

## 파이썬 프로그래밍 기초 실습5

### 1. Pythonic Code

- 파이썬 특유의 문법을 활용하여 효율적으로 코드를 표현(파이썬 스타일의 코딩 기법)

예) 여러 문자열을 하나로 이을 때

```
>>> colors = ['red', 'green', 'blue']
>>> result = ''
>>> for word in colors:      # 일반적으로
    result += word

>>> result = ''.join(colors)
```

#### (1) split(), join()

1일차 자료 참고

#### (2) List Comprehensions

- 리스트에 for 문을 사용하여 반복적으로 표현식을 실행해서 리스트 요소들을 정의하는 용법
- 기존 리스트를 사용하여 간단히 다른 리스트를 만드는 기법

[표현식 for 요소 in 시퀀스자료형 [if 조건식]]

```
>>> ls = []                # 일반적으로~
>>> for i in range(1,6):
    ls.append(i)
>>> ls
[1, 2, 3, 4, 5]
```

```
>>> ls = [i for i in range(1,6)]
>>> ls
[1, 2, 3, 4, 5]
```

```
>>> [i ** 2 for i in range(1,6)]
```

```
>>> [i for i in range(1,6) if i % 2 == 0]
```

# 리스트가 2개

```
>>> L1 = [1, 2, 3]
>>> L2 = [3, 4, 5]
>>> L1
[1, 2, 3]
>>> L2
[3, 4, 5]
```

```
>>> [x * y for x in L1 for y in L2]
```

#### (3) enumerate(), zip()

# enumerate()

- 리스트의 모든 요소를 인덱스와 쌍으로 추출

```
>>> colors = ['red', 'green', 'blue']
>>> result = list(enumerate(colors))
>>> result
[(0, 'red'), (1, 'green'), (2, 'blue')]

>>> for idx, color in enumerate(colors):
    print(idx, color)
```

# zip()

- 2개 이상의 리스트를, 각 리스트의 같은 인덱스 원소끼리 묶은 튜플을 요소로 하는 리스트를 만들어 줌

```
>>> list(zip([1,2,3], [4,5,6])) # 리스트
[(1, 4), (2, 5), (3, 6)]
>>> list(zip([1,2,3], [4,5,6], ['a','b','c']))
[(1, 4, 'a'), (2, 5, 'b'), (3, 6, 'c')]
```

```
>>> list(zip([1,2,3], [4,5,6])) # 튜플
[(1, 4), (2, 5), (3, 6)]
```

```
>>> [sum(x) for x in zip([1,2,3], [4,5,6])]
```

# enumerate(), zip()

```
for i, (a,b) in enumerate(zip(['a','b','c'], ['d','e','f'])):
    print(i, a, b)
```

#### 실습1)

아래와 같이 주어진 2개의 리스트로 딕셔너리를 만들어 출력하시오. 단, 순서는 다를 수 있음

```
L1 = ['one', 'two', 'three', 'four']
L2 = [1, 2, 3, 4]
```

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4}
```

힌트) zip()