



내장 함수

Index

abs

all

any

chr

dir

divmod

enumerate

eval

filter

hex

id

input

int

isinstance

len

list

map

max

min

oct

open

ord

pow

range

인수가 하나일 경우

인수가 2개일 경우

인수가 3개일 경우

round

sorted

str

sum

tuple

type

zip

- 프로그램 내 미리 기능을 구현 해놓은 것

abs

- 입력받은 숫자의 절댓값을 리턴

```
>>> abs(3)
3
>>> abs(-3)
3
abs(-1.2)
1.2
```

all

- == 논리회로 and

```
>>> all([1, 2, 3])
True
```

```
>>> all([1, 2, 3, 0])
False
```

```
>>> all([])
True
```

any

- == 논리회로 or

```
>>> any([1, 2, 3, 0])
True
```

```
>>> any([0, ""])
False
```

```
>>> any([])
False
```

chr

- 유니코드 숫자값의 해당하는 문자를 리턴

```
>>> chr(97)
'a'
>>> chr(44032)
'가'
```

dir

- 객체가 지닌 변수나 함수를 보여 주는 함수

```
>>> dir([1, 2, 3])
['append', 'count', 'extend', 'index', 'insert', 'pop',...]
>>> dir({'1':'a'})
['clear', 'copy', 'get', 'has_key', 'items', 'keys',...]
```

divmod

- 몫과 나머지를 튜플로 리턴

```
>>> divmod(7, 3)
(2, 1)
```

enumerate

- 순서가 있는 데이터를 입력받아 인덱스 값을 포함하는 enumerate 객체를 리턴

```
>>> for i, name in enumerate(['body', 'foo', 'bar']):
...     print(i, name)
...
0 body
1 foo
2 bar
```

eval

- 문자열로 구성된 표현식을 입력받아 실행한 결과값을 리턴

```
>>> eval('1+2')
3
>>> eval("'hi' + 'a'")
'hia'
>>> eval('divmod(4, 3)')
(1, 1)
```

filter

- 리턴값이 참인것만을 리턴

```
def positive(x):  
    return x > 0  
  
print(list(filter(positive, [1, -3, 2, 0, -5, 6])))  
  
[1, 2, 6]
```

```
>>> list(filter(lambda x: x > 0, [1, -3, 2, 0, -5, 6]))  
[1, 2, 6]
```

hex

- 정수를 16진수 문자열로 반환하여 리턴

```
>>> hex(234)  
'0xea'  
>>> hex(3)  
'0x3'
```

id

- 객체를 입력받아 객체의 고유 주솟값을 리턴

```
>>> a = 3  
>>> id(3)  
135072304  
>>> id(a)  
135072304  
>>> b = a  
>>> id(b)  
135072304
```

input

- 사용자의 입력을 받는 함수

```
>>> a = input()
hi
>>> a
'hi'
>>> b = input("Enter: ")
Enter: hi
>>> b
'hi'
```

int

- 정수가 아닌 숫자를 정수로 리턴

```
>>> int('3')
3
>>> int(3.4)
3
```

```
>>> int('11', 2)
3
```

```
>>> int('1A', 16)
26
```

int(x, radix) - radix : 진수

isinstance

- 입력으로 받은 객체가 입력받은 클래스의 인스턴스인지를 판단하여 참인지 거짓인지를 리턴

```
>>> class Person: pass
...
>>> a = Person()
>>> isinstance(a, Person)
True
```

```
>>> b = 3
>>> isinstance(b, Person)
False
```

len

- 입력값의 길이를 리턴

```
>>> len("python")
6
>>> len([1,2,3])
3
>>> len((1, 'a'))
2
```

list

- 반복 가능한 데이터를 입력받아 리스트로 리턴

```
>>> list("python")
['p', 'y', 't', 'h', 'o', 'n']
>>> list((1,2,3))
[1, 2, 3]
```

```
>>> a = [1, 2, 3]
>>> b = list(a)
>>> b
[1, 2, 3]
```

map

- 입력받은 데이터의 각 요소에 입력받은 함수를 적용한 결과를 리턴

```
>>> def two_times(x):
...     return x*2
...
>>> list(map(two_times, [1, 2, 3, 4]))
[2, 4, 6, 8]
```

```
>>> list(map(lambda a: a*2, [1, 2, 3, 4]))
[2, 4, 6, 8]
```

max

- 입력받은 반복 가능한 데이터의 최댓값을 리턴

```
>>> max([1, 2, 3])
3
```

```
>>> max("python")
'y'
```

min

- 입력받은 반복 가능한 데이터의 최솟값을 리턴

```
>>> min([1, 2, 3])
1
>>> min("python")
'h'
```

oct

- 정수를 8진수 문자열로 바꾸어 리턴

```
>>> oct(34)
'0042'
>>> oct(12345)
'0030071'
```

open

- '파일 이름'과 '읽기 방법'을 입력받아 파일 객체를 리턴

mode	설명
w	쓰기 모드로 파일 열기
r	읽기 모드로 파일 열기
a	추가 모드로 파일 열기
b	바이너리 모드로 파일 열기

```
>>> f = open("binary_file", "rb")
# rb : 바이너리 읽기 모드
```

ord

- 문자의 유니코드 숫자 값을 리턴

```
>>> ord('a')
97
>>> ord('가')
44032
```

pow

- x를 y제곱한 결과값을 리턴

```
>>> pow(2, 4)
16
>>> pow(3, 3)
27
```

range

- 입력받은 숫자에 해당하는 범위 값을 반복 가능한 객체로 만들어 리턴

인수가 하나일 경우

시작 숫자를 지정해 주지 않으면 range 함수는 0부터 시작

```
>>> list(range(5))
[0, 1, 2, 3, 4]
```

인수가 2개일 경우

시작 숫자와 끝 숫자, 끝 숫자는 해당 범위에 미포함

```
>>> list(range(5, 10))
[5, 6, 7, 8, 9]
```

인수가 3개일 경우

세 번째 인수는 숫자 사이의 거리

```
>>> list(range(1, 10, 2))
[1, 3, 5, 7, 9]
>>> list(range(0, -10, -1))
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
```

round

- 입력받은 숫자를 반올림해 리턴

```
>>> round(4.6)
5
>>> round(4.2)
4
```

```
>>> round(5.678, 2)
5.68
```

,2 : 소수점 2자리까so지만 반올림

sorted

- 입력 데이터를 정렬한 후 그 결과를 리스트로 리턴

```
>>> sorted([3, 1, 2])
[1, 2, 3]
>>> sorted(['a', 'c', 'b'])
['a', 'b', 'c']
>>> sorted("zero")
['e', 'o', 'r', 'z']
>>> sorted((3, 2, 1))
[1, 2, 3]
```

str

- 문자열 형태로 객체를 변환하여 리턴

```
>>> str(3)
'3'
>>> str('hi')
'hi'
```

sum

- 입력 데이터의 합을 리턴

```
>>> sum([1, 2, 3])
6
>>> sum((4, 5, 6))
15
```

tuple

- 반복 가능한 데이터를 튜플로 바꾸어 리턴

```
>>> tuple("abc")
('a', 'b', 'c')
>>> tuple([1, 2, 3])
(1, 2, 3)
>>> tuple((1, 2, 3))
(1, 2, 3)
```

type

- 입력값의 자료형이 무엇인지를 알려 주는 함수

```
>>> type("abc")
<class 'str'>
>>> type([ ])
<class 'list'>
>>> type(open("test", 'w'))
<class '_io.TextIOWrapper'>
```

zip

- 동일한 개수로 이루어진 데이터들을 묶어서 리턴

```
>>> list(zip([1, 2, 3], [4, 5, 6]))
[(1, 4), (2, 5), (3, 6)]
>>> list(zip([1, 2, 3], [4, 5, 6], [7, 8, 9]))
[(1, 4, 7), (2, 5, 8), (3, 6, 9)]
```

```
>>> list(zip("abc", "def"))  
[('a', 'd'), ('b', 'e'), ('c', 'f')]
```