32, (2,3,4,5), 'hello')
print(t)
print(t[2])
print(t[2][1])
print(t[:2]) # index 포함 X
print(t[2:]) # index 포함 O

print(t[-1])
print(t[-2])
```

```
('john', 32, (2, 3, 4, 5), 'hello')
(2, 3, 4, 5)
3
('john', 32)
((2, 3, 4, 5), 'hello')
hello
(2, 3, 4, 5)
```

List

Mutable (바꿀수 있는, 변경가능한) ordered sequence of items of mixed types

```python
li = ['hallym', 1, 3.141572, 'hello']
print(li)
li[1] = 45
print(li)
li.append('September')
print(li)
```

```
['hallym', 1, 3.141572, 'hello']
['hallym', 45, 3.141572, 'hello']
['hallym', 45, 3.141572, 'hello', 'September']
```

- 비어있는 리스트 만들기

```python
v = []
```

- 비어있는 리스트에 값 추가하기

```python
for i in range(0,3):
    v.append(i*5)
    print(v)
```

```
[0]
[0, 5]
[0, 5, 10]
```

+ 연산자

```python
print((1, 2, 3) + (4, 5, 6))
print([1, 2, 3] + [4, 5, 6])
print("Hello" + " " + "World")
```

```
(1, 2, 3, 4, 5, 6)
[1, 2, 3, 4, 5, 6]
Hello World
```

* 연산자

The * operator produces a new tuple, list, or string that "repeats" the original content.

```python
y = 2.5
print(type(y)) # Prints "<class 'float'>"
print(y, y + 1, y * 2, y ** 2) # Prints "2.5 3.5 5.0 6.25"
```

```
<class 'float'>
2.5 3.5 5.0 6.25
```

Enumeration (열거하기)

```python
for i, val in enumerate(v):
    print('{} ---> {}'.format(i, val))
```

```
0 ---> 0
1 ---> 5
2 ---> 10
```

```python
v2 = [ ':', '?', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'V', 'W', 'Y', '0', '1', '2', '3', '4', '5', '6'
print(v2)
```

```
[':', '?', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'V', 'W', 'Y', '0', '1', '2', '3',
 '4', '5', '6', '7', '8', '9']
```

```
for i, val in enumerate(v2):
    print('{} ---> {}'.format(i, val))
```

```
0 ---> :
1 ---> ?
2 ---> A
3 ---> B
4 ---> C
5 ---> D
6 ---> E
7 ---> F
8 ---> G
9 ---> V
10 ---> W
11 ---> Y
12 ---> 0
13 ---> 1
14 ---> 2
15 ---> 3
16 ---> 4
17 ---> 5
18 ---> 6
19 ---> 7
20 ---> 8
21 ---> 9
```