

# Python 기본 문법(2)



# range() 함수

- range() 함수

- 정수 범위를 표현
- 시작 숫자와 끝 숫자를 지정했을 때 끝 숫자는 범위에 포함되지 않음

- range(숫자)  
: 0부터 숫자미만 까지 알려줍니다.
- range(시작숫자, 끝숫자)  
: 시작 숫자부터 끝숫자 전 까지 알려줍니다.

예제	파라미터 의미	리턴값
range(3)	Stop	0, 1, 2
range(3,6)	Start, Stop	3, 4, 5
range(2,11,2)	Start, Stop, Step	2, 4, 6, 8, 10

# 리스트(list) 생성

- 변수에 값을 저장할 때 [ ](대괄호)로 묶고 각 값을 ,(콤마)로 구분
  - 요소(element) : 리스트에 저장된 각 값

```
#리스트 생성
lst1 = []
lst2 = list()
lst3 = list(range(1, 11, 2))
lst4 = list(range(11, 1, -2))

print(lst1)
print(lst2)
print(lst3)
print(lst4)
```

```
[]
[]
[1, 3, 5, 7, 9]
[11, 9, 7, 5, 3]
```

# 리스트(list) 연산 / 인덱싱

*#리스트 연산*

```
lst = lst3 + lst4  
print(lst)
```

```
lst = lst3 * 3  
print(lst)
```

*#리스트 갯수*

```
print("리스트 전체 개수:", len(lst))
```

*#특정항목 개수*

```
print("3의 개수:", lst.count(3))
```

[1, 3, 5, 7, 9, 11, 9, 7, 5, 3]

[1, 3, 5, 7, 9, 1, 3, 5, 7, 9, 1, 3, 5, 7, 9]

리스트 전체 개수: 15

3의 개수: 3

*#리스트 인덱싱/ 슬라이싱*

```
lst = list(range(5))  
print("리스트 :", lst)  
print("리스트 첫번째 항목 :", lst[0])  
print("리스트 마지막 항목 :", lst[-1])
```

```
lst[1] = 2  
print("리스트 항목 수정 :", lst)
```

```
print("2번째에서 3번째 :", lst[2:4])  
print("처음부터 3개 항목 :", lst[:3])  
print("두번째에서 마지막까지 :", lst[1:])
```

리스트 : [0, 1, 2, 3, 4]

리스트 첫번째 항목 : 0

리스트 마지막 항목 : 4

리스트 항목 수정 : [0, 2, 2, 3, 4]

2번째에서 3번째 : [2, 3]

처음부터 3개 항목 : [0, 2, 2]

두번째에서 마지막까지 : [2, 2, 3, 4]



# 리스트 항목 추가/삭제

```
#리스트 원소 추가/ 삭제
lst = list(range(1,6))
print("리스트 :", lst)

#마지막에 리스트 추가
lst.append(100)
print("리스트 :", lst)

#특정 위치에 추가
lst.insert(1,100)
print("리스트 :", lst)

#리스트 삭제 => 항목이 없으면 오류
lst.remove(100)
print("리스트 :", lst)

#특정 위치 항목 삭제
idx = lst.index(100) #항목 값 찾기
del lst[idx]
print("리스트 :", lst)

#리스트 마지막 항목 삭제 및 특정 항목 삭제
lst.pop()
print("리스트 :", lst)

lst.pop(0)
print("리스트 :", lst)
```

```
리스트 : [1, 2, 3, 4, 5]
리스트 : [1, 2, 3, 4, 5, 100]
리스트 : [1, 100, 2, 3, 4, 5, 100]
리스트 : [1, 2, 3, 4, 5, 100]
리스트 : [1, 2, 3, 4, 5]
리스트 : [1, 2, 3, 4]
리스트 : [2, 3, 4]
```



# 파이썬 랜덤 모듈

- 난수(random number)를 사용하기 위해서는 기본적으로 제공되는 random 모듈을 사용
  - **import random**
  - random() : 0 부터 1 사이의 부동소수점(float) 숫자를 리턴
  - randint(최소, 최대) : 입력 파라미터인 최소부터 최대까지 중 임의의 정수를 리턴
  - uniform(최소, 최대) : 입력 파라미터인 최소부터 최대까지 중 임의의 부동소수점(float) 숫자를 리턴
  - randrange(시작, 끝[, 간격]) : 입력 파라미터인 시작부터 끝 값까지 (지정된 간격으로 나열된) 숫자 중 임의의 정수를 리턴
  - sample(컬렉션, 샘플수) : 지정된 컬렉션으로부터 샘플수만큼 랜덤 추출을 하는 함수
  - shuffle(L): 전달받은 시퀀스 자료형 변수 내용을 임의의 순서대로(랜덤으로) 섞음



# 리스트 정렬/뒤집기/섞기

```
#정렬, 뒤집기  
lst = list(range(1,6))  
print("리스트 :", lst)
```

```
#뒤집기  
lst.reverse()  
print("리스트 :", lst)
```

```
#정렬  
lst.sort()  
print("리스트 :", lst)
```

리스트 : [1, 2, 3, 4, 5]

리스트 : [5, 4, 3, 2, 1]

리스트 : [1, 2, 3, 4, 5]

```
#랜덤 모듈 사용
```

```
import random as r
```

```
#0부터 1사이의 부동소수점 숫자 생성
```

```
rnum = r.random()
```

```
print("랜덤수 : ", rnum)
```

```
#최소부터 최대의 일의의 정수 출력
```

```
rnum = r.randint(1,10)
```

```
print("랜덤수 : ", rnum)
```

```
#특정 범위에서 추출
```

```
rnum = r.randrange(100,200)
```

```
print("랜덤수 : ", rnum)
```

```
#샘플링 추출
```

```
rnum = r.sample(range(1,46), 6)
```

```
print("랜덤수 : ", rnum)
```

```
#리스트 항목 섞기
```

```
r.shuffle(rnum)
```

```
print("랜덤수 : ", rnum)
```

랜덤수 : 0.8121329175880048

랜덤수 : 8

랜덤수 : 164

랜덤수 : [21, 23, 39, 32, 19, 9]

랜덤수 : [19, 32, 23, 9, 39, 21]



# 해결문제

- 다음은 문재인 대통령의 연설문 일부입니다.
  - 문단을 나열하고 문단의 수를 구하시오.

국민 여러분의 위대한 선택에 머리 숙여 깊이 감사드립니다. 저는 오늘 대한민국 제19대 대통령으로서 새로운 대한민국을 향해 첫걸음을 내딛습니다. 지금 제 두 어깨는 국민 여러분으로부터 부여받은 막중한 소명감으로 무겁습니다. 지금 제 가슴은 한번도 경험하지 못한 나라를 만들겠다는 열정으로 뜨겁습니다. 그리고 지금 제 머리는 통합과 공존의 새로운 세상을 열어갈 청사진으로 가득 차 있습니다.



# 튜플(tuple)

- 변경할 수 없는 순서 있는 객체의 집합
  - 리스트와 비슷하지만 한번 생성되면 변경할 수 없음
  - ()를 사용하여 정의

```
tpl = ()
print("튜플 생성 :", tpl)

tpl = tuple()
print("튜플 생성 :", tpl)

tpl = tuple(range(1,6))
print("튜플 생성 :", tpl)

print("튜플 :", tpl)
print("튜플 첫번째 항목 :", tpl[0])
print("튜플 마지막 항목 :", tpl[-1])

print("2번째에서 3번째 :", tpl[2:4])
print("처음부터 3개 항목 :", tpl[:3])
print("두번째에서 마지막까지 :", tpl[1:])

#항목 추가
tpl = tpl + (100,)
print("튜플 :", tpl)

#항목 수정 불가
tpl[1] = 2
print("튜플 항목 수정 :", tpl)
```

```
튜플 생성 : ()
튜플 생성 : ()
튜플 생성 : (1, 2, 3, 4, 5)
튜플 : (1, 2, 3, 4, 5)
튜플 첫번째 항목 : 1
튜플 마지막 항목 : 5
2번째에서 3번째 : (3, 4)
처음부터 3개 항목 : (1, 2, 3)
두번째에서 마지막까지 : (2, 3, 4, 5)
튜플 : (1, 2, 3, 4, 5, 100)
```

```
TypeError                                Traceback (most recent call last)
<ipython-input-62-ace6ac3ff93c> in <module>()
    21
    22 #항목 수정 불가
→   23 tpl[1] = 2
    24 print("튜플 항목 수정 :", tpl)

TypeError: 'tuple' object does not support item assignment
```

# 딕셔너리(dictionary)

- Key와 Value라는 것을 한 쌍으로 갖는 자료형
  - {Key1:Value1, Key2:Value2, Key3:Value3, ...}
  - 리스트나 튜플처럼 순차적으로(sequential) 해당 요소값을 구하지 않고 Key를 통해 Value를 얻음
  - Key는 고유한 값이므로 중복되는 Key 값을 설정해 놓으면 하나를 제외한 나머지 것들이 모두 무시



# 딕셔너리 다루기

```
#딕셔너리 생성  
dic = {}  
print("딕셔너리 :", dic)
```

```
#딕셔너리 항목 추가  
dic['a'] = 1  
dic['b'] = 2  
dic['c'] = 3  
print("딕셔너리 :", dic)
```

```
#딕셔너리 키 가져오기  
print("딕셔너리 키: ", dic.keys())
```

```
#딕셔너리 값 가져오기  
print("딕셔너리 값: ", dic.values())
```

```
#딕셔너리 항목 가져오기  
print("딕셔너리 항목: ", dic.items())
```

```
#딕셔너리 항목 삭제  
del dic['a']  
print("딕셔너리 :", dic)
```

```
#딕셔너리 전체 삭제  
dic.clear()  
print("딕셔너리 :", dic)
```

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



# 해결문제

- 다음은 문재인 대통령의 연설문 일부입니다.

– 국민, 대한민국, 대통령이 각각 몇 번 나왔는지 딕션너리를 이용하여 표시하시오.

국민 여러분의 위대한 선택에 머리 숙여 깊이 감사드립니다. 저는 오늘 대한민국 제19대 대통령으로서 새로운 대한민국을 향해 첫걸음을 내딛습니다. 지금 제 두 어깨는 국민 여러분으로부터 부여받은 막중한 소명감으로 무겁습니다. 지금 제 가슴은 한번도 경험하지 못한 나라를 만들겠다는 열정으로 뜨겁습니다. 그리고 지금 제 머리는 통합과 공존의 새로운 세상을 열어갈 청사진으로 가득 차 있습니다.