

```
In [2]: #문자열 복습
#-문자열기준으로 split , indexing [-1]은 맨끝
# '문자열'.split(' ') 하나씩 꺼낼수있따

value_1 = '5'
value_2 = '5 - 2 - 10 - 10'.split('-')[-1]

print('5 - 2 - 10 - 10'.split('-'))
print(int(value_1) * 3 + float(value_2)) #float

# ' '.join(a) 리스트를 문자열로 바꾸는 것 join
#split
#zip
#enumerate

"H-e-l-l-o-P-y-t-h-o-n"[:2]

['5 ', ' 2 ', ' 10 ', ' 10']
25.0
```

```
In [3]: course_name = 'This is New AI World'
for i in course_name:
    if i == 'world':
        i = course_name.lower() #소문자로 바뀔일이 없음
print(course_name)

This is New AI World
```

```
In [4]: number = 10
day = 3
print("I eat %d oranges every day." % number) #서식

I eat 10 oranges every day.
```

```
In [7]: >>> sentence = "Hello, my name is python?!"
>>> print(sentence[0]+sentence[5]+sentence[8:11]) #[0,1,2,3,4] 이고 스
페이스도 카운트

H,y n
```

```
In [8]: first_word = "Python"
second_word = "Language"
print((first_word + second_word).capitalize())
print(first_word.find("p"))
print(second_word.isdigit())

Pythonlanguage
-1
False
```

```
In [ ]: #1 capitalize() : 첫 문자를 대문자로 변환한다.
        #2 title() : 각 단어의 앞 글자만 대문자로 변환한다.
        # 3 split() : 공백을 기준으로 나눠 리스트를 반환한다.
        #4 isdigit() : 문자열이 숫자인지의 여부를 반환한다.
        # 5 upper() : 문자를 대문자로 변환한다.
```

```
In [ ]: a = 10
        b = 20
        sum_result = f'a + b = {a+b}'  #{변수값이 들어감}
        print(sum_result)
```

```
In [9]: #튜플 () 는 변경 불가
        # ** 5승
        # // 몫, % 나누기
        #-= 빼기
        #dict : dictionary

        #set 집합이라 중복값 없어짐
        #순서없는 애한테 리스트를 적용하면 순서가 생긴다 [1,3,4,5]
        def quiz_2(list_data):
            a = set(list_data)
            return (list(a)[1:5])
        list_1 = [0, 3, 1, 7, 5, 0, 5, 8, 0, 4]
        print(quiz_2(list_1))

        [1, 3, 4, 5]
```

```
In [ ]: #튜플은 ( , ) 괄호에 콤마 있어야함

        stack

        #튜플 값을 순서 없이 저장하면서 중복을 불허한다.
        #스택 값의 변경이 불가능하며, 리스트의 연산, 인덱싱, 슬라이싱 등을 동일하게 사용한다.
        #큐 먼저 넣은 데이터를 먼저 반환하도록 설계된 메모리 구조로, FIFO(First In First Out)로 구현된다.
        #셋 나중에 넣은 데이터를 먼저 반환하도록 설계된 메모리 구조로, LIFO(Last In First Out)로 구현된다.
```

```
In [17]: country_code={"America":1,"Korea":82,"China":86,"Japan":81}
print(country_code.values()) #list 로 뽑힘 [ , , ]

print(country_code)
print(country_code.keys())
print(85 in country_code.values())
print("Korea" in country_code.keys())
```

#딕셔너리 함수

```
dict_values([1, 82, 86, 81])
{'America': 1, 'Korea': 82, 'China': 86, 'Japan': 81}
dict_keys(['America', 'Korea', 'China', 'Japan'])
False
True
```

```
In [56]: a = 0
midterm_set = set([1, 5, 7, 4, 3, 2, 1, 1, 2, 3])
for i in midterm_set:
    a = a + i
print(a)
```

22

```
In [22]: def add_number(original_list):
        original_list += [1]
mylist = [1, 2, 3, 4]
add_number(mylist)
print(mylist) #add 는 extend 인데 set 이니까 중복제거하고 {}
print(set(mylist)) #set 이니까 중괄호!
```

```
[1, 2, 3, 4, 1]
{1, 2, 3, 4}
```

```
In [27]: a = [3, "apple", 2016, 4]
b = a.pop(0) #3 없어짐
print(a)
c = a.pop(1) #2016 없어짐

#자기리스트에서 pop 하면 없앤다. 뽑아낸거 저장하기위해서 a.pop 를 b 로 받음

print(a)

print(b + c)
```

```
['apple', 2016, 4]
['apple', 4]
2019
```

```
In [28]: dict_1 = {2:1, 4:2, 6:3, 8:4, 10:6}
dict_keys = list(dict_1.keys())
dict_values = list(dict_1.values())
dict_2 = dict()
for i in range(len(dict_keys)):
    dict_2[dict_values[i]] = dict_keys[i]
print(dict_2[2])
```

4

```
In [31]: mylist = ['pen', 'pencil', 'sharp']
list(enumerate(mylist)) #enumerate 값을 빼주는것 index 와 값을 세트로 보여줌
print(result)
#문자열로 바꾼다
k='&'.join(mylist) #join
print(k.upper().split('&')) #합한후, 대문자 바꾸기위해 쪼갬다 #split

[(0, 'pen'), (1, 'pencil'), (2, 'sharp')]
['PEN', 'PENCIL', 'SHARP']
```

```
In [34]: #####중요#

mylist = ['pen', 'pencil', 'sharp']
mylist2 = [i.upper() for i in mylist] #문자열에서 리스트 컴프리헨션으로 간단히 가
능(중요!!!)
print(mylist2)

['PEN', 'PENCIL', 'SHARP']
```

```
In [35]: #mylist 를 dictionary 로 변경 (많이사용!!!)
#dictionary comprehension
#key 는 인덱스 0 1 2 로 해라 >> 그러면 #enumerate #인덱스 값을 뽑아옴

d2={i:j for i,j in enumerate(mylist)}
print(d2)

{0: 'pen', 1: 'pencil', 2: 'sharp'}
```

```
In [36]: #{'pen':0, 'pencil':1, 'sharp':2}
d3={j:i for i,j in enumerate(mylist)}
print(d3)

{'pen': 0, 'pencil': 1, 'sharp': 2}
```

```
In [ ]: dict_1 = {2:1, 4:2, 6:3, 8:4, 10:6}
```

```
In [44]: name = "Hanbit University"
student = ["Hong", "Gil", "Dong"]
split_name = name.split()

print(split_name)
join_student = ''.join(student)
print(join_student)
print(join_student[-4:] + split_name[1])
```

```
['Hanbit', 'University']
HongGilDong
DongUniversity
```

```
In [47]: animal = ['Fox', 'Dog', 'Cat', 'Monkey', 'Horse', 'Panda', 'Owl']
ani2=[i for i in animal if 'o' not in i.lower()]
print([ani for ani in animal if 'o' not in ani])
print(ani2)
```

```
['Cat', 'Panda', 'Owl']
['Cat', 'Panda']
```

```
In [48]: result = [i for i in range(10) if i%2 == 0]
print(result)

items = 'zero one two three'.split("two")
result=[i for i in range(10)]
print(result)

items = 'zero one two three'.split()
print(items)

example = 'cs50.gachon.edu'
subdomain, domain, tld = example.split('.')
print(subdomain)
```

```
[0, 2, 4, 6, 8]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
['zero', 'one', 'two', 'three']
cs50
```

```
In [52]: #####중요##문자열로 어떻게 딕셔너리 바꿀까?
#문자열 --> 리스트로 바꾸어야한다

a = "my dog has brown eyes, my dog is cute"
b=a.split()
print(b)
d2={j:i for j,i in enumerate(b)}
print(d2)
d3={j:i for i,j in enumerate(a.split())}
print(d3)

['my', 'dog', 'has', 'brown', 'eyes,', 'my', 'dog', 'is', 'cute']
{0: 'my', 1: 'dog', 2: 'has', 3: 'brown', 4: 'eyes,', 5: 'my', 6: 'dog', 7: 'is', 8: 'cute'}
{'my': 5, 'dog': 6, 'has': 2, 'brown': 3, 'eyes,': 4, 'is': 7, 'cute': 8}
```

```
In [53]: #####중요##문자열로 어떻게 딕셔너리 바꿀까?
#문자열 --> 리스트로 바꾸어야한다
#키는 중복될수 없어서 같이게 나오면 뒤에것으로 나옴

a = "my dog has brown eyes, my dog is cute"
b=a.split()
print(b)
d2={j:i for j,i in enumerate(b)}
print(d2)
d3={j:i for i,j in enumerate(a.split())}
print(d3)

['my', 'dog', 'has', 'brown', 'eyes,', 'my', 'dog', 'is', 'cute']
{0: 'my', 1: 'dog', 2: 'has', 3: 'brown', 4: 'eyes,', 5: 'my', 6: 'dog', 7: 'is', 8: 'cute'}
{'my': 5, 'dog': 6, 'has': 2, 'brown': 3, 'eyes,': 4, 'is': 7, 'cute': 8}
```

```
In [58]: d1={'a':1, 'b':2, 'c':3}
#거꾸로 바꾸기
#{1:'a', 2:'b', 3:'c'}
d2={j:i for i,j in d1.items()}
print(d2)
print(d1.items())

{1: 'a', 2: 'b', 3: 'c'}
dict_items([('a', 1), ('b', 2), ('c', 3)])
```

```
In [61]: #midterm4

#개수세는 코드 외우기!!!!

#{a:xx번, b:?, c:? ...} key:문자 로 딕셔너리형태여야함
##count1[i]=count1[i]+1 #값이 비어있어서 이걸 못쓰니까

#count1[i]=count1.get(i,0)+1 로 딕셔너리 값이 비어있으니까 이렇게 하기!
#(입력한 문자열에 대해서 글자의 빈도수 세는 딕셔너리(키:밸류)로 표현하는 공식)
str1=input() #개수하나씩 세겠다 글자 빈도수세겠다
count1={}
for i in str1:
    count1[i]=count1.get(i,0)+1

print(count1)

aa b c ddd
{'a': 2, ' ': 3, 'b': 1, 'c': 1, 'd': 3}
```

```
In [65]: ##외우기

k=list(count1.items())
k.sort()
print(k)
#정렬하기

#letter_items=list(countTable.item)
count2={i:j for i,j in k}
print(count2)

[(' ', 3), ('a', 2), ('b', 1), ('c', 1), ('d', 3)]
{' ': 3, 'a': 2, 'b': 1, 'c': 1, 'd': 3}
```

```
In [71]: #머릿글 quiz 5

str2=input("머리글자를 만들 문장을 입력하세요:")

#acr1=[i for i in str2.split()]#i를 뽑아라
#i 첫글자만 뽑아라 i[0]

acr1=[i[0] for i in str2.upper().split()]
print(''.join(acr1)) #공백으로 조인!

머리글자를 만들 문장을 입력하세요:hello world
HW
```

In [73]: # quiz 6

```
#초기값이으니 get 필요없음 {"key1"=value1, "key2"=value2}
#{ } 에 딕셔너리에 >>> 숫자/문자/공백 를 만들수있음
```

```
str3=input()
```

```
result={"alpha":0,"digit":0, "space":0}
#틀완료 result={"alpha":0,"digit":0, "space":0}
#aaa bb 111 c
for i in str3:
    if i.isalpha():
        result["alpha"]=result["alpha"]+1
    elif i.isdigit():
        result["digit"]=result["digit"]+1
    elif i.isspace():
        result["space"]=result["space"]+1
print(result)
```

```
aaa 111 bbb ccc
{'alpha': 9, 'digit': 3, 'space': 0}
```


In [77]: # quiz 7

```
text="""
```

```
The wave of drones and missiles that flew towards Israel overnight on  
Sunday brought with it a new phase of tension, uncertainty and confron  
tation in the Middle East.
```

```
Iran launched the unprecedented attack in response to a suspected Isra  
eli strike on the Iranian consulate in Damascus, Syria, earlier this m  
onth.
```

```
It marked a new chapter in a discord between the two states that perco  
lated for years and has spiralled since Israel declared war on Hamas l  
ast October.
```

```
The next steps remain unclear, but Israel is facing pleas from its all  
ies to step back from the brink of open warfare and choose a path of d  
e-escalation.
```

```
Here's what you need to know.
```

```
""""
```

```
k=text.split()
```

```
print(k)
```

```
cnt=0
```

```
for i in k:
```

```
    if(i.lower()=="the"):
```

```
        cnt+=1
```

```
print(cnt)
```

```
#각 단어별로 몇개씩 나타나는지 만들어
```

```
#{ 단어: 갯수...}
```

```
dict2={}
```

```
for i in text.split():
```

```
    dict2[i]=dict2.get(i,0)+1
```

```
print(dict2)
```

['The', 'wave', 'of', 'drones', 'and', 'missiles', 'that', 'flew', 'to
wards', 'Israel', 'overnight', 'on', 'Sunday', 'brought', 'with', 'i
t', 'a', 'new', 'phase', 'of', 'tension,', 'uncertainty', 'and', 'conf
rontation', 'in', 'the', 'Middle', 'East.', 'Iran', 'launched', 'the',
'unprecedented', 'attack', 'in', 'response', 'to', 'a', 'suspected',
'Israeli', 'strike', 'on', 'the', 'Iranian', 'consulate', 'in', 'Damas
cus,', 'Syria,', 'earlier', 'this', 'month.', 'It', 'marked', 'a', 'ne
w', 'chapter', 'in', 'a', 'discord', 'between', 'the', 'two', 'state
s', 'that', 'percolated', 'for', 'years', 'and', 'has', 'spiralled',
'since', 'Israel', 'declared', 'war', 'on', 'Hamas', 'last', 'Octobe
r.', 'The', 'next', 'steps', 'remain', 'unclear,', 'but', 'Israel', 'i
s', 'facing', 'pleas', 'from', 'its', 'allies', 'to', 'step', 'back',
'from', 'the', 'brink', 'of', 'open', 'warfare', 'and', 'choose', 'a',
'path', 'of', 'de-escalation.', 'Here's', 'what', 'you', 'need', 'to',
'know.']

7

```
{'The': 2, 'wave': 1, 'of': 4, 'drones': 1, 'and': 4, 'missiles': 1,
'that': 2, 'flew': 1, 'towards': 1, 'Israel': 3, 'overnight': 1, 'on':
3, 'Sunday': 1, 'brought': 1, 'with': 1, 'it': 1, 'a': 5, 'new': 2, 'p
hase': 1, 'tension,': 1, 'uncertainty': 1, 'confrontation': 1, 'in':
4, 'the': 5, 'Middle': 1, 'East.': 1, 'Iran': 1, 'launched': 1, 'unpre
cedented': 1, 'attack': 1, 'response': 1, 'to': 3, 'suspected': 1, 'Is
raeli': 1, 'strike': 1, 'Iranian': 1, 'consulate': 1, 'Damascus,': 1,
'Syria,': 1, 'earlier': 1, 'this': 1, 'month.': 1, 'It': 1, 'marked':
1, 'chapter': 1, 'discord': 1, 'between': 1, 'two': 1, 'states': 1, 'p
ercolated': 1, 'for': 1, 'years': 1, 'has': 1, 'spiralled': 1, 'sinc
e': 1, 'declared': 1, 'war': 1, 'Hamas': 1, 'last': 1, 'October.': 1,
'next': 1, 'steps': 1, 'remain': 1, 'unclear,': 1, 'but': 1, 'is': 1,
'facing': 1, 'pleas': 1, 'from': 2, 'its': 1, 'allies': 1, 'step': 1,
'back': 1, 'brink': 1, 'open': 1, 'warfare': 1, 'choose': 1, 'path':
1, 'de-escalation.': 1, 'Here's': 1, 'what': 1, 'you': 1, 'need': 1,
'know.': 1}
```

In [78]: `# quiz 8.2`
`#item 로 쪼개기 d.items()`

```
d2={"JOHN":"jonh@GMAIL.com","KANG":"jykang@AJOU.AC.kr", "KIM":"KIM@nav
er.com"}
d3={i.upper():j.lower() for i,j in d2.items()}
print(d3)
```

```
{'JOHN': 'jonh@gmail.com', 'KANG': 'jykang@ajou.ac.kr', 'KIM': 'kim@na
ver.com'}
```

In [81]: #2개의 리스트를 zip(),dic() 이용해서 dictionary 로 변경!!

#공식 외우기!!!

```
a1=[100,200,300,400,500]
a2=["kang","park","kim","lee","lim"]
#k=zip(a1,a2)
#m=dict(k)
m=dict(zip(a1,a2))
print(m)
```

```
{100: 'kang', 200: 'park', 300: 'kim', 400: 'lee', 500: 'lim'}
```

In []: