

# 파이썬 스타일 코드 - 연습 (일반문제)

## 1. Css Selector 수정

- `today_main_news > div.hdline_news > ul > li:nth-child(1)`

In [1]:

```
selector = "#today_main_news > div.hdline_news > ul > li:nth-child(1)"

selector_list = selector.split(">")
selector_list[-1] = selector_list[-1].split(".")[0]

">".join(selector_list)
```

Out[1]:

```
'#today_main_news > div.hdline_news > ul > li:nth-child(1)'
```

- selector에는 F12를 이용해서 불러온 값을 대입한다.
- selector.split를 이용해 ">"를 기준으로 특정 문자를 나눠준다.

## 2. List Comprehension으로 만드는 구구단

In [2]:

```
def gugu_com(x=2):
    [print(f"{x}x{i} = {x*i}") for i in range(1,10)]

gugu_com(x=2)
```

```
2x1 = 2
2x2 = 4
2x3 = 6
2x4 = 8
2x5 = 10
2x6 = 12
2x7 = 14
2x8 = 16
2x9 = 18
```

- gugu\_com이라는 함수를 만들어서 x=2를 대입합니다.
- i의 값이 1부터 10까지 되어있는데, 처음에 대입한 2와 곱셈을 통해 화면에 도출하도록 합니다.

In [3]:

```
def gugu_com(x=7):
    [print(f"{x}x{i} = {x*i}") for i in range(1,10)]
```

```
gugu_com(x=7)
```

```
7x1 = 7
7x2 = 14
7x3 = 21
7x4 = 28
7x5 = 35
7x6 = 42
7x7 = 49
7x8 = 56
7x9 = 63
```

- 이전에 진행했던것과 같이 숫자 2를 7로 바꿔서 진행한다.

### 3. 두 주사위의 곱

x	2	3	4	5	6
1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36

In [4]:

```
m1 = [1, 2, 3, 4, 5, 6]
m2 = [2, 4, 6, 8, 10, 12]
m3 = [3, 6, 9, 12, 15, 18]
m4 = [4, 8, 12, 16, 20, 24]
m5 = [5, 10, 15, 20, 25, 30]
m6 = [6, 12, 18, 24, 30, 36]
```

```
print(m1)
print(m2)
print(m3)
print(m4)
print(m5)
print(m6)
```

```
[1, 2, 3, 4, 5, 6]
[2, 4, 6, 8, 10, 12]
[3, 6, 9, 12, 15, 18]
[4, 8, 12, 16, 20, 24]
[5, 10, 15, 20, 25, 30]
[6, 12, 18, 24, 30, 36]
```

- 제가 진행한 위의 방법 말고도 다른 더 편한 방법이 있을거 같습니다.

In [5]:

```
die=[i for i in range(1,7)]
[[j*i for i in die] for j in die]
```

Out[5]:

```
[[1, 2, 3, 4, 5, 6],
 [2, 4, 6, 8, 10, 12],
 [3, 6, 9, 12, 15, 18],
 [4, 8, 12, 16, 20, 24],
 [5, 10, 15, 20, 25, 30],
 [6, 12, 18, 24, 30, 36]]
```

- 주사위를 던졌을때 나올수 있는 숫자의 곱을 구하는 방법입니다.

## 4. 두 주사위의 합

+	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

In [6]:

```
dice_sum = [[2, 3, 4, 5, 6, 7],
 [3, 4, 5, 6, 7, 8],
 [4, 5, 6, 7, 8, 9],
 [5, 6, 7, 8, 9, 10],
 [6, 7, 8, 9, 10, 11],
 [7, 8, 9, 10, 11, 12]]
```

```
print(dice_sum[1][5])
print(dice_sum[5][1])
```

8  
8

In [7]:

```
dice = [i for i in range(1,7)]
dice_sum=[ [j+i for i in dice] for j in dice]
```

```
print(dice_sum[1][5])
print(dice_sum[5][1])
```

8  
8

## 문제 1 (표절 프로그램 검사 문제)

In [8]:

```
from collections import defaultdict, Counter

text = """Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues."""

text2 = """C is a very difficult programming language so even if you are good at programming, you can learn c with facing any issues."""

text3 = """R Programming is good at statistical analysis. you can learn easily"""
```

In [9]:

```
word_dict = dict()
word_dict["key"]

## KeyError

word_dict = defaultdict(lambda: 0)
word_dict["key"]
## 0

word_dict["key"] += 1
word_dict["key"]
## 1
```

```
-----
-----
KeyError                                Traceback (most recent call last)
<ipython-input-9-584ef229e42d> in <module>
      1 word_dict = dict()
----> 2 word_dict["key"]
      3
      4 ## KeyError
      5
```

**KeyError: 'key'**

- 2번째 줄에 "key"에서 에러가 나타나는 것은 만들어진 dict에서 key라는 단어가 없기 때문이다.
- null값을 없애고 key값이 없으면 0이 나타나게 설정한다.

In []:

```
def word_counter(text):
    word_count = defaultdict(lambda: 0)
    for word in text.lower().split():
        word_count[word] += 1

    return word_count
```

In []:

```
word_counter(text)
```

## 문제 2

- defaultdict를 Counter로 변환하는 법

In []:

```
Counter({"a": 1, "b": 2, "c": 3}) - Counter({"a": 1, "b": 1, "c": 1})
```

- 전체 단어 수의 합 계산

In []:

```
sum(Counter({"a": 1, "b": 2, "c": 3}).values())
```

$$(1 - \sum (\text{Counter}(A) - \text{Counter}(B)) / \sum \text{Counter}(A)) \times 100$$

In []:

```
def text_similarity(text_count_1, text_count_2):
    text1_count = Counter(text_count_1)
    text2_count = Counter(text_count_2)

    word_total = sum(text1_count.values())
    word_diff = sum((text1_count - text2_count).values())

    return (1 - word_diff / word_total) * 100
```

- 함수를 정의 하기에 앞서, text1의 단어의 수를 먼저 계산하고 text2의 단어를 계산한다.
- 또 word\_total 함수에 두 단어의 수를 더한다.
- word\_diff에서는 두 함수의 차이를 알아본다.

In []:

```
text_similarity(word_counter(text), word_counter(text2))
```

- 퍼센트를 사용하여 알아봤을 때, text과 text2는 약 74% 비슷하다.

In []:

```
text_similarity(word_counter(text), word_counter(text3))
```

- 퍼센트를 사용하여 알아보았을 때, text와 text3은 약 21% 유사하다.

## 파이썬 스타일 코드2 - 연습(실습문제)

### 1. 람다함수

In []:

```
def f(x,y):
    return x+y
print(f(1,4))
```

- 람다함수를 사용하지 않고 일반적인 함수를 사용했을때의 수식입니다.

In []:

```
f=lambda x,y: x+y
print(f(1,4))
```

- 람다를 사용하면 1줄정도를 줄일 수 있고, 복잡하게 진행하지 않아도 됩니다.
- 대부분 반복적인 일을 사용하지 않고, 한번의 식만 도출하고 싶을때 람다함수를 이용합니다.

In []:

```
print((lambda x,y:x+y)(1,4))
```

## 2. 맵 리듀스

### 2.1 map 함수

In []:

```
ex=[1,2,3,4,5]
f = lambda x:x**2
print(list(map(f,ex)))
```

In []:

```
ex=[1,2,3,4,5]
f=lambda x:x**2
for value in map(f,ex):
    print(value)
```

- map함수는 리스트나 시퀀스가 뚜렷한 자료들에게 공통적인 수식을 대입하고자 할 때 사용됩니다.

In []:

```
ex = [1,2,3,4,5]
[x**2 for x in ex]
```

In []:

```
ex=[1,2,3,4,5]
f=lambda x,y:x+y
list(map(f,ex,ex))
```

In []:

```
[x+y for x,y in zip(ex,ex)]
```

## 2.2 reduce 함수

In []:

```
from functools import reduce
print(reduce(lambda x,y:x+y, [1,2,3,4,5]))
```

## 3. 별표의 활용

### 3.1 가변 인수로 활용

In []:

```
def asterisk_test(a,*args):
    print(a,args)
    print(type(args))

asterisk_test(1,2,3,4,5,6)
```

In []:

```
def asterisk_test(a,**kargs):
    print(a,kargs)
    print(type(kargs))
asterisk_test(1,b=2,c=3,d=4,e=5,f=6)
```

### 3.2 별표의 언패킹 기능

In []:

```
## 일반 자료형에서 매트릭스 형태로 나타내는 방법입니다.
a,b,c=([1,2], [3,4], [5,6])
print(a,b,c)
data=([1,2], [3,4], [5,6])
print(*data)
```

In []:

```
## 괄호 첫 숫자와 두번째 숫자의 클래스를 확인합니다.
for data in zip(*[[1,2], [3,4], [5,6]]):
    print(data)
    print(type(data))
```

In []:

```
def asterisk_test(a,b,c,d):
    print(a,b,c,d)
data={"b":1, "c":2, "d":3}
asterisk_test(10, **data)
```

## 4. 선형대수학

### 4.1 파이썬 스타일 코드로 표현한 벡터

In []:

```
vector_a=[1,2,10]
vector_b=(1,2,10)
vector_c={'x':1, 'y':2, 'z':10}
```

In []:

```
u=[2,2]
v=[2,3]
z=[3,5]
result=[]

for i in range(len(u)):
    result.append(u[i]+v[i]+z[i])
print(result)
```

- 기존에 존재하는 리스트에 새로운 값을 추가하기 위해 append를 사용합니다.

In []:

```
u=[2,2]
v=[2,3]
z=[3,5]
result=[sum(t) for t in zip(u,v,z)]
print(result)
```

In []:

```
def vector_addition(*args):
    return[sum(t) for t in zip(*args)]

vector_addition(u,v,z)
```

In []:

```
a=[1,1]
b=[2,2]
[x+y for x,y in zip(a,b)]
```

In []:

```
u=[1,2,3]
v=[4,4,4]

alpha=2

result=[alpha*sum(t) for t in zip(u,v)]
result
```



## 4.2 파이썬 스타일코드로 표현한 행렬

In []:

```
matrix_a= [[3,6],[4,5]]
matrix_b=[(3,6), (4,5)]
matrix_c={(0,0):3, (0,1):6, (1,0):4, (1,1):5}
```

In []:

```
matrix_a=[[3,6], [4,5]]
matrix_b=[[5,8], [6,7]]

result=[sum(row) for row in zip(*t)] for t in zip(matrix_a, matrix_b)]
print(result)
```

## 주민등록번호로 성별 찾기 with map

In []:

```
pins = ["891120-1234567", "931120-234567", "911120-1234234", "951120-1234567"]

list(map(lambda x:x.split("-")[1][0], pins))
```

- 성별은 뒷자리의 가장 앞자리로 구분할 수 있습니다.
- 1은 남자고 2는 여자를 뜻하기때문에, lambda를 이용하여 숫자를 나눠줍니다.

## 벡터의 내적

In []:

```
a=[1,2]
b=[3,4]

dot=lambda a,b: sum([x*y for x,y in zip(a,b)])
dot(a,b)
```

- 괄호안의 1과3을 먼저 곱하고 2와4를 곱한 값을 더하면 원하는 최종값이 도출됩니다.
- 함수람다를 이용하여, 리스트 a와 b안의 값에 대칭되는 값이 서로 곱하여 계산됩니다.