



변수, 기초 자료형, 연산자

2019년 7월

Contents

- 주석
- 출력문 print
- 이스케이프 문자
- 변수
- 입력 input
- 기본 자료형
- 데이터형 변환 Casting
- 연산자 : 산술
- 연산자 : 대입
- 연산자 : 관계(비교)
- 연산자 : 논리(조건)
- 연산자 우선 순위

주석

- 주석(Comment)란 실행에 상관없는 코드.
- 한줄 주석
 - # comment
 - 파이참과 같은 텍스트 에디터에서는 [Ctrl]+[/] 단축키 사용가능
- 여러줄 주석
 - 시작과 종료 부분에 홑따옴표(')를 3개 연속으로 삽입한다.
 - ```
' ' '
Comment
' ' '
```

# 출력문

- 형식1

`print( value )`  
`print(variable)`

```
print('hello world')
txt = 'python is fun'
print(txt)
```

- 형식2

`print( value1, value2 ... )`  
`print( variable1, variable2 ... )`

- 형식3

`print( value, end="" )`

# 출력문과 이스케이프 문자

| 이스케이프 문자 | 역할        | 설명                       |
|----------|-----------|--------------------------|
| \n       | 새로운 줄로 이동 | <b>Enter</b> 를 누른 효과     |
| \t       | 다음 탭으로 이동 | <b>Tab</b> 을 누른 효과       |
| \b       | 뒤로 한 칸 이동 | <b>Backspace</b> 를 누른 효과 |
| \\       | \ 출력      |                          |
| \'       | ' 출력      |                          |
| \"       | " 출력      |                          |

```
1 print("\n줄바꿈\n연습 ")
2 print("\t탭키\t연습")
3 print("글자가 \"강조\"되는 효과1")
4 print("글자가 \'강조\' 되는 효과2")
5 print("\\\\\\\\ 역슬래시 세 개 출력")
6 print(r"\n \t \" \"를 그대로 출력")
```

# 변수

- 다음 예와 같은 a, b, c를 변수라고 한다.

```
>>> a = 1
>>> b = "python"
```

변수명 = 값

- 변수란 어떠한 값을 저장하는 메모리 공간(그릇)
- 파이썬에서 사용하는 변수는 객체를 가리키는 것
- 변수는 객체가 저장된 메모리의 위치를 가리키는 레퍼런스(Reference) 이다.
- 변수를 만들 때는 대입연산자, =(assignment) 기호를 사용한다.

# 변수

변수명 = 값

변수명1 = 변수명2 = 값

# 변수값 교체하기

변수1, 변수2 = 변수2값, 변수1값

# 변수

- 변수명(식별자) 규칙

- 대 · 소문자를 구분한다( `myVar`와 `MyVar`는 다른 변수).
- 문자, 숫자, 언더바(`_`)를 포함할 수 있다. 하지만 숫자로 시작하면 안 된다  
( `var2(0)`, `_var(0)`, `var_2(0)`, `2Var(X)`).
- 예약어는 변수명으로 쓰면 안 된다.  
파이썬의 예약어는 `True`, `False`, `None`, `and`, `or`, `not`, `break`, `continue`, `return`, `if`, `else`, `elif`, `for`, `while`, `except`, `finally`, `global`, `import`, `try` 등이다.
- 변수명은 의미있는 단어로 표기하는 것이 좋다.



# 변수

```
키워드 목록 확인하기
import keyword
print('키워드 목록')
print(keyword.kwlist)
print('키워드 갯수 : ', len(keyword.kwlist))
```

키워드 목록

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async',
'await', 'break', 'class', 'continue', 'def', 'del', 'elif',
'else', 'except', 'finally', 'for', 'from', 'global', 'if',
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
키워드 갯수 : 35
```

# 변수

- 변수 삭제하기
  - `del(변수명)`

*#### 선언된 a 변수 삭제 후 출력하기*

```
del(a)
```

```
print(a)
```

```
>>> NameError: name 'a' is not defined
```

# 변수

- is 연산자 이용하기
  - 변수1 is 변수2
  - True / False 결과값 표시

```
퀴즈 1
변수 a,b 에 같은 값으로 선언
a = b = 10
print('a = %d' % a)
print('b = %d' % b)
```

```
퀴즈2
is 연산자를 이용하여 a,b 가 같은지 출력하여라
print(a is b)
result - True
```

# 입력문

- Input
  - 사용자가 값을 입력할 때 사용한다.
  - 입력되는 모든 것을 문자열로 처리한다.
- 변수에 저장하기  
변수명 = input("질문")

```
a = input("What is your name?")
print(a)
b = input("What is your ID?")
print(b)
```

# 기본 자료형

## ■ 자료형

- 프로그래밍을 할 때 쓰이는 숫자, 문자열 등등의 것
- 숫자형, 문자열, 리스트, 튜플, 딕셔너리, 집합, 불린(Boolean)

## ■ 숫자형

- 정수형 (1, 2, -2)
- 실수 (1.24, -34.56)
- 컴퓨터식 지수 표현 방식 (4.24e10, 4.24e-10)  
: e표기법. E뒤의 값이 10지수임을 표현
- 복소수 (1+2j)
- 8진수 (0o37)
- 16진수 (0x7A)

# 기본 자료형

## ■ 기본 자료형 알아보기

| 유형        |     |         | 설명                         | 예시              | 선언 형태        |
|-----------|-----|---------|----------------------------|-----------------|--------------|
| 수치<br>자료형 | 정수형 | Integer | 양/음의 정수                    | 1,2,3,100, -9   | data = 1     |
|           | 실수형 | Float   | 소수점이 포함된 실수                | 10.2, -9.3, 9.0 | data = 9.0   |
| 문자형(문자형)  |     | String  | 따옴표 (' / ")에 들어가 있는<br>문자형 | abc, a20abc     | data = 'abc' |
| 논리/불린 자료형 |     | Boolean | 참 또는 거짓                    | True, False     | data = True  |

# 기본 자료형

## ■ 문자열형

- 문자의 나열. 문자들의 집합
- 문자열은 작은 따옴표( ' ' ), 큰 따옴표( " " ), 인용부호 3개로 감싸야 한다.
- 파이썬에서는 JAVA, C언어에 존재하는 문자형(char형)이 따로 없다.

```
a = "이건 큰따옴표 \" 모양."
b = '이건 작은따옴표 \' 모양.'
print(a, b)
```

역슬래시(\) 뒤에 큰따옴표나 작은따옴표를 사용해도  
글자로 인식한다.

### 출력 결과

이건 큰따옴표 " 모양. 이건 작은따옴표 ' 모양.

```
큰 따옴표안에 작은 따옴표를 포함할 수 있다.
예> "I'm OK!!"
print("I'm OK!!")
```

# 기본 자료형

- 문자열형

## 문자열 자료형 만드는 4가지 방법

```
"Hello World"
```

```
'Python is fun'
```

```
"""Life is too short, You need python"""
```

```
'''Life is too short, You need python'''
```



# 기본 자료형

- 문자열형

## 여러 줄로 이루어진 문자열 - 3개의 따옴표 이용

```
>>> multiline = "Life is too short \n You need python"
```

```
>>> multiline='''
... Life is too short
... You need python
... '''
```

# 기본 자료형

- 불(Boolean)형
  - 참과 거짓
  - True / 1
  - False / 0

```
a = (100 == 100)
b = (10 > 100)
print(a, b)
```

출력 결과

True False

# 기본 자료형

## 자료형의 참과 거짓

: 모든 자료형에는 참과 거짓이 있다.

| 값         | 참 or 거짓 |
|-----------|---------|
| "python"  | 참       |
| ""        | 거짓      |
| [1, 2, 3] | 참       |
| []        | 거짓      |
| ()        | 거짓      |
| {}        | 거짓      |
| 1         | 참       |
| 0         | 거짓      |
| None      | 거짓      |

# 자료형 변환 Casting

- 캐스팅(Casting)이란?
  - 데이터간의 형 변환
- 파이썬의 대표적인 캐스팅관련 함수
  - float() - 실수형 형변환
  - int() - 정수형 형변환
  - str() - 문자열 형변환
  - type() - 데이터의 형 출력

# 연산자

## ■ 산술 연산자

| 연산 기호 | 뜻                   | 예시        | 결과                     |
|-------|---------------------|-----------|------------------------|
| +     | 더하기                 | $7+4$     | 11                     |
| -     | 빼기                  | $7-4$     | 3                      |
| *     | 곱하기                 | $7*4$     | 28                     |
| /     | 나누기                 | $7/4$     | 1.75                   |
| **    | 제곱 (같은 수를 여러 번 곱함)  | $2**3$    | 8 (2를 세 번 곱함 $2*2*2$ ) |
| //    | 정수로 나누었을 때의 몫       | $7//4$    | 1 (나눗셈의 몫)             |
| %     | 정수로 나누었을 때의 나머지     | $7\%4$    | 3 (나눗셈의 나머지)           |
| ( )   | 다른 계산보다 괄호 안을 먼저 계산 | $2*(3+4)$ | 14                     |

# 연산자

- 대입 연산자

| 연산자              | 사용 예                 | 설명                           |
|------------------|----------------------|------------------------------|
| <code>+=</code>  | <code>a += 3</code>  | <code>a = a + 3</code> 과 동일  |
| <code>-=</code>  | <code>a -= 3</code>  | <code>a = a - 3</code> 과 동일  |
| <code>*=</code>  | <code>a *= 3</code>  | <code>a = a * 3</code> 과 동일  |
| <code>/=</code>  | <code>a /= 3</code>  | <code>a = a / 3</code> 과 동일  |
| <code>//=</code> | <code>a //= 3</code> | <code>a = a // 3</code> 과 동일 |
| <code>%=</code>  | <code>a %= 3</code>  | <code>a = a % 3</code> 과 동일  |
| <code>**=</code> | <code>a **= 3</code> | <code>a = a ** 3</code> 과 동일 |

# 연산자

## ■ 관계 연산자

- 어떤 것이 크거나 작거나 같은지 비교하는 것(참은 True로, 거짓은 False로 표시)
- 반복문, 조건문에 주로 사용

| 연산자 | 의미      | 설명            |
|-----|---------|---------------|
| ==  | 같다.     | 두 값이 동일하면 참   |
| !=  | 같지 않다.  | 두 값이 다르면 참    |
| >   | 크다.     | 왼쪽이 크면 참      |
| <   | 작다.     | 왼쪽이 작으면 참     |
| >=  | 크거나 같다. | 왼쪽이 크거나 같으면 참 |
| <=  | 작거나 같다. | 왼쪽이 작거나 같으면 참 |

# 연산자

- 논리 연산자

조건이 참인지 거짓인지 판단할 때 사용하는 연산자  
조건문에 주로 사용

| 연산자       | 의미       | 설명             | 사용 예                     |
|-----------|----------|----------------|--------------------------|
| and(논리곱)  | ~이고, 그리고 | 둘 다 참이어야 참     | (a > 100) and (a < 200)  |
| or(논리합)   | ~이거나, 또는 | 둘 중 하나만 참이어도 참 | (a == 100) or (a == 200) |
| not(논리부정) | ~아니다, 부정 | 참이면 거짓, 거짓이면 참 | not(a < 100)             |



# 연산자

■ 연산자 우선 순위  
: 여러 개의 연산자가  
있을 경우 정해진  
순서

| 우선순위 | 연산자                      | 의미                  |
|------|--------------------------|---------------------|
| 1    | () [] {}                 | 괄호, 리스트, 딕셔너리, 세트 등 |
| 2    | **                       | 지수                  |
| 3    | + - ~                    | 단항 연산자              |
| 4    | * / % //                 | 산술 연산자              |
| 5    | + -                      |                     |
| 6    | << >>                    | 비트 시프트 연산자          |
| 7    | &                        | 비트 논리곱              |
| 8    | ^                        | 비트 배타적 논리합          |
| 9    |                          | 비트 논리합              |
| 10   | <> <=                    | 관계 연산자              |
| 11   | == !=                    | 동등 연산자              |
| 12   | = %= /= //= -= += *= **= | 대입 연산자              |
| 13   | not                      | 논리 연산자              |
| 14   | and                      |                     |
| 15   | or                       |                     |
| 16   | if~else                  | 비교식                 |