
Lecture 8. Pythonic Code

BAE SEKANG(배세강)



Contents

1. Pythonics

- raw string
- split
- strip
- join
- list comprehension
- map
- enumerate
- zip
- filter
- lambda function



Pythonics

♣ raw string

- 내부의 문자를 있는 그대로 나타내는 문자열
- 파일 경로를 나타낼 때 유용하다.

```
print("hello\nworld")
```

```
>> hello  
world
```

```
print(r"hello\nworld")
```

```
>> hello\nworld
```



Pythonics

♣ split()

- 문자열을 구분자를 기준으로 나누어 list를 반환한다.
- 기본적으로 공백(스페이스, 탭, 엔터)를 기준으로 구분한다.

```
my_str = "I like Python"  
print(my_str.split())
```

```
>> ['I', 'like', 'Python']
```

```
friends = "Tom,James,Minji"  
print(friends.split(","))
```

```
>> ['Tom', 'James', 'Minji']
```



Pythonics

♣ strip()

- 문자열에서 특정 문자를 양 끝에서부터 찾아 모두 제거한다.
- 기본적으로 공백(스페이스, 탭, 엔터)을 제거한다.
- lstrip(), rstrip()도 있다.

```
text1 = " Hello Python \n"  
print(text1.strip())
```

```
>> Hello Python
```

```
text2 = ",,,rrttgg.....apple....rrr"  
print(text2.strip(",.grt"))
```

```
>> apple
```



Pythonics

♣ join()

- iterable한 객체 안의 str 요소들을 하나의 문자열로 합친다.
- 구분자를 설정할 수 있다. "구분자".join(iterable)의 형태로 사용한다.

```
drinks = ['coke', 'coffee', 'juice']  
print("".join(drinks))
```

```
>> cokecoffeejuice
```

```
nums = ("1", "2", "3")  
print(", ".join(nums))
```

```
>> 1, 2, 3
```



Pythonics

♣ list comprehension

- iterable한 객체를 바탕으로 새로운 list를 간결히 생성할 수 있다.
- [Expression for i in Iterable Conditions]

```
result = [x if x % 2 == 0 else 'odd' for x in range(5)]  
print(result)
```

```
>> [0, 'odd', 2, 'odd', 4]
```

```
data = " apple , grape , cherry "  
print([item.strip() for item in data.split(",")])
```

```
>> ['apple', 'grape', 'cherry']
```



Pythonics

♣ map()

- iterable한 객체의 모든 요소에 특정 함수를 일괄 적용한다.
- map(함수, iterable)

```
str_nums = ["1", "2", "3", "4"]  
int_nums = list(map(int, str_nums))
```

```
# 변수가 3개이므로, 입력값도 정확히 3개여야 한다.  
a, b, c = map(int, input().split())
```




Pythonics

[Practice]

예시와 같이 두 정수 A, B를 입력 받을 때, A보다 크고 B보다 작은 정수들을 오름차순으로 정렬하여 담고 있는 리스트를 출력한다.

[입력 1]

10 15

[입력 2]

31 28

[출력 1]

[11, 12, 13, 14]

[출력 2]

[]



Pythonics

♣ enumerate()

- iterable한 객체를 입력으로 받아 index와 value를 동시에 반환한다.
- for문에서 사용하면 유용하다.

```
fruits = ['apple', 'banana', 'cherry']  
  
for index, fruit in enumerate(fruits):  
    print(f"{index}번 과일: {fruit}")
```



Pythonics

♣ zip()

- iterable한 객체를 인자로 받아 동일한 index의 요소들을 묶어 tuple로 반환
- 딕셔너리를 만들 때 유용하다. 예) dict(zip(keys, values))
- 길이가 짧은 iterable 기준으로 멈춘다.

```
names = ['Alice', 'Bob', 'Charlie']  
scores = [82, 74, 100]
```

```
for name, score in zip(names, scores):  
    print(f"{name}의 점수: {score}")
```



Pythonics

♣ filter()

- iterable한 객체에서 특정 조건에 맞는 요소들만 골라낸다.
- filter(조건함수, iterable)의 형태로, 결과값이 True인 요소만 남긴다.

```
def is_even(n):  
    return n % 2 == 0  
  
nums = [1, 2, 3, 4, 5, 6]  
evens = list(filter(is_even, nums))  
print(evens)
```



Pythonics

♣ lambda function

- 이름 없이 한 줄로 간결히 정의하여 사용하는 익명 함수
- `lambda [Parameter]: [Expression]`

```
def add(x, y):  
    return x + y
```

```
add_lambda = lambda x, y: x + y
```

```
print(add(3, 5))           # 8
```

```
print(add_lambda(3, 5))    # 8
```



Pythonics

♣ lambda function 활용

```
# 모든 요소에 10을 곱한다.  
nums = [1, 2, 3, 4]  
result = list(map(lambda x: x * 10, nums))  
print(result)
```

```
# 정렬 기준을 정한다.  
students = [('Alice', 90), ('Bob', 85), ('Charlie', 95)]  
students.sort(key=lambda x: x[1])  
print(students)
```

```
# 짝수만 남겨놓는다. filter() 활용  
evens = list(filter(lambda x: x % 2 == 0, nums))
```



Pythonics

[Practice]

다음 데이터를 활용하여 80점 이상인 학생들만 골라낸 뒤 순위를 매겨 출력한다.

```
data = " Alice:85 , Bob:72 , Charlie:95 , David:80 , Ellen:65 "
```

[출력]

1등: Charlie (95점)

2등: Alice (85점)

3등: David (80점)



THANK YOU!

Any Question?