

랜덤모듈의 활용 학습 활동을 마친 후 "어려웠던 개념" 에 관한 우리들의 아이디어!

오늘 어려웠던 개념, 이해가 가지 않은 내용이 있다면 무엇인가요? \*

count+=1

---

오늘 어려웠던 개념, 이해가 가지 않은 내용이 있다면 무엇인가요? \*

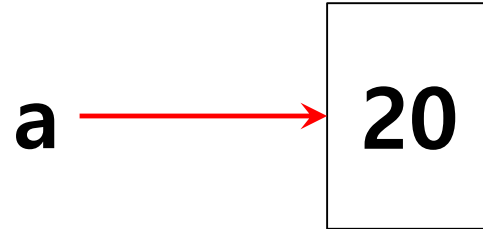
randrange와 randint의 범위가 다른 이유가 무엇인가요??

---

# count+=1 , 이것은 이렇습니다.

- 대입 연산자(=) : 변수에 값을 부여하는 것

a=20



변수 a에 20을 대입

- 대입 연산자의 축약(복합 대입 연산자) : 동일한 변수에 연산을 수행한 결과를 다시 대입할 경우

복합 대입 연산자	의미
a += 1	a = a + 1
a -= 1	a = a - 1
a *= 1	a = a * 1
a /= 1	a = a / 1
a //= 1	a = a // 1
a %= 1	a = a % 1

# randrange( )와 randint( ) 범위가 다른 이유는..?

- **randint(a,b) == randrange(a, b+1)**

파이썬 설치 경로의 random.py

( 예를 들면, C:\Users\사용자이름\Anaconda3\pkgs\python-3.6.5-h0c2934d\_0\Lib\random.py)  
의 내용을 살펴보세요~

- 언어는 변화하고, 발전합니다.

랜덤모듈의 활용 학습 활동을 마친 후 "랜덤 모듈 활용" 에 관한 **우리들의 아이디어!**

랜덤 모듈을 어디에 활용하면 좋을까요? 자신의 아이디어를 거침없이 적어 보세요~ \*

요즘 통계에서 표본을 임의로 추출하는데 이를 컴퓨터로 시켜볼 수도 있을 것 같다. 랜덤으로 리스트를 만들고 반복해서 이 평균이 얼마나 되나 해보고 싶다

뭔가 방향성이 있으면 완전 랜덤이 아니라는 소리고 불규칙하면 완전랜덤이라는 소린데 궁금하다

---

**파이썬 설치 경로의 random.py**

( 예를 들면, C:\Users\사용자이름\Anaconda3\pkgs\python-3.6.5-h0c2934d\_0\Lib\random.py)  
의 내용을 살펴보세요~

랜덤 모듈을 어디에 활용하면 좋을까요? 자신의 아이디어를 거침없이 적어 보세요~ \*

주사위 던지기 프로그램에 적용시킨다.

---



**#1. List**

**#2. Dictionary**

**#3. Sequence 자료형**

**#4. 개인별 프로젝트 제작하기  
(예 :To Do List 프로그램 만들기)**

# **#1. List**

내가 원하는 자료를 담을 수 있는 공간은 무엇이 있나요?



내가 원하는 자료를 담을 수 있는 공간은 무엇이 있나요?

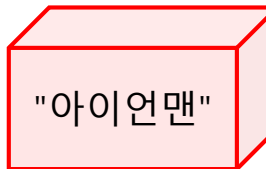
hero1="아이언맨"

hero2="토르"

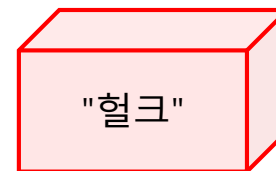
hero3="헐크"

hero4="캡틴아메리카"

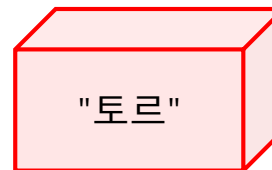
hero1



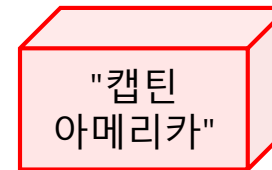
hero3



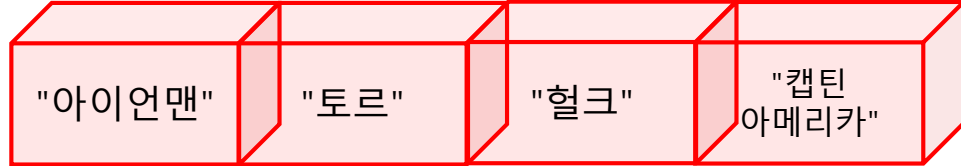
hero2



hero4

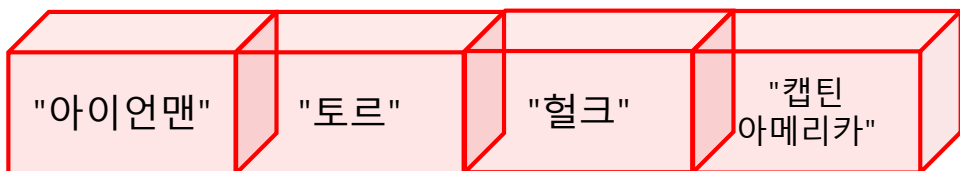


내가 원하는 자료를 묶어서 표현할 수 있을까?



# [List]

내가 원하는 자료를 묶어서 표현할 수 있을까?



```
heros = ["아이언맨", "토르", "헐크", "캡틴아메리카"]
```

#자료를 묶어서 저장할 수 있는 공간 : 리스트

```
heros = ["아이언맨", "토르", "헐크", "캡틴아메리카"]  
print(heros)
```

```
['아이언맨', '토르', '헐크', '캡틴아메리카']
```

```
heros = ["아이언맨", "토르", "헐크", "캡틴아메리카"]  
for item in heros :  
    print(item)
```

```
아이언맨  
토르  
헐크  
캡틴아메리카
```

# [List]

리스트는 다 묶을 수 있다!!!

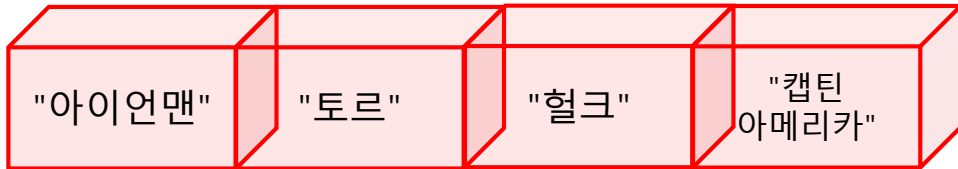
```
list1=[] #빈 리스트  
list2=[1,2,3,4,5] #숫자로 구성된 리스트  
list3=['a','b','c'] #문자로 구성된 리스트  
list4=[1,'a','abc',[1,2,3,4,5],('a','b','c')] #숫자, 문자, 리스트, 튜플을 모두 가진 리스트
```

\* 참고 : 심지어 이렇게도 묶을 수 있다

```
def myfunc() :  
    print("안녕하세요!")  
list4=[1,2,myfunc]
```

# [List]

묶여 있는 자료 중 **각각의 요소에 어떻게 접근**할까?

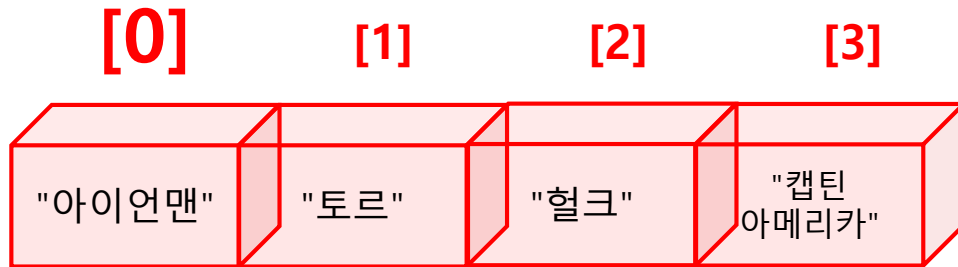


```
heros= ["아이언맨","토르","헐크","캡틴아메리카"]
```

```
heros=["아이언맨","토르","헐크","캡틴아메리카"]  
print(heros[0])  
print(heros[3])  
print(heros[4])
```

## [List]

묶여 있는 자료 중 **각각의 요소에 어떻게 접근**할까?

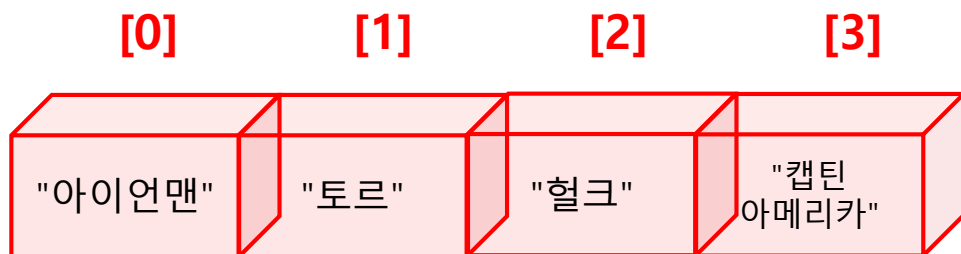


**heros=["아이언맨","토르","헐크","캡틴아메리카"]**

heros[0]      heros[1]      heros[2]      heros[3]

Four red arrows point from the labels heros[0], heros[1], heros[2], and heros[3] to the corresponding elements in the list: "아이언맨", "토르", "헐크", and "캡틴아메리카".

## [List : 리스트의 index]



```
heros=["아이언맨","토르","헐크","캡틴아메리카"]
```

```
heros=["아이언맨","토르","헐크","캡틴아메리카"]  
i=0  
while(i<4):  
    print(heros[i])  
    i=i+1
```

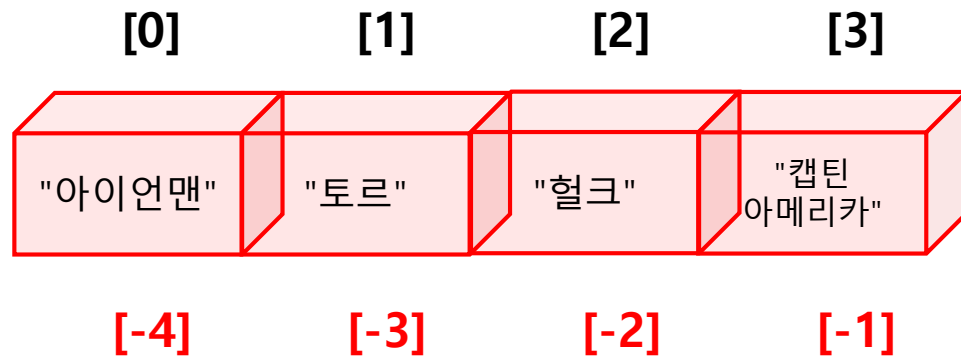
아이언맨  
토르  
헐크  
캡틴아메리카

```
heros=["아이언맨","토르","헐크","캡틴아메리카"]  
i=0  
while (i<4) :  
    print("heros[%d] : %s"%(i,heros[i]))  
    i+=1
```

heros[0] : 아이언맨  
heros[1] : 토르  
heros[2] : 헐크  
heros[3] : 캡틴아메리카

## [List : 리스트의 index]

index에 음수 값을 넣을 수도 있다



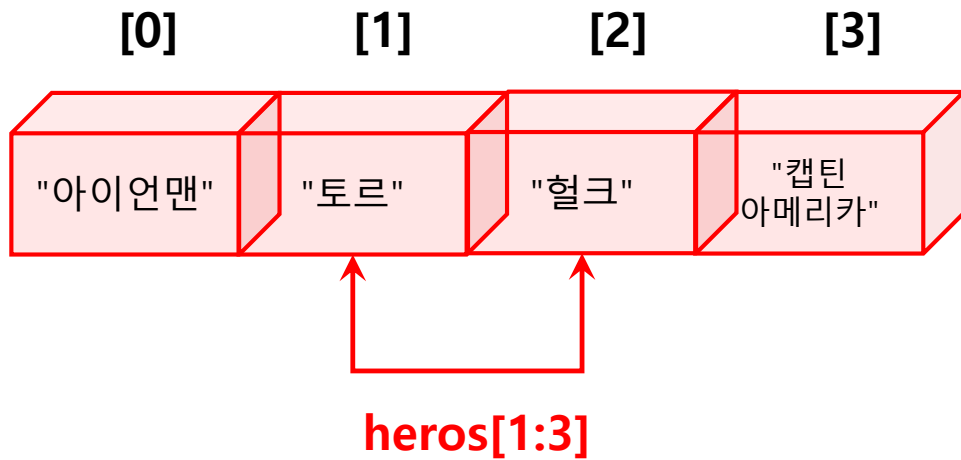
```
heros=["아이언맨","토르","헐크","캡틴아메리카"]  
print(heros[-1])  
print(heros[-2])  
print(heros[-3])  
print(heros[-4])
```

캡틴아메리카  
헐크  
토르  
아이언맨



# [List : 리스트의 index slicing]

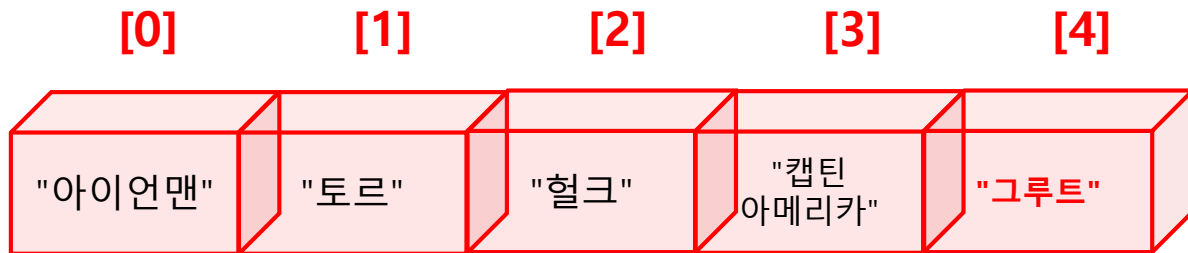
index를 잘라서 표현 할 수도 있다



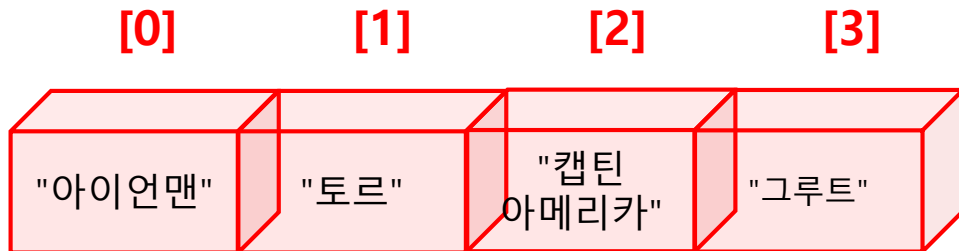
```
heros=["아이언맨","토르","헐크","캡틴아메리카","그루트"]  
print(heros[1:3])  
print(heros[-3:-1])  
print(heros[0:2])
```

```
['토르', '헐크']  
['헐크', '캡틴아메리카']  
['아이언맨', '헐크', '그루트']
```

## [List : 리스트에 요소를 추가/ 삭제하기]



```
heros=["아이언맨","토르","헐크","캡틴아메리카","그루트"]
```



```
heros=["아이언맨","토르","캡틴아메리카","그루트"]
```

```
heros=["아이언맨","토르"]  
print(heros)  
heros.append("그루트")  
print(heros)  
heros.remove("헐크")  
print(heros)
```

## [List의 함수]

함수	설명	사용법
append()	리스트 맨 뒤에 항목을 추가한다.	리스트명.append(값)
pop()	리스트 맨 뒤의 항목을 빼낸다(리스트에서 해당 항목이 삭제된다).	리스트명.pop()
sort()	리스트의 항목을 정렬한다.	리스트명.sort()
reverse()	리스트 항목의 순서를 역순으로 만든다.	리스트명.reverse()
index()	지정한 값을 찾아 해당 위치를 반환한다.	리스트명.index(찾을값)
insert()	지정된 위치에 값을 삽입한다.	리스트명.insert(위치, 값)
remove()	리스트에서 지정한 값을 삭제한다. 단 지정한 값이 여러 개면 첫 번째 값만 지운다.	리스트명.remove(지울값)
extend()	리스트 뒤에 리스트를 추가한다. 리스트의 더하기(+) 연산과 기능이 동일하다.	리스트명.extend(추가할리스트)
count()	리스트에서 해당 값의 개수를 센다.	리스트명.count(찾을값)
clear()	리스트의 내용을 모두 지운다.	리스트명.clear()
del()	리스트에서 해당 위치의 항목을 삭제한다.	del(리스트명[위치])
len()	리스트에 포함된 전체 항목의 개수를 센다.	len(리스트명)
copy()	리스트의 내용을 새로운 리스트에 복사한다.	새리스트=리스트명.copy()
sorted()	리스트의 항목을 정렬해서 새로운 리스트에 대입한다.	새리스트=sorted(리스트)

## [문제]

1부터 100까지의 숫자 중 홀수로만 구성된 30개의 원소가 있는 oddList를 만들고 이 중 첫 번째 값과 다섯 번째 값을 출력해 보자.

## [참고1] : Tuple

# ( )로 표현. 리스트와 비슷한 성질을 가지고 있지만 변경 불가능하다. 읽기만 가능  
# 변경되어서는 안되는 값을 tuple형으로 만들어서 관리하자!

```
tuple1=(1,2,3,4,5)
tuple2=('a','b','c')
tuple3=(1,'a','abc',[1,2,3,4,5])
def myfunc() :
    print("안녕하세요?")
tuple4=(1,2,myfunc)
```

## [참고2] : Set

- # { }로 표현. 리스트와 비슷한 성질을 가지고 있지만 중복 불가능하다.
- # 중복된 값을 배제하면서 자료를 모으고자 할 때 set형으로 만들어서 관리하자!
- # 교집합(&), 합집합(|), 차집합(-)과 같은 연산이 가능하다.

```
set1={1,1,2,2,3} #중복된 값을 set에 넣을 경우  
print(set1) # 출력 결과를 보면, 중복된 값은 포함되지 않음
```

```
{1, 2, 3}
```

## [문제]

**myList를 만들고**

**사용자로부터 문자열을 5개 입력받아 myList에 추가하기**

### <출력예시>

값 입력(1) : 파이썬

값 입력(2) : 공부는

값 입력(3) : 꿀잤

값 입력(4) : 경복고

값 입력(5) : SW특기자 가나요?

<myList의 내용 출력>

파이썬 공부는 꿀잤 경복고 SW특기자 가나요?

## [문제-solution]

```
1  #비어있는 myList를 생성한다.
2  myList=[]
3  #사용자로부터 값을 5회 입력받는다.
4  for i in range(0,5) :
5      str=input("값 입력(%d) : "%(i+1))
6      myList.append(str)
7  #myList의 내용 출력
8  print("<myList의 내용 출력>")
9  for item in myList :
10     print(item,end=" ")
```

값 입력(1) : 파이썬

값 입력(2) : 공부는

값 입력(3) : 꿀잼

값 입력(4) : 경복고

값 입력(5) : SW특기자 가나요?

<myList의 내용 출력>

파이썬 공부는 꿀잼 경복고 SW특기자 가나요?



## [문제-advanced]

### myList를 만들고

사용자로부터 값을 5개 입력받아 myList에 추가하기  
(단, 입력 값이 정수일 경우 정수형 데이터로 저장한다)

[참고] <https://docs.python.org/3/library/stdtypes.html#string-methods>

#### [입/출력 예시]

값 입력 : 파이썬

값 입력 : 2018

값 입력 : 공부는

값 입력 : 10

값 입력 : 꿀잤

<myList의 내용 출력>

['파이썬', 2018, '공부는', 10, '꿀잤']

## [문제]

candy\_bag에 다음과 같이 사탕이 들어가 있습니다.

하지만, 저는 박하맛이 싫어요!

**박하맛을 딸기맛으로 바꿔주세요!**

```
candy_bag = ['레몬맛', '박하맛', '딸기맛', '우유맛', '박하맛', '커피맛', '포도맛', '박하맛']
```

## [문제]

candy\_bag에 다음과 같이 사탕이 들어가 있습니다.

하지만, 저는 박하맛이 싫어요!

**박하맛을 딸기맛으로 바꿔주세요!**

```
candy_bag = ['레몬맛', '박하맛', '딸기맛', '우유맛', '박하맛', '커피맛', '포도맛', '박하맛']
```

[알고리즘 생각하기]

## [문제 : 박하맛이 싫어요!-solution]

```
1 candy_bag=[ '레몬맛', '박하맛', '딸기맛', '우유맛', '박하맛', '커피맛', '포도맛', '박하맛' ]
2 print("<candy_bag의 내용확인>")
3 for item in candy_bag :
4     print(item,end=" ")
5 print("\n박하맛이 싫어요! \n박하맛을 모두 딸기맛으로!!!")
6 i=0
7 while(i<len(candy_bag)) :
8     if(candy_bag[i]=="박하맛") :
9         candy_bag[i]="딸기맛"
10
11     i=i+1
12 print("<candy_bag의 내용 다시 확인>")
13 for item in candy_bag :
14     print(item,end=" ")
15
```

## [문제]

candy\_bag에 다음과 같이 사탕이 들어가 있습니다.

저는 딸기맛도 싫어요!

**안되겠어요! 딸기맛을 삭제 해 주세요!**

```
candy_bag = ['레몬맛', '딸기맛', '딸기맛', '우유맛', '딸기맛', '커피맛', '포도맛', '딸기맛']
```

[알고리즘 생각하기]

# [문제 : 딸기맛도 싫어요!-solution]

```
: import time
candy_bag=['레몬맛', '딸기맛', '딸기맛', '우유맛', '딸기맛', '커피맛', '포도맛', '딸기맛']
print("<candy_bag의 내용확인>")
for item in candy_bag :
    print(item,end=" ")
print("\n딸기맛도 싫어요! \n딸기맛을 모두 삭제할래요!")
i=0
while(i<len(candy_bag)) :
    time.sleep(1)
    print(candy_bag[i])
    if(candy_bag[i]=="딸기맛") :
        print("딸기맛 발견.....삭제")
        del candy_bag[i]
    else : i=i+1
print("<candy_bag의 내용 다시 확인>")
for item in candy_bag :
    print(item,end=" ")
```

<candy\_bag의 내용확인>

레몬맛 딸기맛 딸기맛 우유맛 딸기맛 커피맛 포도맛 딸기맛

딸기맛도 싫어요!

딸기맛을 모두 삭제할래요!

레몬맛

딸기맛

딸기맛 발견.....삭제

딸기맛

딸기맛 발견.....삭제

우유맛

딸기맛

딸기맛 발견.....삭제

커피맛

포도맛

## [문제]

영어 단어 시험 문제 출제를 위해 다음과 같이 예비 list 3개를 만들었다.

- list\_ex1=["risk", "issue", "test", "maintenance", "maturity"]
- list\_ex2=["security", "design", "systematic", "safety"]
- list\_ex3=["maintenance", "verification", "validation"]

위 세 가지 예비 list중 하나를 골라 영어 단어 시험 문제를 출제하려고 한다.  
아래의 <조건>에 따라 시험 문제로 쓰일 수 있는 list를 찾아내는 프로그램을 작성해 보자.

### <조건>

1. "maintenance"라는 단어를 포함해야 하고
2. 영어 단어 개수가 최소 5개는 되어야 한다.

# [문제 : 영어 단어 시험 리스트 선택- 알고리즘 생각하기]

<조건>

1. "maintenance"라는 단어를 포함해야 하고

"maintenance" in list

2. 영어 단어 개수가 최소 5개는 되어야 한다.

len(list) >= 5

3. 1과 2를 둘 다 만족

(조건1) and (조건2)



## [문제 : 영어 단어 시험 리스트 선택 - Solution]

```
list_ex1=["risk", "issue", "test", "maintenance", "maturity"]
list_ex2=["security", "design", "systematic", "safety"]
list_ex3=["maintenance", "verification", "validation"]

includeWord="maintenance"
wordCount=5
if includeWord in list_ex1 and len(list_ex1)>=wordCount :
    print("list_ex1는 영어 테스트로 사용할 수 있습니다")
if includeWord in list_ex2 and len(list_ex2)>=wordCount :
    print("list_ex2는 영어 테스트로 사용할 수 있습니다")
if includeWord in list_ex3 and len(list_ex3)>=wordCount :
    print("list_ex3는 영어 테스트로 사용할 수 있습니다")
```

## [문제]

find\_word라는 list 안에 technology라는 단어가 요소로 들어가 있다.

이 list 안에서 찾을 수 있는 단어를 사용자가 찾는 프로그램을 만들어 보자.

```
find_word = ['t', 'e', 'c', 'h', 'n', 'o', 'l', 'o', 'g', 'y']
```

### <출력>

1. 찾은 단어의 리스트
2. 찾은 단어의 수

```
['t', 'e', 'c', 'h', 'n', 'o', 'l', 'o', 'g', 'y']
```

```
찾은 단어 : ['no', 'echo', 'ten']
```

```
찾은 단어의 수: 3
```

## [문제 : 단어 찾기 프로그램 - Solution]

```
word="technology"
find_word=list(word)
word_list=list()
print(find_word)

word1=find_word[4:6]
word1="".join(word1) #문자 리스트를 문자열로 변환 : join()
word_list.append(word1)
word2=find_word[1:4]+find_word[5:6]
word2="".join(word2)
word_list.append(word2)
word3=find_word[:2]+find_word[4:5]
word3="".join(word3)
word_list.append(word3)

print("찾은 단어 :",word_list)
print("찾은 단어의 수:",len(word_list))
```

['t', 'e', 'c', 'h', 'n', 'o', 'l', 'o', 'g', 'y']

찾은 단어 : ['no', 'echo', 'ten']

찾은 단어의 수: 3

## [문제]

여러 가지 기분을 리스트에 넣어놓고  
랜덤으로 오늘의 기분을 알려주는

포춘 쿠키(fortune cookie) 프로그램을 만들어 보자.

```
#####  
#Today's Feeling#  
#####  
  
1  
2  
3  
" fortunate "
```

## [문제 : 포춘 쿠키 프로그램 - solution]

```
import random
import time

goodsay=['happy','love','sad','hot','fortunate']

todayFeeling = random.choice(goodsay)
time.sleep(1)
print("#*17)
print("#Today's Feeling#")
print("#*17)
print("")
for i in range(1,4) :
    print(i)
    time.sleep(1)
print("###",todayFeeling,"###")
```

## **#2. Dictionary**

# [Dictionary]

# 키와 값을 하나의 요소로 하는 순서가 없는 집합

# {key : value} 쌍으로 정의

키	값
학번	1000
이름	홍길동
학과	컴퓨터학과

# [Dictionary]

# 키와 값을 하나의 요소로 하는 순서가 없는 집합

# {key : value} 쌍으로 정의

키	값
학번	1000
이름	홍길동
학과	컴퓨터학과

```
student=  
{ '학번':1000,  
  '이름':'홍길동',  
  '학과':'컴퓨터학과' }
```



# [Dictionary의 자료(item)들을 다루기]

# 추가 : **key와 value 쌍으로 추가**

```
student={'학번':1000, '이름':'홍길동', '학과':'컴퓨터학과'}  
student['성별']='남자' #딕셔너리에 새로운 항목 추가  
print(student)
```

{'학번': 1000, '이름': '홍길동', '학과': '컴퓨터학과', '성별': '남자'}

# 삭제 : **del 딕셔너리명[key]**

```
student={'학번':1000, '이름':'홍길동', '학과':'컴퓨터학과', '성별':'남자'}  
print(student)  
del student['성별'] #딕셔너리 항목 삭제  
print(student)
```

{'학번': 1000, '이름': '홍길동', '학과': '컴퓨터학과', '성별': '남자'}  
{'학번': 1000, '이름': '홍길동', '학과': '컴퓨터학과'}

## [Dictionary - 키(key)]

# 딕셔너리는 key값을 통해 value를 찾는 구조이므로

# key는 고유한 값이어야 한다.

# key는 변경할 수 없는 자료형만 가능

# key들로 이루어진 리스트를 얻을 때 : `keys()`

## [Dictionary - 값(value)]

# string형, list형, tuple형 등 모든 자료형을 값으로 사용

# value값은 대입문(=)으로 변경 가능

# value들로 이루어진 리스트를 얻을 때 : values()

## [Dictionary - 출력하기]

```
student={'학번':1000, '이름':'홍길동', '학과':'컴퓨터학과'}
```

```
student={'학번':1000, '이름':'홍길동', '학과':'컴퓨터학과'}
student['학번']='std1000'
print(student)
print(">>>키 출력")
for key in student.keys():
    print(key,end=" ")

print("\n>>>값 출력")
for value in student.values():
    print(value,end=" ")

print("\n>>>키-값 출력")
for key, value in student.items():
    print(key,"-",value)
```

## [문제]

어느 커피숍에는 메뉴가 4가지 있다.

Americano, Café latte, Green Tea latte, Mocha late  
각 메뉴의 가격은 2000원, 2500원, 3000원, 3500원이다.

이 목록을 딕셔너리로 작성하고  
사용자가 입력한 메뉴가 있는지 확인해 보자.

### <출력 예시>

메뉴 입력 : Americano

입력하신 메뉴( Americano )은(는) 메뉴에 있습니다.

가격은 2000 원 입니다

## [문제 - Solution]

```
menu={"Americano":2000, "Cafe latte":2500,"Green Tea latte":3000, "Mocha latte" : 3500}

user_input=input("메뉴 입력 :")
if user_input in menu.keys():
    print("입력하신 메뉴(",user_input,")은(는) 메뉴에 있습니다.")
    print("가격은", menu[user_input],"원 입니다")
else :
    print(user_input,"은 메뉴에 없습니다.")
```

메뉴 입력 :Americano

입력하신 메뉴( Americano )은(는) 메뉴에 있습니다.

가격은 2000 원 입니다

## [문제]

한 판매 회사에서 판매 실적 점수와 고객 평가 점수를 통해  
우수 제품을 선별하고자 한다.

### <조건>

1. 판매 실적 점수가 4 이상이고, 고객 평가 점수가 4이상인 제품은 우수 제품이다.
2. 두 점수 모두 4미만인 제품은 판매 중지 목록에 들어간다.
3. 우수 제품 출력: 제품명, 판매실적, 고객평가 출력 / 판매 중지 품목 출력 : 제품명

제품명	판매실적	고객평가
비누	3	2
칫솔	5	4
샴푸	2	1
치약	4	4
로션	5	3

## [문제]

한 판매 회사에서 판매 실적 점수와 고객 평가 점수를 통해  
우수 제품을 선별하고자 한다.

### <출력 예시>

<<우수 제품 목록>>

칫솔 : 판매실적 :5 고객평가 4

치약 : 판매실적 :4 고객평가 4

<<판매 중지 목록>>

비누

샴푸



# [문제 - solution]

```
product=["비누","칫솔","샴푸","치약","로션"]
sales=[3,5,2,4,5]
consumer=[2,4,1,4,3]
goodProduct={}
badProduct={}
tmpList=[]
for i in range(0,len(product)) :
    tmpList.append(sales[i])
    tmpList.append(consumer[i])
    if sales[i]>=4 and consumer[i]>=4 :
        goodProduct[product[i]]=tmpList
    elif sales[i] < 4 and consumer[i] < 4 :
        badProduct[product[i]]=tmpList
    else :
        continue
    tmpList=[]
print("<<우수 제품 목록>>")
for key,value in goodProduct.items() :
    print("%s : 판매실적 :%d 고객평가 %d"%(key,value[0],value[1]))
print("<<판매 중지 목록>>")
for key in badProduct.keys() :
    print("%s"%(key))
```

<<우수 제품 목록>>

칫솔 : 판매실적 :5 고객평가 4

치약 : 판매실적 :4 고객평가 4

<<판매 중지 목록>>

비누

샴푸

[문제]

각 나라 이름의 수도를 맞추는 게임 작성하기



이미지 출처 : <http://fiancée.tistory.com/>

# [문제-여러 나라의 수도 맞추는 게임]

[알고리즘 생각하기]

# [문제-여러 나라의 수도 맞추는 게임]

## <출력 예시>

<<<나라의 수도 맞추는 게임>>>

3초 후에 나타나는 나라의 수도를 맞추세요

3

2

1

브라질

>>>>기회는 7번

++++1번째 시도

수도 입력: 브라질

다시 생각해 보세요(기회는 6번 남았습니다.)

++++2번째 시도

수도 입력: 브라질리아

정답입니다!!!

2회에 맞추셨네요!

실력이 좋네요!!

## **#3. Sequence 자료형**

# [Sequence 자료형]

객체가 **순서를 가지고 나열되어** 있는 것

# [Sequence 자료형]

객체가 순서를 가지고 나열되어 있는 것

- 문자열 : 'abced'
- 리스트 : [1,[2,3],'안녕']
- 튜플 : (100, 200, 300)

## [Sequence 자료형의 특징]

특성	설명
인덱싱	<b>인덱스</b> 를 통하여 값에 접근. 인덱스는 0부터 시작
슬라이싱	<b>특정 구간</b> 의 값을 취할 수 있음. 구간은 시작 인덱스와 끝 인덱스로 정의함
연결	'+'연산자를 이용해 두 시퀀스 자료를 <b>연결</b> 할 수 있음
반복	'*'연산자를 이용해 시퀀스 자료를 여러 번 <b>반복</b> 할 수 있음
멤버체크	'in' 키워드를 사용하여 특정 값이 시퀀스 자료의 <b>요소인지 확인</b> 할 수 있음
크기정보	<b>len( )</b> 을 이용해 시퀀스 <b>자료의 크기</b> 를 알 수 있음 문자열은 문자의 개수, 리스트와 튜플은 멤버의 개수



## [Sequence : indexing]

시퀀스 자료형에서 **인덱스**를 통해 값을 얻는 방법

```
1  strdata='Time is money!'  
2  listdata=[1,2,[1,2,3]]  
3  print(strdata[5])  
4  print(listdata[0])  
5  print(listdata[-1])  
6  print(listdata[2][-1])
```

# [Sequence : indexing]

strdata	T	i	m	e		i	s		m	o	n	e	y	!
인덱스	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

# [Sequence : slicing]

# 시퀀스 자료의 일부 범위를 취하는 것

# [시작 인덱스 : 끝 인덱스 : 스텝] 으로 표현. 스텝은 생략 가능.

# 시작 인덱스  $\leq$  [시작 인덱스 : 끝 인덱스]  $<$  끝 인덱스

```
1  strdata='Time is money!'
2  print(strdata[0:5])
3  print(strdata[:7])
4  print(strdata[8:])
5  print(strdata[:-3])
6  print(strdata[-3:])
7  print(strdata[:])
8  print(strdata[::2])
```

## [Sequence : 연결(+)]

# **자료형이 동일한** 두 개의 시퀀스 자료는 **+연산자로 연결** 가능

```
strdata1='I love '  
strdata2='Python '  
strdata3='꿀잼 '  
print(strdata1+strdata2)  
print(strdata2+strdata3)
```

```
listdata1=[1,2,3]  
listdata2=[4,5,6]  
listdata3=listdata2+listdata1  
print(listdata3)
```

## [Sequence : 반복(\*)]

# 시퀀스 자료는 \*연산자로 n번 반복 가능

```
strdata1="Gee"  
strdata2="baby"  
print(strdata1*3, strdata2*3)  
lstdata1=[1,2,3]  
lstdata2=lstdata1*2  
print(lstdata2)
```

## [Sequence : len( )으로 자료의 크기 알기]

# 시퀀스 자료의 크기 : 자료를 구성하는 **멤버(요소)의 개수**

```
strdata1="python"
strdata2="꿀잼"
listdata=['kyung', 'bock', strdata1, strdata2]
print("len(strdata1) :", len(strdata1))
print("len(strdata2): ", len(strdata2))
print("len(listdata): ", len(listdata))
```

## [Sequence : in으로 멤버(요소) 확인하기]

# **in**은 시퀀스 자료에 어떤 값이 있는지 없는지 확인할 때 쓰인다.

# <값> **in** <자료>

```
listdata=['a','b','c',1,2,3]
if 'a' in listdata :
    print("ㄷ'aㄷ가 listdata에 있습니다.")
    print(listdata)
else :
    print("ㄷ'aㄷ가 listdata에 없습니다.")
```

```
from time import sleep
for i in range(100) :
    msg='ㄷr진행률%d%%'%(i+1)
    print(' '*len(msg),end='')
    print(msg,end='')
    sleep(0.1)
```

## **#4. 개인별 프로젝트 제작**

**(예 : To do List 프로그램 만들기)**



# (To do List 프로그램 만들기)

## <알고리즘 생각하기 >

1. 해야 할 일의 목록을 정리 한다. (수학, 영어, 컴퓨터)
2. 각 항목 별로 해야할 일을 날짜를 키 값으로 하여 할 일을 딕셔너리에 저장한다.
3. 사용자에게 날짜를 입력 받아 해당 날짜에 해야 할 일을 출력한다.

## [수업 7차시-정리하기]

<https://goo.gl/forms/j67419BF6xVH9Mx62>