



9장. 리스트와 튜플

파이썬 정복



Contents

❖ 목차

- 1. 리스트
- 2. 리스트 관리
- 3. 튜플



1. 리스트

❖ 자료의 집합

- 리스트는 여러 개 값을 집합적으로 저장
- 요소 (Element)
 - 리스트에 소속되는 각각의 값
 - 리스트에는 주로 같은 타입 요소를 모음

```
score = [ 88, 95, 70, 100, 99 ]  
name = [ "최상미", "이한승", "김기남" ]
```

listscore

```
score = [ 88, 95, 70, 100, 99 ]  
sum = 0  
for s in score:  
    sum += s  
print("총점 : ", sum)  
print("평균 : ", sum / len(score))
```

실행결과

```
총점 : 452  
평균 : 90.4
```



1. 리스트

❖ 리스트의 요소

- 개별요소 읽기 : 대괄호 안에 읽고자 하는 요소의 순서값 적음

```
score = [ 88, 95, 70, 100, 99 ]  
print(score[0])      # 88  
print(score[2])      # 70  
print(score[-1])     # 99
```

- 요소 분리 : 범위 지정

- [begin:end:step]

listslice

```
nums = [0,1,2,3,4,5,6,7,8,9]  
print(nums[2:5])      # 2~5까지  
print(nums[:4])       # 처음부터 4까지  
print(nums[6:])       # 6에서 끝까지  
print(nums[1:7:2])    # 1~7까지 하나씩 건너뛰며
```

실행결과

```
[2, 3, 4]  
[0, 1, 2, 3]  
[6, 7, 8, 9]  
[1, 3, 5]
```



1. 리스트

- 대괄호에 첨자 지정할 수 있음

listassign

```
score = [ 88, 95, 70, 100, 99 ]  
print(score[2])    # 70  
score[2] = 55      # 값 변경  
print(score[2])    # 55
```

실행결과

70
55

listreplace

```
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
nums[2:5] = [20, 30, 40]  
print(nums)  
nums[6:8] = [90, 91, 92, 93, 94]  
print(nums)
```

실행결과

[0, 1, 20, 30, 40, 5, 6, 7, 8, 9]
[0, 1, 20, 30, 40, 5, 90, 91, 92, 93, 94, 8, 9]



1. 리스트

❖ 이중 리스트

- 리스트의 요소로 리스트 넣어 중첩할 수 있음

nestlist

```
lol = [ [1, 2, 3], [4, 5], [6, 7, 8, 9]]  
print(lol[0])  
print(lol[2][1])
```

```
for sub in lol:  
    for item in sub:  
        print(item, end=' ')  
    print()
```

실행결과

```
[1, 2, 3]  
7  
1 2 3  
4 5  
6 7 8 9
```

- 이중 리스트 순회하여 최종값 읽으려면 루프도 이중으로 하여야 함



1. 리스트

❖ 리스트 컴프리헨션 (List Comprehension)

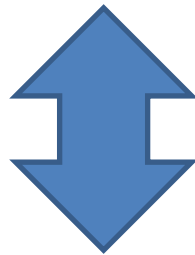
- [수식 for 변수 in 리스트 if 조건]

listcomp

```
nums = [n * 2 for n in range(1, 11)]  
for i in nums:  
    print(i, end = ', ')
```

실행결과

2, 4, 6, 8, 10, 12, 14, 16, 18, 20,



```
nums = []  
for n in range(1, 11):  
    nums.append(n * 2)
```



2. 리스트 관리

❖ 삽입

- **append** : 인수로 전달한 요소를 리스트 끝에 추가
- **insert** : 삽입할 위치와 요소값을 전달받아 리스트 중간에 삽입

listinsert

```
nums = [1, 2, 3, 4]
nums.append(5)
nums.insert(2, 99)
print(nums)
```

실행결과

[1, 2, 99, 3, 4, 5]



2. 리스트 관리

- 범위에 리스트 대입하여 여러 요소 한꺼번에 삽입 가능

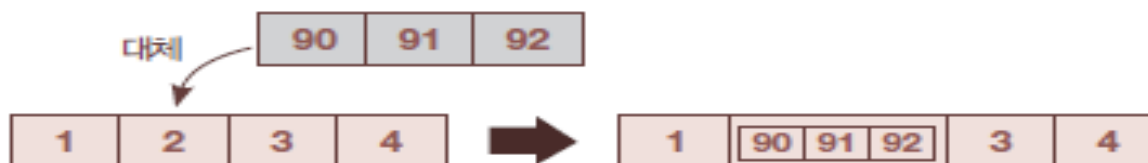
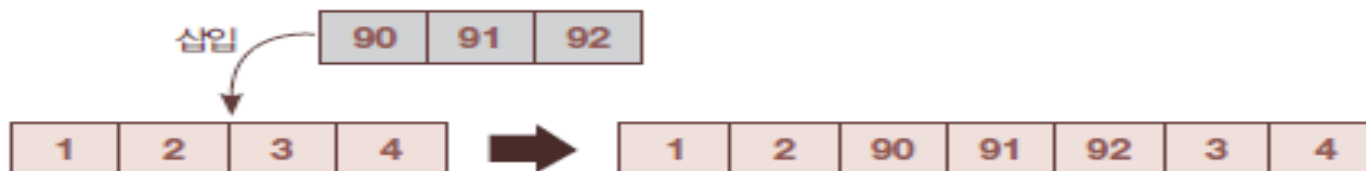
insertrange

```
nums = [1, 2, 3, 4]  
nums[2:2] = [90, 91, 92]  
print(nums)
```

```
nums = [1, 2, 3, 4]  
nums[2] = [90, 91, 92]  
print(nums)
```

실행결과

```
[1, 2, 90, 91, 92, 3, 4]  
[1, 2, [90, 91, 92], 4]
```



2. 리스트 관리

❖ 삭제

- 대상 선택 방법에 따라 다른 메서드 사용

listremove

```
score = [ 88, 95, 70, 100, 99, 80, 78, 50 ]  
score.remove(100)  
print(score)  
del(score[2])  
print(score)  
score[1:4] = []  
print(score)
```

실행결과

```
[88, 95, 70, 99, 80, 78, 50]  
[88, 95, 99, 80, 78, 50]  
[88, 78, 50]
```

- **remove** : 인수로 전달받은 요소값 찾아 삭제
- **del** : 순서값 지정하여 삭제
- **clear** : 리스트 모든 요소 삭제
- **빈 리스트 대입** : 일정 범위 요소 다수 삭제



2. 리스트 관리

❖ 검색

- **index** : 특정 요소 위치 찾기
- **count** : 특정 요소값의 개수 조사

listindex

```
score = [ 88, 95, 70, 100, 99, 80, 78, 50 ]  
perfect = score.index(100)  
print("만점 받은 학생은 " + str(perfect) + "번입니다.")  
pernum = score.count(100)  
print("만점자 수는 " + str(pernum) + "명입니다")
```

실행결과

만점 받은 학생은 3번입니다.
만점자 수는 1명입니다

- **min / max** : 리스트 요소 중 최소값 / 최대값 찾기
- **in / not in** : 특정 요소 유무 여부 검사



2. 리스트 관리

❖ 정렬

- 요소를 크기순으로 재배열
- **sort** : 리스트 정렬하며 요소 순서 조정. 리스트 자체 수정
- **reverse** : 요소 순서 반대로

sort

```
score = [ 88, 95, 70, 100, 99 ]  
score.sort()  
print(score)  
score.reverse()  
print(score)
```

실행결과

```
[70, 88, 95, 99, 100]  
[100, 99, 95, 88, 70]
```

- **key** : 정렬 시 요소 비교할 키 추출
- **sorted** : 리스트 그대로 두고 정렬된 새로운 리스트 만들어 리턴



3. 튜플

❖ 불변 자료 집합

- 튜플은 초기화한 후 편집할 수 없다는 점에서 리스트와 차이
- 소괄호 사용하여 정의

tuplescore

```
score = ( 88, 95, 70, 100, 99 )
sum = 0
for s in score:
    sum += s
print("총점 : ", sum)
print("평균 : ", sum / len(score))
```

실행결과

총점 : 452
평균 : 90.4

- print : 튜플 출력 시 소괄호 함께 출력하여 리스트 아님을 나타냄
- 정의할 때에는 소괄호 없이 값만 나열해도 무관함
 - 요소 하나밖에 없는 경우에는 값 다음에 콤마 찍어 튜플임을 표시



3. 튜플

❖ 튜플로 가능한 일

- + / * 연산자 사용하여 연결 및 반복

tupleop

```
tu = 1, 2, 3, 4, 5
print(tu[3])      # 가능
print(tu[1:4])    # 가능
print(tu + (6, 7)) # 가능
print(tu * 2)     # 가능
tu[1] = 100       # 불가능
del tu[1]         # 불가능
```

실행결과

```
4
(2, 3, 4)
(1, 2, 3, 4, 5, 6, 7)
(1, 2, 3, 4, 5, 1, 2, 3, 4, 5)
Traceback (most recent call last):
  File "C:/Python/test.py", line 6, in <module>
    tuple[1] = 100      # 불가능
TypeError: 'tuple' object does not support item assignment
```

- 요소를 변경하거나 삭제할 수는 없음



3. 튜플

- 여러 개의 변수에 값을 한꺼번에 대입
 - 좌변에 변수 목록, 우변에 튜플을 대입

unpacking

```
tu = "이순신", "김유신", "강감찬"  
lee, kim, kang = tu  
print(lee)  
print(kim)  
print(kang)
```

실행결과

이순신
김유신
강감찬



3. 튜플

- 두 개 이상 값을 반환
 - 내부에 요소 포함하는 튜플 사용

tworeturn

```
import time

def gettime():
    now = time.localtime()
    return now.tm_hour, now.tm_min

result = gettime()
print("지금은 %d시 %d분입니다" % (result[0], result[1]))
```

실행결과

지금은 5시 26분입니다

- **import** : 모듈 기능 사용 명령
- **divmod** : 나눗셈의 몫과 나머지를 튜플로 묶어 리턴





Thank You !

파이썬 정복

