



내장함수

Update – 2018.07

Contents

- `abs()`
- `divmod()`
- `eval()`
- `len()`
- `enumerate()`
- `map()`
- `filter()`
- `zip()`



abs()



▶ 절대값 함수

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





divmod()



- 몫과 나머지를 함께 리턴
- `divmod(n1,n2)`

```
>>> divmod(7, 3)
(2, 1)
>>> divmod(1.3, 0.2)
(6.0, 0.099999999999999978)
```





eval()



- eval(문자열)
- 문자열을 수식으로 처리한다

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



len()



- len(data)
- 데이터의 길이를 산출한다.

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



enumerate()



- enumerate(iterable, 시작번호)
- 색인을 부여한다.

```
my_list = ['apple', 'banana']  
for index, value in enumerate(my_list, 1):  
    print(index, value)
```

1 apple
2 banana



map()



- map(값을 변환할 함수, iterable)
- 각 원소마다 함수가 호출되어 리턴값이 변함

```
def power_fn(value):  
    return value ** 2  
list(map(power_fn, [1, 2, 3]))
```

[1, 4, 9]

filter()

- filter(필터링 여부를 결정할 함수, iterable)
- 각 원소마다 함수가 호출되어, 리턴값이 True 판정될 경우 출력

```
def judge_fn(value):  
    return value % 2 == 0      # 짝수만 pass  
list(filter(judge_fn, [1,2,3,4,5]))
```

[2, 4]

zip()

- zip(*iterable)
- 여러개의 리스트나 튜플 또는 순차 자료형을 짝지움

```
actor = ['정우성', '나나']  
gender = ['여', '여']  
for i,j in zip(actor, gender):  
    print(i, '-', j)
```

정우성 - 여
나나 - 여

```
#zip을 통해서 자료가 묶인 걸 풀수도 있다.  
actor = [('나나', '정우성'), ('여', '남')]  
a, b = zip(*actor)  
a
```

('나나', '여')