

## 파이썬 실습문제 4장-4 함수와 전역변수

문제 1) 함수호출시 전역변수와의 관계를 분석해 봄

1. my\_list = [1, 2, 3, 4]로 선언
2. 함수 my\_ave1()을 만들어 함수 내부에서 my\_list 에 5를 추가한 후 [1, 2, 3, 4, 5]의 평균값을 반환하게 함
3. 이 때, 함수 내부에서 my\_list 에 append 메소드를 써서 5를 추가함.
4. 함수 호출 후, 전역변수의 내용이 변화되었음을 이해함.
5. 출력예: 다음과 같이 출력함

```
my_list = [1, 2, 3, 4]. id(my_list)=10542856
```

함수 my\_ave1() 내부:

전역변수 my\_list=[1, 2, 3, 4, 5]로 변경

my\_list의 평균값은 my\_ave1()=3.0

함수 my\_ave1() 호출 후, my\_list = [1, 2, 3, 4, 5]. 변경 되었음.

문제 2) 위의 문제를 아래와 같이 변형하여 차이점을 이해함.

1. my\_list = [1, 2, 3, 4]로 다시 선언
2. 함수 my\_ave2()을 만들어 함수 내부에서 my\_list = [1, 2, 3, 4, 5] 으로 새로운 치환문을 사용함. my\_list 의 평균값을 반환하게 함
3. 함수 호출 후, 전역변수가 변하지 않은 상태임을 확인하고, 함수 내에서 전역변수와 같은 변수명을 썼는데 왜 위와 다른 결과인지를 이해함.

4. 출력예: 다음과 같이 출력함

```
my_list = [1, 2, 3, 4]. id(my_list)=10709384
```

함수 my\_ave2() 내부:

my\_list=[1, 2, 3, 4, 5]로 재할당. id(my\_list)=10542856로 전역변수 my\_list 와 달라짐. 이름은 같으나 이제 더 이상 같은 변수가 아님.

my\_list의 평균값은 my\_ave2(my\_list)=3.0.

함수 my\_ave2() 호출 후, my\_list = [1, 2, 3, 4]. 변경 안되었음.

문제 3) 위의 문제를 아래와 같이 변형하여 차이점을 이해함.

1. my\_list = [1, 2, 3, 4]로 다시 선언
2. global my\_list를 선언함.
3. 함수 my\_ave3()을 만들어 함수 내부에서 my\_list = [1, 2, 3, 4, 5] 으로 새로운 치환문을 사용함. my\_list 의 평균값을 반환하게 함
4. 함수 호출 이후 my\_list 의 값이 이번에는 변경되었음을 이해함.
5. 출력예: 다음과 같이 출력함

```
my_list = [1, 2, 3, 4]. id(my_list)=10542856
```

함수 my\_ave3() 내부:

my\_list=[1, 2, 3, 4, 5]로 재할당. id(my\_list)=10709384로 전역변수 my\_list 와 동일함.

my\_list의 평균값은 my\_ave3(my\_list)=3.0

함수 my\_ave3() 호출 후, my\_list = [1, 2, 3, 4, 5]. 변경 되었음.

문제 4) 위의 문제를 아래와 같이 변형하여 차이점을 이해함.

1. my\_list = [1, 2, 3, 4]로 다시 선언
2. 이번에는 매개변수를 갖는 함수 my\_ave4(my\_list)을 만들어 함수 호출시 my\_list를 전달함.
3. 함수 내부에서 my\_list 에 5를 추가하고, my\_list 의 평균값을 반환하게 함.
4. 함수 호출 이후 my\_list 의 값이 변경되었음을 이해함. 매개변수 my\_list 와 전역변수 my\_list 사이의 관계를 이해함.
5. 출력예: 다음과 같이 출력함

my\_list = [1, 2, 3, 4]. id(my\_list)=10542856

함수 my\_ave4() 내부:

전역변수 my\_list=[1, 2, 3, 4, 5]로 변경

매개변수 my\_list 의 id는 10542856. 전역변수 my\_list 와 서로 같은 메모리를 참조함

my\_list의 평균값은 my\_ave4(my\_list)=3.0

함수 my\_ave4() 호출 후, my\_list = [1, 2, 3, 4, 5]. 변경 되었음.

문제 5) 위의 문제를 아래와 같이 변형하여 차이점을 이해함.

1. my\_list = [1, 2, 3, 4]로 다시 선언
2. 이번에는 매개변수를 갖는 함수 my\_ave5(my\_list)을 만들어 함수 호출시 my\_list를 전달함.
3. 함수 내부에서 my\_list = [1, 2, 3, 4, 5]로 재할당 후, 그것의 평균값을 반환하게 함.
4. 함수 호출 이후 my\_list 의 값이 변경 안되었음을 이해함. 문제4와의 차이를 이해함.
5. 출력예: 다음과 같이 출력함

my\_list = [1, 2, 3, 4]. id(my\_list)=10709384

함수 my\_ave5() 내부:

my\_list = [1, 2, 3, 4]

my\_list 의 id는 10709384. 전역변수 my\_list 와 서로 같은 메모리를 참조함

my\_list = [1, 2, 3, 4, 5]로 재할당

my\_list 의 id는 10542856. 이제 전역변수 my\_list 와 서로 다른 메모리를 참조함

my\_list의 평균값은 my\_ave4(my\_list)=3.0.

함수 my\_ave5() 호출 후, my\_list = [1, 2, 3, 4]. 변경 안되었음.

문제 6) 위의 문제를 아래와 같이 변형하여 차이점을 이해함.

1. my\_list = [1, 2, 3, 4]로 다시 선언
2. 이번에는 매개변수 my\_list2를 갖는 함수 my\_ave6(my\_list2)을 만들어 함수 호출시 my\_list를 전달함.
3. 함수 내부에서 my\_list2 = [1, 2, 3, 4, 5]로 재할당 후, 그것의 평균값을 반환하게 함.
4. 함수 호출 이후 my\_list 의 값이 변경 안되었음을 확인하고. 문제5와 비교하여 매개변수명을 바꾸었으나 변수의 행동은 동일하였음을 이해함.
5. 출력예: 다음과 같이 출력함

my\_list = [1, 2, 3, 4]. id(my\_list)=10542856

함수 my\_ave6() 내부:

```
my_list2 = [1, 2, 3, 4]
```

my\_list2 의 id는 10542856. 전역변수 my\_list 와 서로 같은 메모리를 참조함

```
my_list2 = [1, 2, 3, 4, 5]로 재할당
```

my\_list2 의 id는 10709384. 이제 전역변수 my\_list 와 서로 다른 메모리를 참조함

my\_list의 평균값은 my\_ave6(my\_list)=3.0.

함수 my\_ave6() 호출 후, my\_list = [1, 2, 3, 4]. 변경 안되었음.