

▼ Python Data Type and Operations

```
import warnings
warnings.filterwarnings('ignore')
```

▼ I. Python Function

▼ 1) 함수 실행

- print() 실행

```
print('Hello World')
```

Hello World

- 매개변수(Parameter) 및 인자/인수(Argument) 지정

```
print('Hello World', end = 'Wt')
```

Hello World

```
print('Hello', 'World', sep = '-')
```

Hello-World

- 내장 매뉴얼

```
help(print)
```

Help on built-in function print in module builtins:

```
print(...)
print(value, ..., sep=' ', end='\\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.

▼ 2) 객체(Object) 선언

- 파이썬 식별자(Identifiers)는 객체, 함수, 클래스, 모듈 또는 다른 개체 식별에 사용되는 이름
 - 식별자는 문자(A-Z, a-z)로 시작하고 밑줄(_) , 숫자(0~9)를 사용
 - 특수문자 @, \$, %는 식별자로 사용할 수 없음
 - 파이썬은 대소문자를 구분

```
koo = 'Data Analytics'
print(koo)
```

Data Analytics

```
type(koo)
```

str

- 예약어(35개)
 - 객체명으로 사용할 수 없음

```
import keyword
```

keyword.kwlist

```
['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']
```

▼ II. Python Data Type

▼ 1) Numeric

- 정수

```
print(888)  
print(type(888))  
  
888  
<class 'int'>
```

- 8진수

```
print(0o77)  
print(type(0o77))  
  
63  
<class 'int'>
```

- 16진수

```
print(0xff)  
print(type(0xff))  
  
255  
<class 'int'>
```

- 2진수

```
print(0b1001)  
print(type(0b1001))  
  
9  
<class 'int'>
```

- 실수

```
print(3.14)
print(type(3.14))
```

```
3.14
<class 'float'>
```

- 과학적 표기법

```
print(5e-4)
print(type(5e-4))
```

```
0.0005
<class 'float'>
```

```
print(5e+4)
print(type(5e+4))
```

```
50000.0
<class 'float'>
```

- 복소수(Complex)

```
print(8 + 9j)
print(type(8 + 9j))
```

```
(8+9j)
<class 'complex'>
```

▼ 2) String

```
print('A')
print(type('A'))
```

```
A
<class 'str'>
```

```
print('Data Analytics')
print(type('Data Analytics'))
```

```
Data Analytics
<class 'str'>
```

▼ 3) Logical

- 참(True)
- 거짓(False)

```
print(True)
print(type(True))
```

```
True
<class 'bool'>
```

```
print(False)
print(type(False))
```

```
False
<class 'bool'>
```

▼ III. 산술 연산(Arithmetic Operation)

▼ 1) 사칙연산

- 덧셈

```
8 + 9
```

```
17
```

- 뺄셈

```
8 - 9
```

```
-1
```

- 곱셈

```
8 * 9
```

```
72
```

- 실수 나눗셈

```
10 / 3
```

```
3.3333333333333335
```

- 나눗셈 후 몫 반환

```
10 // 3
```

```
3
```

- 나눗셈 후 나머지 반환

```
10 % 3
```

```
1
```

- 제곱

```
9 ** 3
```

```
729
```

```
pow(9, 2)
```

```
81
```

▼ 2) 절댓값

```
abs(-3)
```

```
3
```

▼ 3) 진법 변환

- 16진수

```
hex(16)
```

- 8진수

```
oct(8)
```

- 2진수

```
bin(9)
```

▾ 4) 반올림

- `round(number[, ndigits])`

```
round(24.47)
```

```
24
```

```
round(24.57)
```

```
25
```

```
round(24.57, ndigits = 0)
```

```
25.0
```

```
round(24.57, ndigits = 1)
```

```
24.6
```

```
round(24.57, ndigits = -1)
```

```
20.0
```

```
round(25.47, ndigits = -1)
```

```
30.0
```

```
round(25.478, ndigits = 2)
```

```
25.48
```

▾ IV. 비교 연산(Comparison Operation)

▾ 1) Numeric

```
8 > 9
```

```
False
```

```
8 < 9
```

```
True
```

```
8 >= 9
```

```
False
```

```
8 <= 9
```

```
True
```

```
8 == 9
```

```
False
```

```
8 != 9
```

True

▼ 2) Character

- 동작-1

```
'A' > 'B'
```

False

- 동작-2

```
'A' < 'B'
```

True

```
'A' == 'B'
```

False

```
'A' != 'B'
```

True

- Error-1

```
'팔' > 9
```

- Error-2

```
'A' > 9
```

▼ V. 논리 연산(Logical Operation)

```
X = True  
Y = False  
Z = True  
  
print(X, Y, Z)
```

True False True

▼ 1) AND

```
X and Y
```

False

▼ 2) OR

X or Y

True

▼ 3) NOT

not X

False

▼ 4) 괄호 연산자

(X and Y) and (Y or Z)

False

#

#

#

The End

#

#

#