

Unit 22.

Status	완료
담당자	
마감일	
완료일	@2022년 11월 23일

Unit 22. 리스트와 튜플 응용하기

- 리스트에는 요소를 추가할 수 있는데 요소를 추가할 수 있게 하는 메서드들이 있다
 - `append`: 요소 하나를 추가
 - `extend`: 리스트를 연결하여 확장
 - `insert`: 특정 인덱스에 요소 추가
- 가장 대표적인 3가지

22.1.1 리스트에 요소 하나 추가하기

- `append`는 리스트 끝에 요소 하나를 추가한다
- 빈 리스트에도 가능

```
>>> a = [10, 20, 30]
>>> a.append(500)
>>> a
[10, 20, 30, 500]
>>> len(a)
4
```

22.1.2 리스트 안에 리스트 추가하기

- `append`는 **`append(리스트)`**처럼 리스트를 넣으면 리스트 안에 리스트가 들어가 중첩 리스트가 된다

```
>>> a = [10, 20, 30]
>>> a.append([500, 600])
>>> a
[10, 20, 30, [500, 600]]
>>> len(a)
4
```

22.1.4 리스트 확장하기

- 리스트에 요소를 여러개 추가하고 싶다면 `extend`를 사용한다
- 리스트 끝에 리스트를 연결한다

```
>>> a = [10, 20, 30]
>>> a.extend([500, 600])
>>> a
[10, 20, 30, 500, 600]
>>> len(a)
5
```

22.1.5 리스트의 특정 인덱스에 요소 추가하기

- 원하는 위치에 요소를 추가하고 싶을때는 `insert`를 사용
- **`insert(인덱스, 요소)`**는 리스트의 특정 인덱스에 요소 하나 추가

```
>>> a = [10, 20, 30]
>>> a.insert(2, 500)
>>> a
[10, 20, 500, 30]
>>> len(a)
4
```

- **`insert(0, 요소)`**: 리스트의 맨 처음에 요소를 추가

- `insert(len(리스트), 요소)`: 리스트 끝에 요소를 추가

22.1.6 리스트에서 요소 삭제하기

- `pop`: 마지막 요소 또는 특정 인덱스의 요소를 삭제
- `remove`: 특정 값을 찾아서 삭제

22.1.7 리스트에서 특정 인덱스의 요소를 삭제

- `pop()` 은 리스트의 마지막 요소를 삭제한 뒤 삭제한 요소를 반환

```
>>> a = [10, 20, 30]
>>> a.pop()
30
>>> a
[10, 20]
```

- 원하는 인덱스의 요소를 삭제할때는 `pop`에 인덱스를 지정하면 된다

```
>>> a = [10, 20, 30]
>>> a.pop(1)
20
>>> a
[10, 30]
```

`pop` 대신 `del` 써도 상관없음

22.1.8 리스트에서 특정 값을 찾아서 삭제하기

- 리스트에서 원하는 값을 찾아서 삭제하고 싶을때 `remove`사용

```
>>> a = [10, 20, 30]
>>> a.remove(20)
>>> a
[10, 30]
```

22.1.10 특정 값의 인덱스, 개수 구하기

- `index(값)` 은 리스트에서 특정 값의 인덱스를 구한다

```
>>> a = [10, 20, 30, 15, 20, 40]
>>> a.index(20)
1
```

- `count(값)` 은 리스트에서 특정 값의 개수를 구한다

```
>>> a = [10, 20, 30, 15, 20, 40]
>>> a.count(20)
2
```

22.1.11 리스트의 순서 뒤집기

- `revers`는 리스트 요소 순서를 뒤집는다

```
>>> a = [10, 20, 30, 15, 20, 40]
>>> a.reverse()
>>> a
[40, 20, 15, 30, 20, 10]
```

22.1.12 리스트의 요소를 정렬하기

- `sort()` 는 리스트의 요소를 작은 순서대로 정렬하기(오름차순)
- `sort()` 또는 `sort(reverse=False)`: 리스트의 값을 작은 순서대로 정렬(오름차순)
- `sort(reverse=True)`: 리스트의 값을 큰 순서대로 정렬(내림차순)

```
>>> a = [10, 20, 30, 15, 20, 40]
>>> a.sort()
```

```
>>> a
[10, 15, 20, 20, 30, 40]
```

22.1.3 리스트의 모든 요소를 삭제하기

`clear()`는 리스트의 모든 요소를 삭제

22.2 리스트의 할당과 복사

- 할당과 복사는 비슷하지만 큰 차이점이 있다

```
>>> a = [0, 0, 0, 0, 0]
>>> b = a
```

이런식으로 a에 할당해주면 리스트가 2개가 될까 같지만 실제로는 리스트가 하나다

a와 b를 2개로 만드려면 copy메서드로 모든 요소를 복사해야 한다

- 그뒤에 a와 b를 비교해보면 두 리스트는 다른 객체

22.2.3 반복문으로 리스트의 요소를 모두 출력하기

```
for 변수 in 리스트:
    반복할 코드
```

22.2.3 인덱스와 요소를 같이 출력하기

- `enumerate`를 사용한다
- **for 인덱스, 요소 in enumerate(리스트):**

```
>>> a = [38, 21, 53, 62, 19]
>>> for index, value in enumerate(a):
...     print(index, value)
...
0 38
1 21
2 53
3 62
4 19
```

- 인덱스를 원하는 부분에서 출력하고 싶을 때는

- **for 인덱스, 요소 in enumerate(리스트, start=숫자):**

```
>>> for index, value in enumerate(a, start=1):
...     print(index, value)
...
1 38
2 21
3 53
4 62
5 19
```

22.4.1 가장 작은 수와 가장 큰 수 구하기

- `sort` 메서드를 이용해서

```
>>> a = [38, 21, 53, 62, 19]
>>> a.sort()
>>> a[0]
19
>>> a.sort(reverse=True)
>>> a[0]
62
```

- `min`, `max` 함수를 이용

```
>>> a = [38, 21, 53, 62, 19]
>>> min(a)
19
>>> max(a)
62
```

22.4.2 요소의 합계 구하기

- `sum` 이용하기

```
>>> a = [10, 10, 10, 10, 10]
>>> sum(a)
50
```

22.5. 리스트 표현식 사용하기

- 파이썬의 리스트는 안에 for 반복문과 if 조건문을 사용 가능
 - 이렇게 리스트 안에 for 반복문 if 조건문 등을 지정하여 리스트를 생성하는게 리스트 컴프리헨션 이다
 - [식 for 변수 in 리스트]
 - list(식 for 변수 in 리스트)
- 리스트 컴프리헨션 식

```
>>> a = [i for i in range(10)]      # 0부터 9까지 숫자를 생성하여 리스트 생성
>>> a
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> b = list(i for i in range(10))  # 0부터 9까지 숫자를 생성하여 리스트 생성
>>> b
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

22.5.1 리스트 표현식에서 if 조건문 사용하기

- [식 for 변수 in 리스트 if 조건식]
- list(식 for 변수 in 리스트 if 조건식)

```
>>> a = [i for i in range(10) if i % 2 == 0]    # 0~9 숫자 중 2의 배수인 숫자(짝수)로 리스트 생성
>>> a
[0, 2, 4, 6, 8]
```

22.6 리스트에 map 사용하기

- map은 리스트의 요소를 지정된 함수로 처리해주는 함수
- list(map(함수, 리스트))
- tuple(map(함수, 튜플))

```
>>> a = [1.2, 2.5, 3.7, 4.6]
>>> a = list(map(int, a))
>>> a
[1, 2, 3, 4]
```

map에 int와 리스트를 넣으면 리스트의 모든 요소를 int로 변환

- map은 리스트뿐 아니라 모든 반복 가능한 객체를 넣을 수 있다

```
>>> a = list(map(str, range(10)))
>>> a
['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

22.7 튜플 응용하기

- 튜플은 내용을 변경할 수 없기 때문에 요소를 변경하는 메서드는 사용 불가능
- index, count같이 요소의 내용을 알수있는 메서드만 사용이 가능하다
- map, min, max과 for 반복문으로 요소 출력하는 방법은 리스트와 동일하다