

Fastcampus

컴퓨터공학 입문 스쿨

Python Basic_Day2

어제의 숙제!

sample output

```
radius = 10  
diameter = 20  
circumference = 62.830  
area = 314.15  
volume = 4188.666666666666
```

```
r = 10
pi = 3.1415
d = r * 2
c = 2 * pi * r
a = pi * r * r
v = (4 / 3) * pi * r * r * r

print("radius =", r)
print("diameter =", d)
print("circumference =", c)
print("area =", a)
print("volume =", v)
```

Let's Code PYTHONIC

Important Python Enhance Proposal

Layout

- 들여쓰기: 공백 4칸 or 탭(섞어쓰면 안됨)
- 한 줄은 79자(120자도 상관없음)
- 클래스정의와 최상위 함수는 두 줄을 띄움
- 클래스 내 메소드는 한 줄을 띄움

Important Python Enhance Proposal

Variables

- `_variable` : 내부적으로 사용되는 변수
- `print_` : 파이썬 키워드와 충돌 방지

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)

type casting

`float(3) --> int to float`

`int(3.6) --> float to int`

`str(1) --> int to string`

`int("12") --> string to int`

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

String Formatting

```
print("I have a %s, I have an %s." % ("pen", "apple"))
```

- `%s` – string
- `%c` – character
- `%d` – Integer(decimal)
- `%f` – floating-point
- `%o` – 8진수(octal)
- `%x` – 16진수(hexadecimal)
- `%%` – %

Strings

```
some_string = "python"  
len(some_string)
```

- index

p	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

```
some_string[3:5] = "ho"  
some_string[1:5:2] = "yh"  
some_string[:] = some_string[0:len(some_string):1]  
some_string[::-1] = "nohtyp"
```

but, strings are immutable

```
>>> some_string = "python"

>>> some_string[0] = "c"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment

>>> some_string = "c" + some_string[1:]
```

String Functions

```
func = "python is easy programming language"  
func.count('p')
```

```
func.find('p')
```

```
comma = ","  
func = comma.join('python')
```

```
func.split(',')
```

```
python_is_easy = "python is easy"  
python_is_easy.split()
```

```
python_is_easy.replace("python", "golang")
```

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??

For, while

```
for 변수 in (리스트 or 문자열):  
    실행문1  
    ...
```

```
for i in ["python", "java", "golang"]:  
    print(i)
```

For, while

```
sum = 0
for i in range(1,11):
    sum += i
    sum = sum + i
    print(sum)
```

For, while

```
while 조건:  
    실행문1  
    ...
```

```
while name != "foo bar":  
    name = input("What's your name? ")  
    print("Hi, " + name + "So, where is foo bar?")
```

```
while 1:  
    print("Hello world!")
```

Fizzbuzz

Fizzbuzz

```
num = eval(input("type the number: "))

for i in range(1, num + 1):
    if i % 15 == 0:
        print("fizzbuzz")
    elif i % 3 == 0:
        print("fizz")
    elif i % 5 == 0:
        print("buzz")
    else:
        print(i)
```

오늘의 숙제!

Monty Hall Problem



Monty Hall Problem



Let's Simulate This Problem