

파이썬 기초

## 08 리스트와 튜플



홍필두 교수  
(파이썬기초)



## 학습내용

- 01 리스트
- 02 리스트 관리
- 03 튜플





## 학습목표

- 컬렉션 형의 리스트에 대하여 이해하고 사용할 수 있다.
- 리스트에 대한 요소를 삽입, 삭제, 조회, 정렬할 수 있다.
- 컬렉션 형의 튜플에 대하여 이해하고 사용할 수 있다.



# 사전 학습

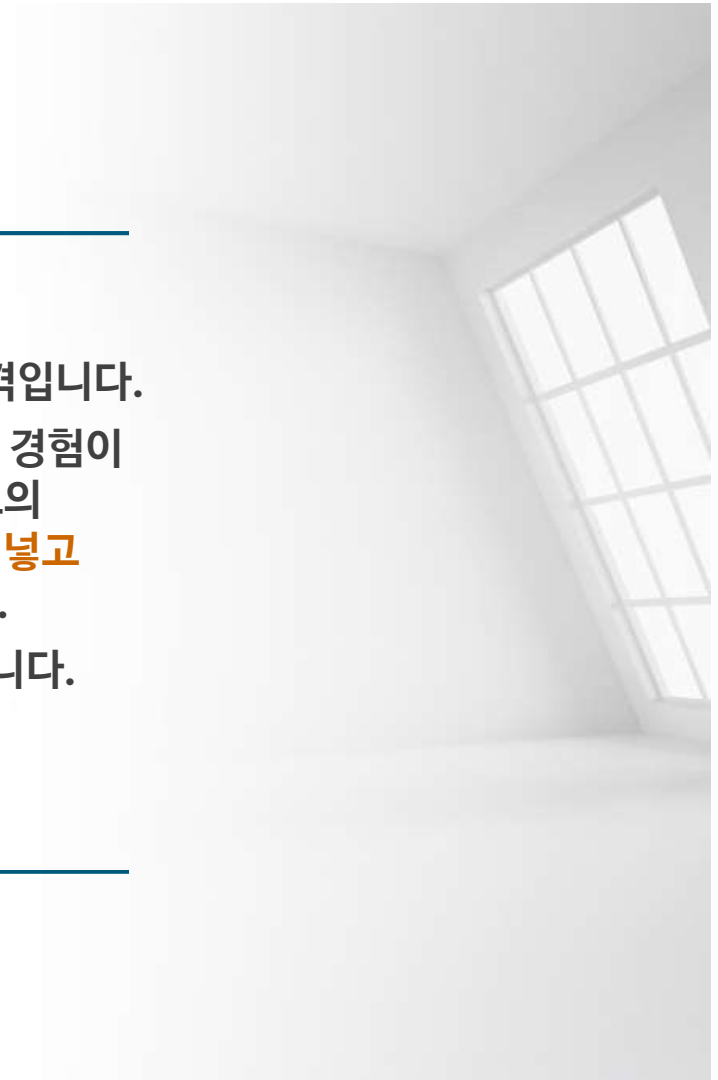
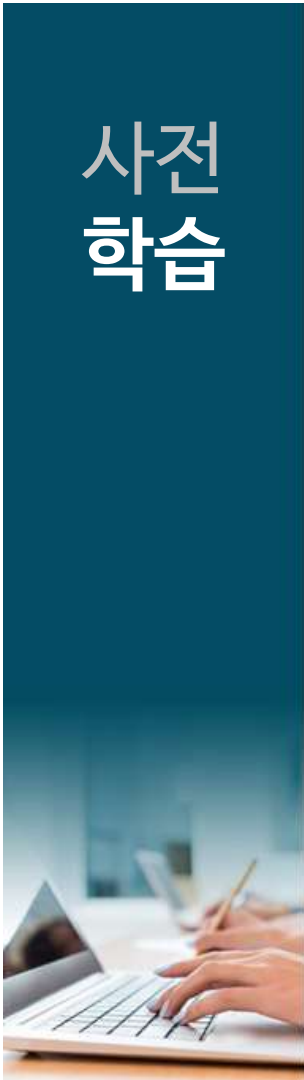
## “파이썬의 특징적인 자료형”

파이썬에서 매우 특징적인 자료형으로  
리스트가 있습니다.

리스트는 다른 언어의 ArrayList와 유사한 성격입니다.

컴퓨터알고리즘, 자료구조라는 과목을 들어본 경험이  
있거나 관심이 있다면 리스트, 큐, 환형 리스트의  
구조를 들어봤을 것입니다. 즉, **하나의 자료를 넣고  
꺼내고 삭제하고 처리하는 자료의 형태**입니다.

이번 강의는 조금 어렵게 느껴질 수 있을 것입니다.  
먼저 자료구조의 리스트, 큐에 대하여  
검색을 통하여 미리 알아보고 조사해 본 후,  
본 강의를 듣도록 합시다.



# 01

## 리스트

- 1) 리스트의 정의
- 2) 리스트의 표현방식
- 3) 다중 리스트
- 4) 리스트 컴프리헨션 (List Comprehension)
- 5) 리스트 연습

## 1) 리스트의 정의

### 리스트

- 컬렉션의 하나로 연속되어 저장되어 묶음으로 관리하는 특별한 자료 저장 형태
- 기존 C,Java 프로그램 언어에서 배열(Array)와 비슷하나 조금 다르고, java의 ArrayList와 유사함

## 1) 리스트의 정의



리스트를 지정하면 첨자 0부터 ~ (n-1)까지의 변수로 지정

```
>>> a=[1,2,3,4,5]
>>> a[0],a[1],a[2],a[3],a[4]
(1, 2, 3, 4, 5)
```



리스트 내부에서는 서로 다른 형태의 자료형도 저장 가능함

```
>>> a=[11,"abc",12.3333,"안녕",2+3j]
>>> a[0],a[1],a[2],a[3],a[4]
(11, 'abc', 12.3333, '안녕', (2+3j))
```

## 1) 리스트의 정의



값의 (형태)변경도 가능

```
>>> a[1]=1234;a[4]="홍길동"
>>> a
[11, 1234, 12.3333, '안녕', '홍길동']
```



## 1) 리스트의 정의



for 반복문에서 많이 다루어 보았음

```
score=[90,80,70,60,30]
sum=0
for s in score:
    print("점수(",s,")")
    sum+=s
print("총점:",sum);
```



```
>>>
RESTART: ***.py
점수( 90 )
점수( 80 )
점수( 70 )
점수( 60 )
점수( 30 )
총점: 330
>>>
```

## 2) 리스트의 표현방식



일반적으로 리스트를 지정하면  
첨자 0부터 ~ (n-1)까지의 변수로 지정



하지만 역방향으로 -n ~ -1로 지정도 가능

-5 -4 -3 -2 -1

```
>>> a=[1,2,3,4,5]
>>> a[0],a[-5]
(1, 1)
>>> a[1],a[-4]
(2, 2)
>>> a[2],a[-3]
(3, 3)
>>> a[3],a[-2]
(4, 4)
>>> a[4],a[-1]
(5, 5)
```

## 2) 리스트의 표현방식



리스트의 첨자는 [begin:end:step]으로 표현 가능

[begin:end:step]

## 2) 리스트의 표현방식



첨자가 0부터 시작하고,  
end 값은 그 값보다 작을 때까지 임에 주의할 것

```
>>> a=[1,2,3,4,5,6,7,8,9,10]
>>> a[2:8] # 2부터 8보다 작을 때까지(2,3,4,5,6,7)
[3, 4, 5, 6, 7, 8]
>>> a[:4] #처음부터 4보다 작을 때까지(0,1,2,3 주의
a[4]제외)
[1, 2, 3, 4]
>>> a[5:] #5부터 끝까지
[6, 7, 8, 9, 10]
>>> a[0:6:3] #0부터 6보다 작을때 까지(6은 제외 0,3)
3씩 증가하여
[1, 4]
>>>
```

### 3) 다중 리스트



리스트의 요소로 리스트를 저장하여 2차, 3차 및 재귀적인 리스트 형태도 가능함



배열과 달리 가변 형식의 리스트도 가능함

### 3) 다중 리스트

```
>>> a=[[1,2,3],[4,5,6],[7,8,9]] # 리스트로 구성된 리스트
>>> a[2]
[7, 8, 9]
>>> a[2][2]
9
>>> a[1][3]
Traceback (most recent call last):
  File "<pyshell#49>", line 1, in <module>
    a[1][3]
IndexError: list index out of range
>>> a[1][2]
6
>>> a=[[1,2],[3,4,5],6] # 가변 형태의 리스트
>>> a
[[1, 2], [3, 4, 5], 6]
>>> a[2]
6
>>> a[1]
[3, 4, 5]
```

첨자(index)의  
범위를 벗어남

## 4) 리스트 컴프리헨션 (List Comprehension)

### 리스트 컴프리헨션 (List Comprehension)

리스트 중 조건을 비교하거나 범위를 주어 선택 사용이 가능한 방법

```
[요소 for 변수 in (리스트) (range) {if 조건}]
```

## 4) 리스트 컴프리헨션 (List Comprehension)

### 예제1

- ▶ 1부터 10으로 리스트를 구성하여 그 요소 하나하나에 3을 곱한 것을 구함

```
>>> a=[ n * 3 for n in range (1,10)]  
>>> a  
[3, 6, 9, 12, 15, 18, 21, 24, 27]
```



## 4) 리스트 컴프리헨션 (List Comprehension)

### 예제2

- ▶ 1부터 10중 2로 나누어 나머지가 0인(짝수인) 숫자로 리스트를 구성하여 그 요소 하나하나에 3을 곱한 것을 구함

```
>>> a=[ n * 3 for n in range (1,10) if n % 2 ==0 ]  
>>> a  
[6, 12, 18, 24]
```

## 5) 리스트 연습



앞장에서 연습한 내용을 리스트 구조를 변경된 사항으로 다시 작성

이름	국어	영어	수학	총점	평균
나연	100	90	100	290	96
정연	100	100	80	280	93
모모	90	100	100	290	96
사나	90	80	90	260	86
지효	100	100	100	300	100
미나	80	70	90	240	80
다현	70	60	100	230	76
채영	80	100	90	270	90
쯔위	100	100	100	300	100
반총점	810	800	850	273	91.0

## 5) 리스트 연습



앞장에서 연습한 내용을 리스트 구조를 변경된 사항으로 다시 작성



함수정의는 일단 pass로 정의하고 진행함

## 5) 리스트 연습

```
def titleprint():
    pass

def itemprint(name,kortotal,engtotal,mathtotal):
    pass

def tailprint():
    pass

#데이터 2차원 리스트
score=[["나연",100,90,100],
        ["정연",90,90,100],
        ["모모",80,70,90],
        ["사나",90,90,80],
        ["지효",100,80,80],
        ["미나",50,90,90],
        ["다현",80,60,100],
        ["채영",70,80,90],
        ["쯔위",100,90,90]]
#프로그램을 보기 좋게 하려고 상수를 정의
name=0;kor=1;eng=2;mat=3

#여기부터 프로그램
titleprint()
for one_student in score:
    itemprint(one_student[name],one_student[kor],one_student[eng],one_student[mat])
tailprint()
```

## 5) 리스트 연습



함수를 자세히 작성해 봄

```
def titleprint():
    pass

def itemprint(name, kdef titleprint():
    print("="*50)
    print("이름\t국어\t영어\t수학\t총점\t평균")
    print("="*50)

def itemprint(name, kor, eng, mat):
    print(name, "\t", kor, "\t", eng, "\t", mat, "\t", kor+eng+mat, "\t", (kor+eng+mat)//3)

def tailprint():
    totkor=0
    toteng=0
    totmat=0
    for one_student in score:
        totkor+=one_student[kor]
        toteng+=one_student[eng]
        totmat+=one_student[mat]
    print("="*50)
    print("반총점\t", totkor, "\t", toteng, "\t", totmat, "\t"\
        , (totkor+toteng+totmat)//9, "\t", (totkor+toteng+totmat)//3//9)
    print("="*50)
ortotal, engtotal, mathtotal):
    pass
```

-- 뒷장 계속 --

## 5) 리스트 연습



함수를 자세히 작성해 봄

```
def titleprint():
    pass

def itemprint(name, kor, eng, mat):
    print("="*50)
    print("이름\t국어\t영어\t수학\t총점\t평균")
    print("="*50)

def itemprint(name, kor, eng, mat):
    print(name, "\t", kor, "\t", eng, "\t", mat, "\t", kor+eng+mat, "\t", (kor+eng+mat)//3)

def tailprint():
    totkor=0
    toteng=0
    totmat=0
    for one_student in score:
        totkor+=one_student[kor]
        toteng+=one_student[eng]
        totmat+=one_student[mat]
    print("="*50)
    print("반총점\t", totkor, "\t", toteng, "\t", totmat, "\t"\
        , (totkor+toteng+totmat)//9, "\t", (totkor+toteng+totmat)//3//9)
    print("="*50)
    ortotal, engtotal, mathtotal = 0, 0, 0
    pass
```

## 5) 리스트 연습



함수를 자세히 작성해 봄

```
def tailprint():
    pass

#데이터 2차원 리스트
score=[["나연",100,90,100],
        ["정연",90,90,100],
        ["모모",80,70,90],
        ["사나",90,90,80],
        ["지효",100,80,80],
        ["미나",50,90,90],
        ["다현",80,60,100],
        ["채영",70,80,90],
        ["쯔위",100,90,90]]
#프로그램을 보기 좋게 하려고 상수를 정의
name=0;kor=1;eng=2;mat=3

#여기부터 프로그램
titleprint()
for one_student in score:
    itemprint(one_student[name],one_student[kor],one_student[eng],one_student[mat])
tailprint()
```

# 02

## 리스트 관리

- 1) 리스트 삽입, 삭제
- 2) 리스트 검색
- 3) 리스트 정렬
- 4) 심화학습



## 1) 리스트 삽입, 삭제



리스트 생성시 처음 값을 지정하거나, **삽입 (append)**,  
**해당 위치 삭제 (del)**, **전부삭제 (clear)**하여 값을 지정함



리스트값을 주고 생성한 후 변수를 지우는 del 명령어로  
삭제가 가능함



del a[2]로 2번 리스트요소를 삭제하면 뒤에 리스트들이  
앞으로 당겨짐



a.clear를 사용하면 모든 리스트가 삭제됨



빈 리스트는 남음

## 1) 리스트 삽입, 삭제

```
>>> a=[1,2,3,4,5]
>>> a
[1, 2, 3, 4, 5]
>>> del a[2] #2번 리스트요소 삭제
>>> a
[1, 2, 4, 5]
>>> del a[2]
>>> a
[1, 2, 5]
>>> a.clear() #리스트 전부 지움
>>> a #빈 리스트는 남아있음
[]
>>> del a #a공간도 지움
>>> a
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    a
NameError: name 'a' is not defined
```

## 1) 리스트 삽입, 삭제



a.append(값)으로 리스트를 추가하기 위하여  
최소 빈 리스트는 정의되어 있어야 함



리스트의 맨 뒤로부터 하나씩 요소가 추가되어 들어감

## 1) 리스트 삽입, 삭제

```
>>> del a
>>> a.append(1) #리스트가 정의되어 있지 않은 경우 추가는
불가
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    a.append(1)
NameError: name 'a' is not defined
>>> a=[] #빈 리스트를 정의
>>> a.append(1) #1값의 리스트요소를 추가
>>> a.append(3)
>>> a.append(5)
>>> a
[1, 3, 5]
>>> a=[4,5,6] #기존 a를 무시하고 다시 정의
>>> a.append(10)
>>> a
[4, 5, 6, 10]
>>>
```

## 1) 리스트 삽입, 삭제



한 개의 요소를 삽입시 insert를 사용하고  
**list.insert(삽입위치,삽입내용)**으로 사용

```
>>> a=[1,2,3,4,5]
>>> a.append(6) #append는 맨 뒤에 하나의 요소를 삽입
>>> a
[1, 2, 3, 4, 5, 6]
>>> a.insert(3,12)
>>> a
[1, 2, 3, 12, 4, 5, 6]
>>> a.insert(4,15)
>>> a
[1, 2, 3, 12, 15, 4, 5, 6]
>>> a.insert(3,1)
>>> a
[1, 2, 3, 1, 12, 15, 4, 5, 6]
```

## 1) 리스트 삽입, 삭제



여러 개의 요소를 삽입 시 `list[n:n]` = 방식으로 사용  
(n은 삽입위치)

```
>>> a=[1, 2, 3, 1, 12, 15, 4, 5, 6]
>>> a[5:5]=[1,2,3,4,5,6] #5에서 5까지 영역에 해당 리스트 값을 대입
>>> a
[1, 2, 3, 1, 12, 1, 2, 3, 4, 5, 6, 15, 4, 5, 6]
```

## 1) 리스트 삽입, 삭제



list[n]=을 사용하면 해당n요소의 리스트가 대입된  
경우임에 주의할 것

```
>>> a=[1, 2, 3, 1, 12, 15, 4, 5, 6]
>>> a[5]=[1,2,3,4,5,6] #5요소에 이 리스트를 대입(큰 차이)
>>> a
[1, 2, 3, 1, 12, [1, 2, 3, 4, 5, 6], 4, 5, 6]
```

## 1) 리스트 삽입, 삭제

두 개의 리스트를  
병합하는 방법

- `list.extend`를 사용
- `+` 연산을 이용

```
>>> a=[1,2,3]
>>> b=[4,5,6]
>>> a=a+b
>>> a
[1, 2, 3, 4, 5, 6]
>>> a=[1,2,3]
>>> b=[4,5,6]
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6]
```



## 1) 리스트 삽입, 삭제



리스트를 지우기 위하여 `list.remove(원하는값)`,  
`del(list[n:m])`, `list.clear()`를 사용

```
>>> a=[1,2,3,3,4,4,5,6,7,7,7,7,8]
>>> a
[1, 2, 3, 3, 4, 4, 5, 6, 7, 7, 7, 7, 8]
>>> a.remove(7) #첫 번째 만나는 7값을 가진 리스트 삭제
>>> a
[1, 2, 3, 3, 4, 4, 5, 6, 7, 7, 7, 8]
>>> a.remove(7) #첫 번째 만나는 7값을 가진 리스트 삭제
>>> a
[1, 2, 3, 3, 4, 4, 5, 6, 7, 7, 8]
>>> a[5]=[] #다섯 번째에 빈 값[]이 들어 있는 리스트를
만듬(주의: a[5:5]=[]라고 해야 함)
>>> a
[1, 2, 3, 3, 4, [], 5, 6, 7, 7, 8]
>>> a[2:4]=[]
```

-- 뒷장 계속 --

## 1) 리스트 삽입, 삭제



리스트를 지우기 위하여 **list.remove(원하는값)**,  
**del(list[n:m])**, **list.clear()**를 사용

```
>>> a
[1, 2, 4, [], 5, 6, 7, 7, 8]
>>> del(a[4])
>>> a
[1, 2, 4, [], 6, 7, 7, 8]
>>> del(a[3])
>>> a
[1, 2, 4, 6, 7, 7, 8]
>>> a.clear() #다 지우나 빈 값에 리스트가 남음
>>> a
[]
>>> del(a) #리스트 정의까지 삭제
>>> a
Traceback (most recent call last):
  File "<pysHELL#48>", line 1, in <module>
    a
NameError: name 'a' is not defined
>>>
```

## 1) 리스트 삽입, 삭제



**list.pop(n)**은 해당위치를 리턴하고 삭제함 n이 없다면 맨 마지막 요소 리턴

```
>>> a=[1,2,3,3,4,4,5,6,7,7,7,7,8]
>>> a.pop()
8
>>> a
[1, 2, 3, 3, 4, 4, 5, 6, 7, 7, 7, 7]
>>> a.pop()
7
>>> a.pop(3)
3
>>> a
[1, 2, 3, 4, 4, 5, 6, 7, 7, 7]
>>> a.pop(6)
6
>>> a
[1, 2, 3, 4, 4, 5, 7, 7, 7]
>>> a.pop(6)
7
>>> a.pop(0)
1
>>> a
[2, 3, 4, 4, 5, 7, 7]
```

## 2) 리스트 검색

구분	설명
<code>list.index(값)</code>	해당 값이 처음 나오는 위치를 리턴
<code>list.count(값)</code>	해당 값이 몇 번 나오는지 리턴

```
>>> a=[1,2,3,3,4,4,5,6,7,7,7,7,8]
>>> a.index(7)
8
>>> a.index(8) #값 8의 위치는 12(0부터 시작)
12
>>> a.index(4) #값 4의 위치는 4(0부터 시작)
4
>>> a.count(7) #7은 4번 나옴
4
```

## 2) 리스트 검색

구분	설명
<code>min(list),max(list)</code>	리스트중 최소값과 최대값을 구함
<code>len(list)</code>	리스트의 크기(개수)를 구함

```
>>> a=[1,2,3,3,4,4,5,6,7,7,7,7,8]
>>> min(a);max(a);len(a)
1
8
13
```

### 3) 리스트 정렬

구분	설명
<code>list.sort()</code>	리스트 내용을 정렬함
<code>list.reverse()</code>	리스트 내용을 역으로 보여줌

```
>>> a=[1,8,5,6,2,4,5,2,6,9,4]
>>> a.reverse()
>>> a
[4, 9, 6, 2, 5, 4, 2, 6, 5, 8, 1]
>>> a.sort()
>>> a
[1, 2, 2, 4, 4, 5, 5, 6, 6, 8, 9]
```

## 4) 심화학습

구분	설명
import 문	외부의 함수를 가지고 올 때 사용
import random	라이브러리를 가져다 사용하겠다는 정의
import random as r	프로그램 안에서 r.random() r.randrange(1,10)과 같이 r 다음에 오는 함수는 random라이브러리로부터 가지고 온 것임을 정의

## 4) 심화학습

다음은 앞에서 실습한 트와이스 성적표에서 score리스트를 임의로 생성된 값으로 테스트하는 내용

```
import random as r
def titleprint():
    pass

def itemprint(name,kor,eng,mat):
    pass

def tailprint():
    pass

one_person=[]
score=[]
for i in range(100):
    one_person.append("name"+str(i)) #이름
    one_person.append(r.randrange(10,100)) #국어 10부터 100까지 임의에 값을 추출
    one_person.append(r.randrange(10,100)) #영어
    one_person.append(r.randrange(10,100)) #수학
    score.append(one_person[:]) #값을 복사해서 추가함에 주의할 것
    one_person.clear()

#프로그램을 보기 좋게 하려고 상수를 정의
name=0;kor=1;eng=2;mat=3

#여기부터 프로그램
titleprint()
for one_student in score:
```



# 03

## 튜플

- 1) 튜플
- 2) 튜플의 예시

## 1) 튜플

### 튜플

- 튜플은 값을 정의시 [] 대신 ()로 대입
- 튜플은 값을 변경할 수 없는 자료형

## 1) 튜플



값을 조회는 가능

```
>>> a=(1,8,5,6,2,4,5,2,6,9,4)
>>> a
(1, 8, 5, 6, 2, 4, 5, 2, 6, 9, 4)
>>> a[5]
4
>>> a[3:5]
(6, 2)
>>> a[3:8]
(6, 2, 4, 5, 2)
>>> a*2
(1, 8, 5, 6, 2, 4, 5, 2, 6, 9, 4, 1, 8, 5, 6,
2, 4, 5, 2, 6, 9, 4)
```

## 1) 튜플



값을 대입하거나 변경하거나 구조를 지우는 것은 불가능


```
>>> a
(1, 8, 5, 6, 2, 4, 5, 2, 6, 9, 4, 1, 8, 5, 6, 2, 4, 5, 2, 6, 9, 4)
>>> a[1]=5 #값 변경불가
Traceback (most recent call last):
  File "<pyshell#86>", line 1, in <module>
    a[1]=5
TypeError: 'tuple' object does not support item assignment
>>> del a[4] #변수 지우기 불가
Traceback (most recent call last):
  File "<pyshell#87>", line 1, in <module>
    del a[4]
TypeError: 'tuple' object doesn't support item deletion
>>> >>> a.append(10)
Traceback (most recent call last):
  File "<pyshell#88>", line 1, in <module>
    a.append(10)
AttributeError: 'tuple' object has no attribute 'append'
```

## 2) 튜플의 예시



함수의 리턴 값을 여러 개 정의할 경우 튜플로 전달됨

```
def location():  
    latitude = 2, 17403 # 위도  
    longitude=41,40338 #경도  
    return latitude,longitude #이런방식의 리턴은 튜플로 반환  
  
a=location()  
print(a)
```



```
>>>  
== RESTART: ** .py ==  
((2, 17403), (41, 40338))  
>>>
```

# 04

## 실습하기 I

- 1) 리스트 - 정의,표현방식
- 2) 리스트 - 다중 리스트, 리스트 컴프리헨션
- 3) 리스트 연습

# 실습내용

- 1) 리스트 - 정의,표현방식
- 2) 리스트 - 다중 리스트, 리스트 컴프리헨션
- 3) 리스트 연습

# 05

## 실습하기 II

- 1) 리스트 관리 - 리스트 삽입, 삭제
- 2) 리스트 관리 - 검색, 정렬, 심화학습
- 3) 튜플 - 정의, 예제



# 실습내용

- 1) 리스트 관리 - 리스트 삽입, 삭제
- 2) 리스트 관리 - 검색, 정렬, 심화학습
- 3) 튜플 - 정의, 예제



## 학습활동

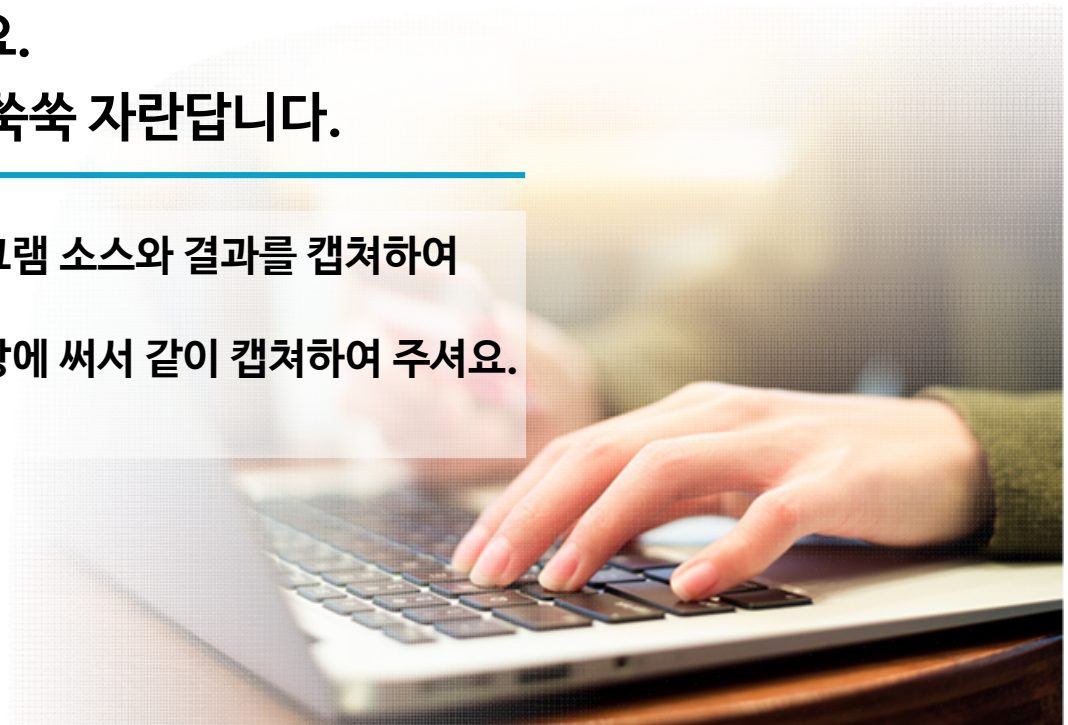
일시정지 버튼을 누른 후, 아래의 학습활동에 참여하세요.

Q

오늘 배운 내용을 스스로 실습하여  
자유게시판에 올려 주세요.

이렇게 정리하면 실력이 쑥쑥 자란답니다.

- ① 본인이 실습한 내용을 프로그램 소스와 결과를 캡처하여 올려주세요.
- ② 본인의 학번과 이름을 메모장에 써서 같이 캡처하여 주세요.
- ③ 그리고 설명도 달아 주세요.





## 학습활동에 대한 교수님 의견

Q

오늘 배운 내용을 스스로 실습하여 자유게시판에 올려 주세요.  
이렇게 정리하면 실력이 쑥쑥 자란답니다.

A

[ 오늘 학습한 내용의 실습 사항 ]

- ① 리스트 - 정의, 표현방식
- ② 리스트 - 다중 리스트, 리스트 컴프리헨션
- ③ 리스트 연습
- ④ 리스트 관리 - 리스트 삽입, 삭제
- ⑤ 리스트 관리 - 검색, 정렬, 심화학습
- ⑥ 튜플 - 정의, 예제

# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

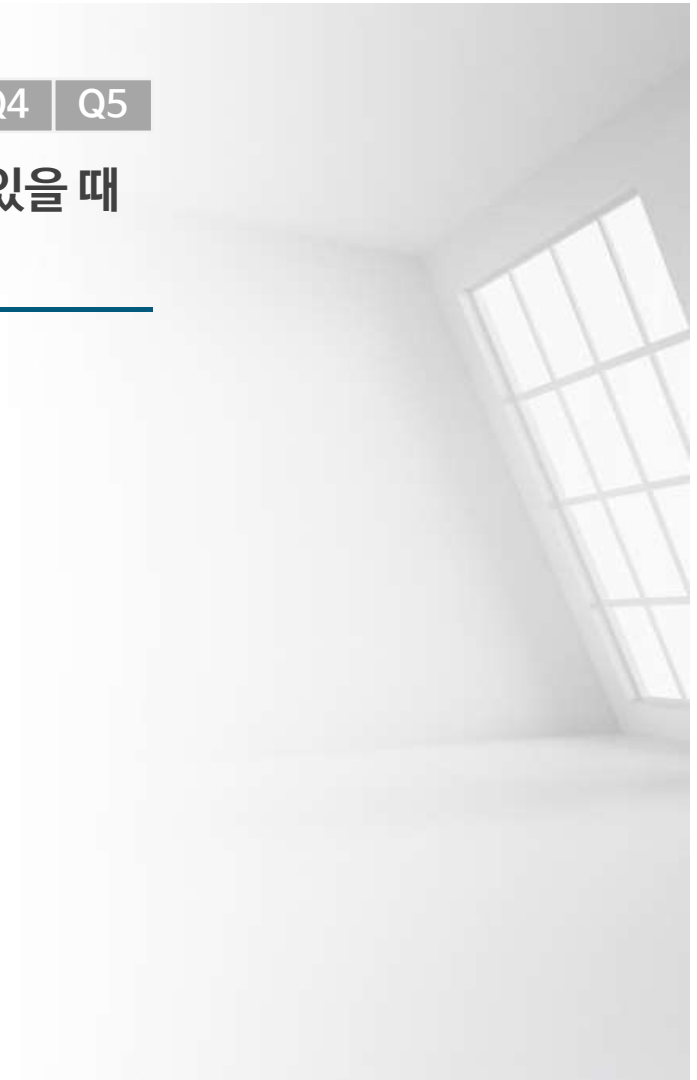
다음 중  $a=[1,2,3,4,5]$ 로 정의되어 있을 때  $a[-4]$ 의 값은?

1 1

2 2

3 error

4 4



# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q1

다음 중  $a=[1,2,3,4,5]$ 로 정의되어 있을 때  $a[-4]$ 의 값은?

1 1

☒ 2

3 error

4 4

정답

2번

해설

참자가 음수인 경우 역방향으로 지정됩니다.

# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음  $a = [n * 3 \text{ for } n \text{ in range } (1, 10) \text{ if } n \% 2 == 0]$ 로 정의되어 있을 때 a리스트의 값이 될 수 **없는** 것은?

1 6

2 12

3 14

4 18



# 학습 평가

Q1

Q2

Q3

Q4

Q5

Q2

다음  $a = [n * 3 \text{ for } n \text{ in range } (1, 10) \text{ if } n \% 2 == 0]$ 로 정의되어 있을 때 a리스트의 값이 될 수 **없는** 것은?

1 6

2 12

☒ 3 14

4 18

정답

3번

해설

a는 1부터 10보다 작은 짝수 [2,4,6,8] 값에 3을 곱한 값으로 [6,12,18,24] 입니다.

# 학습 평가

Q1

Q2

Q3

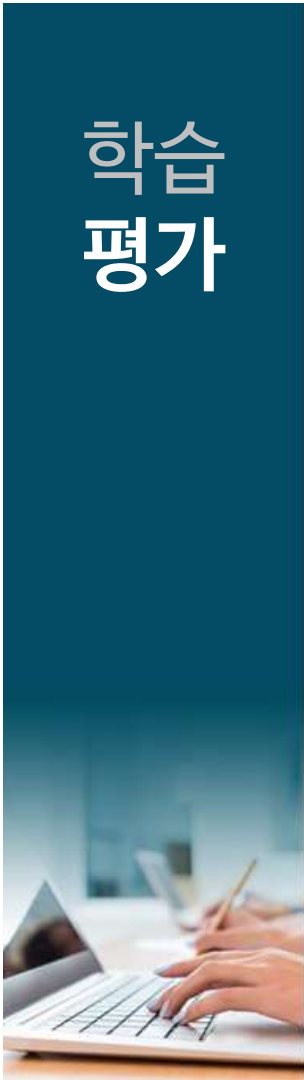
Q4

Q5

## Q3

다음 `a = [1, 2, 3]`으로 정의되어 있을 때 맨 뒤에 값 3을 추가하기 위하여 바르게 사용된 것은?

- 1 `a.insert(3)`
- 2 `a.copy(3)`
- 3 `a.append(2)`
- 4 `a[3] = 3`





# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q3

다음 `a = [1, 2, 3]`으로 정의되어 있을 때 맨 뒤에 값 3을 추가하기 위하여 바르게 사용된 것은?

1 `a.insert(3)`

2 `a.copy(3)`

☒ 3 `a.append(2)`

4 `a[3] = 3`

정답

3번

해설

`append`로 맨 마지막에 추가를 합니다.

# 학습 평가

Q1

Q2

Q3

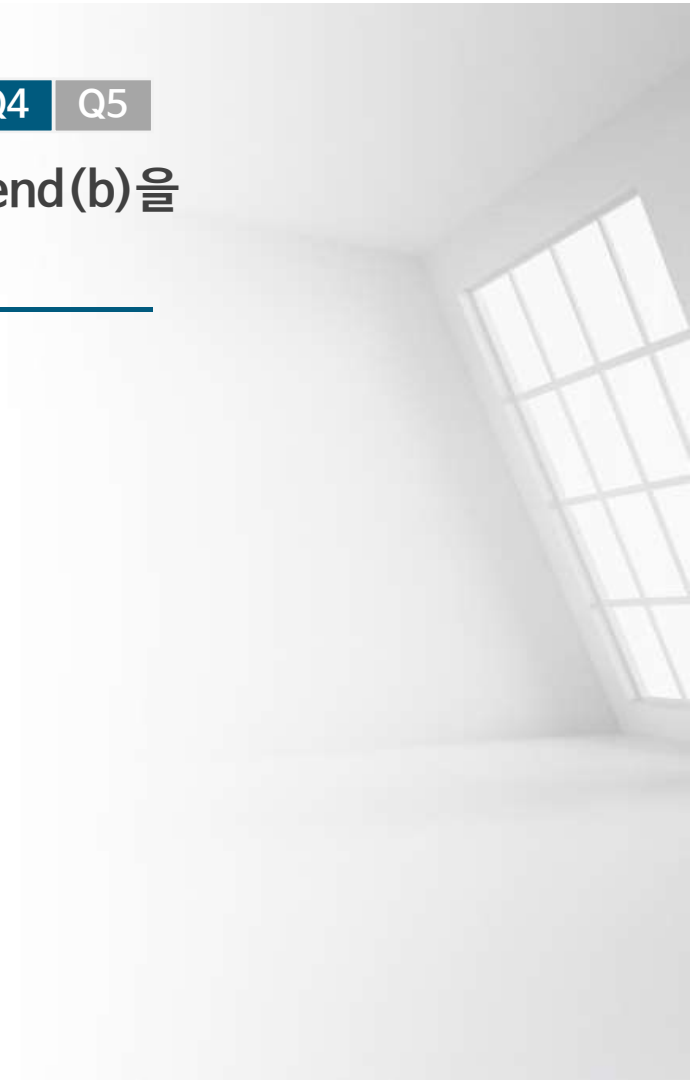
Q4

Q5

## Q4

다음 `a=[1,2,3]`; `b=[4,5,6]`; `a.extend(b)`을 실행하였을 때 리스트 `b`의 값은 ?

- 1 `[1,2,3]`
- 2 `[1,2,3,4,5,6]`
- 3 `[4,5,6]`
- 4 `[]`



# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q4

다음 `a=[1,2,3]`; `b=[4,5,6]`; `a.extend(b)`을 실행하였을 때 리스트 `b`의 값은 ?

1 `[1,2,3]`

2 `[1,2,3,4,5,6]`

☒ 3 `[4,5,6]`

4 `[]`

정답

3번

해설

`a.extend(b)`는 `a`리스트 끝에 `b` 리스트를 추가하여 `a`리스트가 커진 것으로 `b` 리스트의 값은 변화가 없다.

# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q5

다음  $a=(1,2,3)$ 으로 정의 되었을 때  
**잘못** 사용된 것은?

- 1 `print(a)`
- 2 `print(a[4])`
- 3 `print(a*3)`
- 4 `a[2]=1`



# 학습 평가

Q1

Q2

Q3

Q4

Q5

## Q5

다음  $a=(1,2,3)$ 으로 정의 되었을 때  
**잘못** 사용된 것은?

1 `print(a)`

2 `print(a[4])`

3 `print(a*3)`

☒ `a[2]=1`

정답

4번

해설

튜플은 값을 변경할 수 없는 자료형이다.



# 정리하기

## 리스트

- ✓ 컬렉션의 하나로 연속되어 저장되어 묶음으로 관리하는 특별한 자료 저장 형태
- ✓ 리스트의 첨자는 **[begin:end:step]**으로 표현 가능하며 음수로 표현도 가능
- ✓ **리스트 컴프리헨션은 리스트 중 조건을 비교하거나 범위를 주어 선택 사용이 가능한 방법**





## 정리하기

### 리스트 관리

- ✓ 리스트 생성시 처음 값을 지정하거나, 삽입(append), 해당 위치 삭제(del), 전부삭제(clear)하여 값을 지정
- ✓ 한 개의 요소를 삽입시 insert를 사용하고 list.insert(삽입위치, 삽입내용)으로 사용
- ✓ 여러 개의 요소를 삽입 시 list[n:n] = 방식으로 사용 (n은 삽입위치)
- ✓ 두 개의 리스트를 병합하는 방법은 list.extend를 사용하거나 + 연산을 이용
- ✓ 리스트를 지우기 위하여 list.remove(원하는값), del(list[n:m]), list.clear()를 사용
- ✓ min(list), max(list)은 리스트 중 최소값과 최대값을 구하며 len(list)은 리스트의 크기(개수)를 구함





## 정리하기

### 튜플

- ✓ 튜플은 값을 변경할 수 없는 자료형

