



**#1. List**

**#2. Sequence 자료형**

**#3. '스마트 냉장고' 프로그램 제작**

# **#1. List**

내가 원하는 자료를 답을 수 있는 공간은 무엇이 있나요?

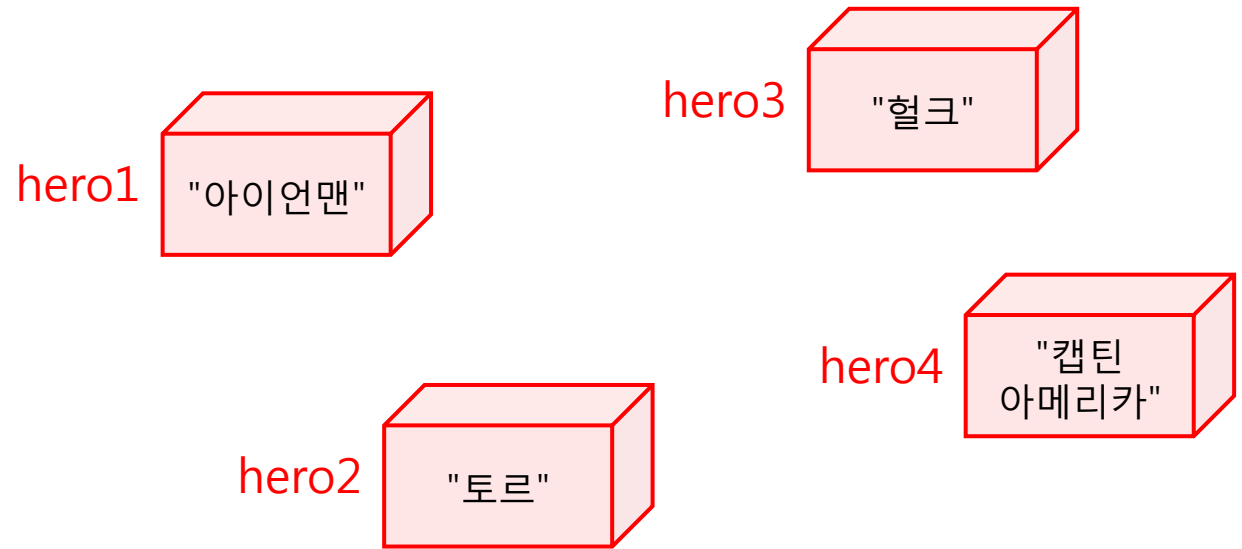
내가 원하는 자료를 담을 수 있는 공간은 무엇이 있나요?

hero1="아이언맨"

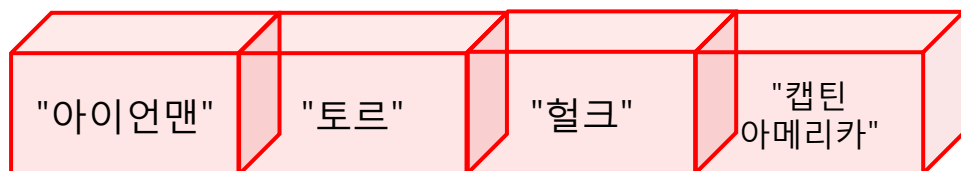
hero2="토르"

hero3="헐크"

hero4="캡틴아메리카"

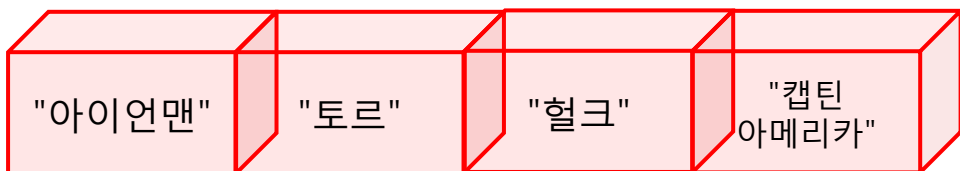


내가 원하는 자료를 묶어서 표현할 수 있을까?



# [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
```

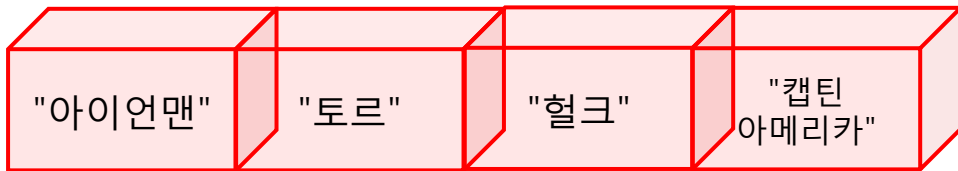
\* 참고 : 심지어 이렇게도 묶을 수 있다

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



# [List]

묶여 있는 자료 중 **각각의 요소에 어떻게 접근**할까?

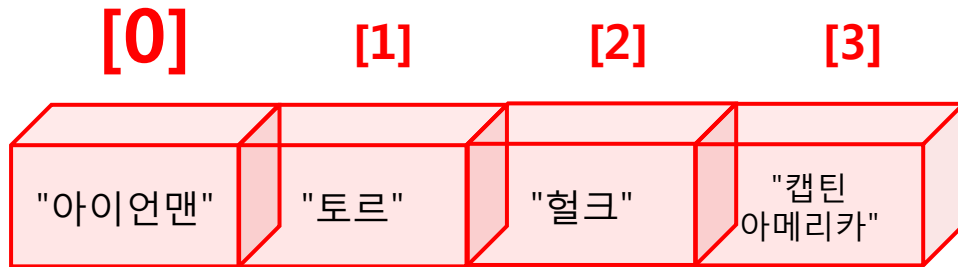


```
heros= ["아이언맨","토르","헐크","캡틴아메리카"]
```

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

## [List]

묶여 있는 자료 중 **각각의 요소에 어떻게 접근**할까?

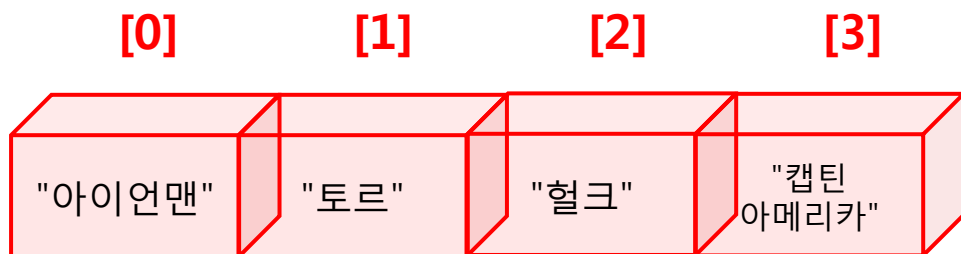


**heros=["아이언맨","토르","헐크","캡틴아메리카"]**

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
```

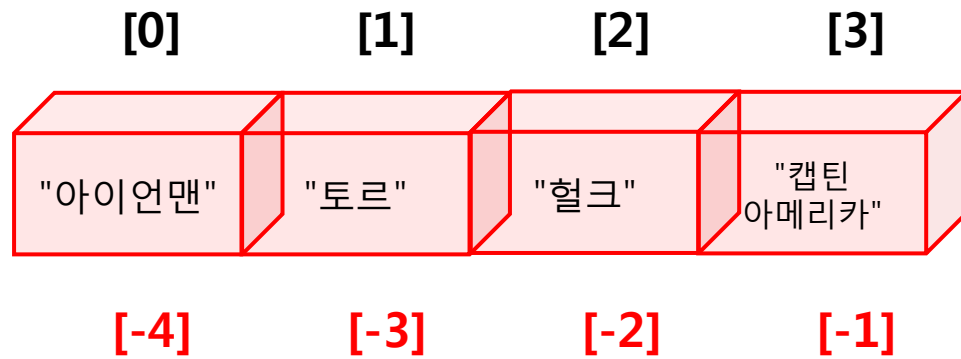
아이언맨  
토르  
헐크  
캡틴아메리카

```
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

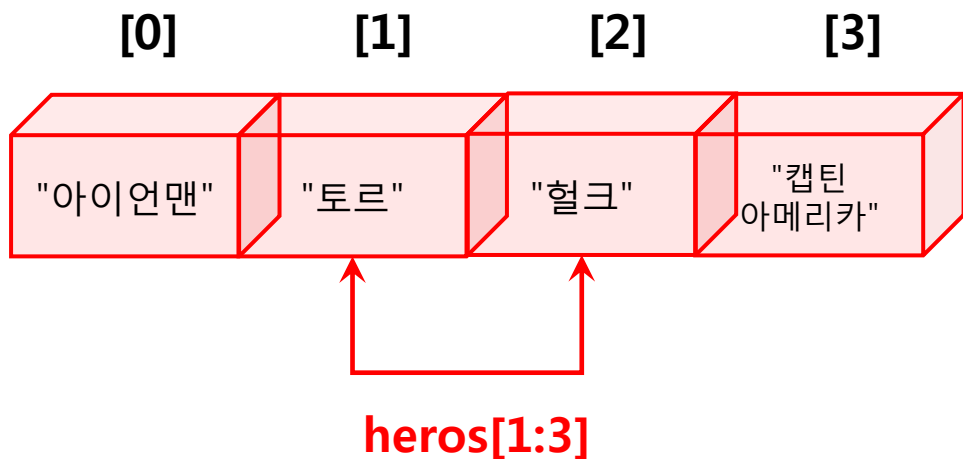


```
heros=["아이언맨","토르","헐크","캡틴아메리카"]  
print(heros[-1])  
print(heros[-2])  
print(heros[-3])  
print(heros[-4])
```

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']
```

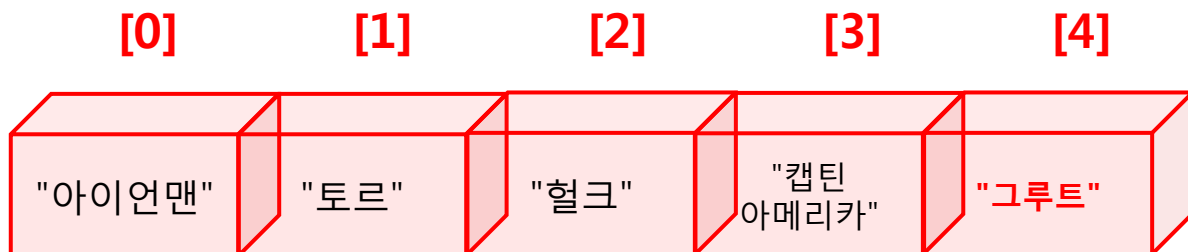
## # 시퀀스 자료형의 슬라이싱

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

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

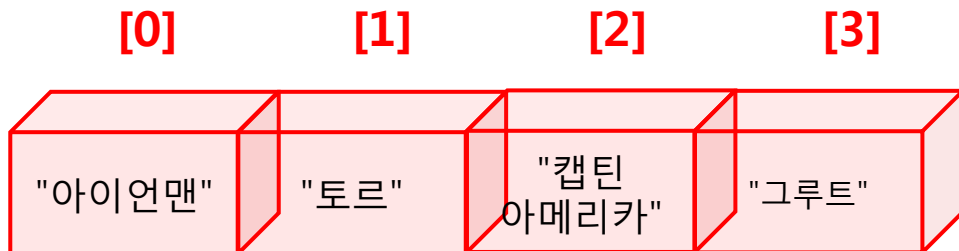
# 시퀀스 자료의 첫 번째 요소의 인덱스는 0

## [List : 리스트에 요소를 추가/ 삭제하기]



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

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



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

# [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.

## [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]

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]

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. 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 범위**를 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("##'a##'가 listdata에 있습니다.")
    print(listdata)
else :
    print("##'a##'가 listdata에 없습니다.")
```

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

**[참고 : List Vs. Tuple Vs. Set]**

## [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}
```

# **#3. 스마트 냉장고 프로그램 제작**



# (스마트 냉장고 프로그램)

<알고리즘 생각하기 >

1. 냉장고에 식품을 차례대로 넣는다
  2. 냉장고에 있는 식품과 갯수 등 필요한 정보를 보여준다
- ...각자 알고리즘을 생각해서 구상해 보세요~^^

## [수업 7차시-정리하기]

<https://forms.gle/zCHkd8oTUPxEUHt1A>