
PYTHON
PROGRAMMING
BASIC 1

01

파이썬의 특징

PYTHON

- 사람이 생각하는 대로 코드를 작성할 수 있어서, 아이디어를 보다 편하게 구현할 수 있다.
  [이 과정에서 문법적인 엄밀성이 다른 언어보다 유연하게 작동한다.]

- 아이디어) 1,2,3,4 중에서 3이 있으면, 3을 출력하기
- 구현) for i in [1,2,3,4]: print("3이 있습니다")
- 문법이 간결하고 다른 언어에 비해서 비교적 쉬움.
- 가격? 무료!
- 개발 속도가 비교적 빠르다.
- 다른 언어들과 다르게 {} 대신에, 들여쓰기를 사용하여서 비교적 읽기가 편하게 되어 있다.

- 단, 속도적인 부분이 중요한 것은 파이썬의 단점이지만, 그래서 외형적인 부분은 파이썬이나 프로토타입의 경우 파이썬으로 구현을 하고, 속도 및 내부에 중요한 부분은 C 등으로 구현을 하여 하이브리드 방식으로도 사용함.

01

파이썬의 특징

PYTHON

```python
def factorial(x):
    if x == 0:
        return 1
    else:
        return x * factorial (x - 1)
```

```c
int factorial(int x)
{
  if(x == 0)
  {
     return 1;
  }
  else
  {
    return x * factorial(x -1);
  }
}
```

- 동일한 내용이지만
  위는 파이썬, 아래는 C로
  작성된 것이다.

- 동일한 내용을 표현하는데,
  파이썬이 { }대신 들여쓰기를 사용하여
  코드를 읽는 가독성에 있어
  비교적 편한 부분이 있다.

02

기본적인 프로그램 과정

PYTHON

- **변수 :** 값을 저장하기 위한 공간
- **제어문 / 반복문 :** 상황에 따른 제어 및 단순한 반복을 수행하기 위한 부분
- **기타 :** 통신, 파일 등 여러가지 부분

03

기본적인 파이썬코드 작성 방법

PYTHON

- 변수 등에서 대소문자를 구별을 하고 있음.
- C언어와 다르게 변수 지정할 경우에 “형”지정이 없음.
  또한 중간에 다른 형태의 값을 지정하면서, “자동적으로 형”이 변경이 됨.
- 들여쓰기로 구분함.
- 참고) 과정에서는 파이썬 2.*버젼을 사용할 것이며, 파이썬 2.*버전은 기본 인코딩이 ascii로 되어 있어서(파이썬 3.*는 Unicode), 주석과 같은 부분에서 한글을 사용하기 위해서는 추가적인 부분을 해주어야 함.
- 맨 위에 #-*- encoding: utf-8 -*- 을 입력해주어야 한글 주석이 가능 함.

03

기본적인 파이썬코드 작성 방법

PYTHON

```
# 대소문자 구별함
A = "a"
a = "1"
print A
print a
```

```
a
1
```

```
# 변수의 형을 마음대로 변경 가능함
a = 3
print type(a)
a = 3.111111
print type(a)
a="122334abcd"
print type(a)
```

```
<type 'int'>
<type 'float'>
<type 'str'>
```

```
# 들여쓰기 사용
def madefunction():
    print "Function"
```

04

변수

PYTHON

- 변수 선언 : 항상 왼쪽에 문자로 선언을 하고,
  =(대입 연산자)를 통해서 값(숫자, 문자, 객체 등)을 할당한다.

- 중요한 점은 값을 할당하는 것이 아니라 서로 다른 객체들을 서로 “연결”을 하는 것이다.

- 변수명으로 사용할 수 없는 예약어들이 있음
  (예 : and, as, for, print etc)

04

변수

PYTHON

```
# 변수명은 항상 왼쪽에
a = 12
12 = a
```

```
  File "<ipython-input-7-d5d40f6d065e>", line 3
    12=a
SyntaxError: can't assign to literal
```

```
a = 12
b = "Hello"
print a
print b
print "---------"

b = a
print a
print b
print "---------"

a = 98
print a
print b
```

```
12
Hello
---------
12
12
---------
98
12
```

05

변수의 자료형

PYTHON

- 변수가 가질 수 있는 여러가지 자료형이 있음.
- 숫자 – 정수형, 실수형 등 [1, 1.4, 2.33333]
- 문자 – 문자 / 문자열 [“a”, “hello”]
- 리스트 – 여러개 모아 놓은 것 [1,2,3,4]
- 듀플 : 리스트와 비슷, 값 변경 못함
- Dictionary : 사전구조{ “a” : 1234}

06

산술연산자/이항연산자

PYTHON

- a + b
- a - b
- a * b
- a / b
- a // b (몫)
- a ** b (a^b)
- a & b (AND)
- a | b (or)
- a == b
- a != b

07

문자열

PYTHON

문자열 표기 : ‘, “

문자열 내부에 ‘, “ 같은 것이 데이터로 있어야 하는 경우에는 “’’, “””으로 사용함.

- 단, 3개의 기호로 사용을 해도 문자 시작이나 끝에 위치하게 되면
  자동적으로 4개의 기호가 사용되어서 에러가 발생할 수 있다.
- 해결책 : “등 특수기호 앞에 \를 표시함 \”

문자열 연결 : + 를 사용하면 쉽게 연결함.

str(1) : 확실하게 문자열로 인식하게 함.

07

문자열
PYTHON

```
a = "abcd"
a += "abcd"
print a
```

```
abcdabcd
```

```
# 아무리 3개로 표현을 해도 마지막에 4개로 되면 문제가 발생함.
a = """Hello. “What!!!"""
print a
```

```
  File "<ipython-input-406efbf3a81503>", line 1
    a = """Hello. “What!!!"""
                             ^
SyntaxError: EOL while scanning string literal
```

```
a = """Hello. “What!!! \"""
print a
Hello. "What!!!"

a = "Hello. \“What!!! \""
print a
Hello. "What!!!"
```

08

리스트

PYTHON

- 생성 : [ ]
- 원소 추가 : append(그냥 뒤에 추가), insert(원하는 위치에 추가)
- 원소 삭제 : pop(위치), remove(값)
- 리스트끼리 합치기 : +

08

리스트

PYTHON

```
a_list = [1,2,3,"abcd"]
print a_list
```

```
[1,2,3,'abcd']
```

```
a_list.append("a")
print a_list
```

```
[1,2,3,'abcd', 'a']
```

```
a_list.insert(2,"b")
print a_list
```

```
[1,2,'b',3,'abcd','a']
```

```
print a_list
a_list.pop(3)
print a_list
```

```
[1,2,'b',3,'abcd','a']
[1,2,'b','abcd','a']
```

```
print a_list
a_list.remove(1)
print a_list
```

```
[1,2,'b','abcd','a']
[2,'b','abcd','a']
```

09

Dictionary

PYTHON

- key + value 구조
- 생성 : { }

09

Dictionary

PYTHON

```
d1 = {"a":1234, "b":5678}
print d1
```

```
{'a':1234, 'b':5678}
```

```
d1.keys()
```

```
['a', 'b']
```

```
d1.values()
```

```
[1234,5678]
```

```
d1["a"]
```

```
1234
```

10

if 문

PYTHON

```
if 조건문1 :
    내용
elif 조건문2 :
    내용
else :
    내용
```

10

if 문

PYTHON

```
a = 2
if a > 0:
    print "a는 양수"
elif a ==0:
    print "a는 0"
else:
    print "a는 음수"
```

11

for

PYTHON

for 변수 in (범위):

참고 : range(숫자), range(시작점, 끝점, 간격)

```
for i in range(0,10):
    print i
```

```
0
1
2
3
4
5
6
7
8
9
```

```
a = [12,34,56]
for i in a:
    print i
```

```
12
34
56
```

11

실습1

PYTHON

a = ["a","b","c","d","f","g","c","i"]

a에서 c만 제거해서 다음과 같이 만들기

b = ["a","b","d","f","g","i"]

12

실습2

PYTHON

- a = [["a","b","c","d"],"f","g",["c","i"]]
- a에서 c만 제거해서 다음과 같이 만들기
- b = ["a","b","d","f","g","i"]