

Fastcampus Data Science SCHOOL

Python Basic

Introduce

최우영

- Solution Architect, Back-End Developer at unknown team
- Solution Architect, Web Developer, Instructor
- Skills: Python, Golang, Julia, Node.js, Google tag manager ...

blog: <http://blog.ulgoon.com/>

github: <https://github.com/ulgoon/>

email: me@ulgoon.com

Python Basic

Python은?

1989년 크리스마스 연휴를 보내던 Guido van Rossum이 만든 고급 프로그래밍 언어

특징

- 인터프리터
- 객체지향
- 동적타이핑
- 엄격한 문법

Python Basic

Python으로 할 수 있는 것들!

- System Programming
- Web Programming
- Data Analysis
- ...

Zen of Python - PEP 20

```
import this
```

The Zen of Python, by Tim Peters

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better

Python Basic

REPL - Read - Eval - Print Loop

코드를 입력하면 바로 결과를 확인할 수 있음!!

We'll use python3

difference of 2.x , 3.x

Short version: Python 2.x is legacy, Python 3.x is the present and future of the language

Online Python Code Editor

<https://repl.it/>

Hello python!

So, let's try!!

```
print("hello python!")
```

Naming Convention

- 클래스 이름은 `CamelCase`
- 함수, 변수, 메소드 이름은 `snake_case`

파이썬에서 쓰이지 않는 네이밍 규칙

- `chHungarianNotation`
- `javaScriptStyleCamelCase`

Syntax

문법, 구조, 또는 언어 문장 내에 있는 구성요소의 순서

"나는 입니다 학생" (Syntax Error)

"나는 학생 입니다" (Syntactically Valid)

"Python"5 (Syntax Error)

3.6 * 12 (Syntactically Valid)

Numbers & Math

<object> <operator> <object>

```
print(3 + 7)
print(10 - 3)
print(15 / 7)
print(34 * 100)
```

Numbers & Math

```
print(15 / 7)
print(15 / 5)
type(15 / 5)

print(15 // 5)
type(15 // 5)

print(34 * 100)
print(3 * 2.5)
type(3 * 2.5)
```

Boolean

```
print(3 < 7)
print(10 < 3)
print(15 > 7)
print(34 == 100)
```

!=

>=

<=

Variable

```
print("hello python!")  
hello = "hello"  
python = "python!"  
print(hello, python)
```

```
num1 = 14  
num2 = 5  
  
print(num1+num2)  
print(num1-num2)  
print(num1*num2)  
print(num1/num2)
```

input

```
name = input("What is your name? ")  
print("Hi, ", name)
```

input with evaluation

```
input("How old are you? ")  
eval(input("How old are you? "))
```

equals to `raw_input()`, `input()` in python 2.x

List, Tuple

List

```
animals = [' ', ' ', ' ']
```

Tuple

```
animals = (' ', ' ', ' ')
```

List

빈 list를 선언합니다. 선언과 동시에 값을 채워넣을 수 있습니다.

```
lang = ["python", "c", "java", "golang"]
```

```
lang = []
```

list에 요소를 추가합니다.

```
lang.append("python")
```

```
lang.append("java")
```

```
lang.append("golang")
```

```
print(lang)
```

혹은 특정한 위치에 원하는 값을 추가할 수 있습니다.

```
lang.insert(1, "c")  
print(lang)
```

특정 요소를 삭제할 수도 있습니다.

```
lang.remove("golang")  
print(lang)
```

혹은 리스트에 있던 값을 빼낼 수도 있습니다.

```
java = lang.pop(2)  
print(lang)  
print(java)
```

리스트를 정렬하는 법을 알아보니다.

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

```
print(numbers)
```

```
numbers.sort()
```

```
print(numbers)
```

리스트를 역순으로 출력하고 싶을땐 이렇게 한답니다.

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

```
numbers.reverse()
```

```
print(numbers)
```

특정 값의 위치를 출력할때 이렇게 합니다.

```
index_of_two = numbers.index(2)  
print(index_of_two)
```

리스트끼리 더할 때 extend를 활용합니다.

```
numbers += [5, 6]  
print(numbers)  
numbers.extend([7, 8])  
print(numbers)
```

Tuple

Tuple은 괄호를 이용해 선언할 수 있습니다.

```
tuple1 = (1, 2, 3, 4)
```

tuple은 삭제나 추가가 불가능합니다.

```
del tuple[1]  
tuple1[1] = 'c'
```

tuple끼리 더하거나 반복하는 것은 가능합니다.

```
tuple2 = (5, 6)
```

```
print(tuple1 + tuple2)
```

```
print(tuple1 * 3)
```

tuple은 값을 편하게 바꿀 수 있습니다.

```
x = y
y = x (x)

temp = x
x = y
y = temp

(x,y) = (y,x)
```

혹은 함수에서 하나 이상의 값을 반환할 때 사용합니다.

```
def quot_and_rem(a,b):
    quot = x // y
    rem = x % y
    return (quot, rem)

(quot, rem) = quot_and_rem(3,10)
```

조건문

If

```
if 조건:
    실행문

if 조건1 and 조건2:
    실행문

if 조건1 or 조건2:
    실행문

if not 조건:
    실행문
```

Comparison Operators

```
x == n
x != n

x < n
x > n
x <= n
x >= n
```

else

```
if 조건:  
    실행문1  
else:  
    실행문2
```

else if

```
if 조건1:  
    실행문1  
else:  
    if 조건2:  
        실행문2  
    else:  
        실행문3
```

elif

```
if 조건1:
    실행문1
elif 조건2:
    실행문2
elif 조건3:
    실행문3
...
else:
    실행문n
```

numguess

```
import random

answer = random.randint(1,100)
print(answer)
```

numguess

```
username = input("Hi there, What's your name?? ")
guess = eval(input("Hi, "+ username + "guess the number: "))

if guess == answer:
    print("Correct! The answer was ", str(answer))
else:
    print("That's not what I wanted!! The answer was ", str(
```

numguess advanced!!

how to make it with more fun??

오늘의 숙제

- numguess에 추가할 수 있을 게임요소 생각해보기
- [Hackerrank](#) 30 Days of code Day0, Day1 풀어보기