리스트 관련 함수들

리스트에 요소 추가(append)

```
>>> a = [1, 2, 3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
```

append를 사전에서 검색해 보면 "덧붙이다, 첨부하다"라는 뜻이 있다. 이 뜻을 안다면 아래의 예가 금방 이해가 될 것이다. append(x)는 리스트의 맨 마지막에 x를 추가시키는 함수이다.

리스트 안에는 어떤 자료형도 추가할 수 있다.
아래의 예는 리스트에 다시 리스트를 추가한 결과이다.

```
>>> a.append([5,6])
>>> a
[1, 2, 3, 4, [5, 6]]
```

리스트 정렬(sort)

```
>>> a = [1, 4, 3, 2]
>>> a.sort()
>>> a
[1, 2, 3, 4]
```

sort 함수는 리스트의 요소를 순서대로 정렬해 준다.
문자 역시 알파벳 순서로 정렬할 수 있다.

```
>>> a = ['a', 'c', 'b']
>>> a.sort()
>>> a
['a', 'b', 'c']
```

리스트 뒤집기(reverse)

```
>>> a = ['a', 'c', 'b']
>>> a.reverse()
>>> a
['b', 'c', 'a']
```

reverse 함수는 리스트를 역순으로 뒤집어 준다. 이때 리스트 요소들을 순서대로 정렬한 다음 다시 역순으로 정렬하는 것이 아니라 그저 현재의 리스트를 그대로 거꾸로 뒤집을 뿐이다.

위치 반환(index)

```
>>> a = [1,2,3]
>>> a.index(3)
2
>>> a.index(1)
0
```

index(x) 함수는 리스트에 x라는 값이 있으면 x의 위치값을 리턴한다.
위의 예에서 리스트 a에 있는 3이라는 숫자의 위치는 a[2]이므로 2를 리턴하고, 1이라는 숫자의 위치는 a[0]이므로 0을 리턴한다.

아래의 예에서 0이라는 값은 a 리스트에 존재하지 않기 때문에 값 오류(ValueError)가 발생한다.

```
>>> a.index(0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: 0 is not in list
```

리스트에 요소 삽입(insert)

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

insert(a, b)는 리스트의 a번째 위치에 b를 삽입하는 함수이다. 파이썬에서는 숫자를 0부터 센다는 것을 반드시 기억하자. 위의 예는 0번째 자리, 즉 첫 번째 요소(a[0]) 위치에 4라는 값을 삽입하라는 뜻이다.

```
>>> a.insert(3, 5)
>>> a
[4, 1, 2, 5, 3]
```

위의 예는 리스트 a의 a[3], 즉 네 번째 요소 위치에 5라는 값을 삽입하라는 뜻이다.

리스트 요소 제거(remove)

```
>>> a = [1, 2, 3, 1, 2, 3]
>>> a.remove(3)
>>> a
[1, 2, 1, 2, 3]
```

remove(x)는 리스트에서 첫 번째로 나오는 x를 삭제하는 함수이다.
a가 3이라는 값을 2개 가지고 있을 경우 첫 번째 3만 제

거되는 것을 알 수 있다.

```
>>> a.remove(3)
>>> a
[1, 2, 1, 2]
```

remove(3)을 한 번 더 실행하면 다시 3이 삭제된다.

리스트 요소 끄집어내기(pop)

```
>>> a = [1,2,3]
>>> a.pop()
3
>>> a
[1, 2]
```

pop()은 리스트의 맨 마지막 요소를 돌려 주고 그 요소는 삭제하는 함수이다.
a 리스트 [1,2,3]에서 3을 끄집어내고 최종적으로 [1, 2]만 남는 것을 볼 수 있다.
pop(x)는 리스트의 x번째 요소를 돌려 주고 그 요소는 삭제한다.

```
>>> a = [1,2,3]
>>> a.pop(1)
2
>>> a
[1, 3]
```

a.pop(1)은 a[1]의 값을 끄집어낸다. 다시 a를 출력해 보면 끄집어낸 값이 삭제된 것을 확인할 수 있다.

리스트에 포함된 요소 x의 개수 세기(count)

```
>>> a = [1,2,3,1]
>>> a.count(1)
2
```

count(x)는 리스트 내에 x가 몇 개 있는지 조사하여 그 개수를 돌려주는 함수이다.
1이라는 값이 리스트 a에 2개 들어 있으므로 2를 돌려준다.

리스트 확장(extend)

```
>>> a = [1,2,3]
>>> a.extend([4,5])
>>> a
[1, 2, 3, 4, 5]
>>> b = [6, 7]
>>> a.extend(b)
>>> a
[1, 2, 3, 4, 5, 6, 7]
```

extend(x)에서 x에는 리스트만 올 수 있으며 원래의 a 리스트에 x 리스트를 더하게 된다.
a.extend([4,5])는 a += [4,5]와 동일하다.