



영진전문대학교

글로벌시스템융합과 - GSC

자율학습용 코딩 문제 #7

정영철 교수

글로벌시스템융합과



A.I WITH BETTER LIFE

전역 변수의 활용 예

- 아래에 코드를 실행하고, 출력 결과에 대해 분석하시오.
- 전역변수(count):** 프로그램의 시작 시 전역(main) 메모리 영역에 할당되고 프로그램이 종료될 때까지 유지됩니다. 모든 함수와 스코프에서 접근 가능합니다. 이 예제에서 count는 프로그램 실행 초기에 메모리의 전역 영역에 할당되며, 이후 프로그램 전반에서 참조 및 수정이 가능합니다.
- 지역변수:** 함수가 호출될 때 스택 메모리에 할당되고, 함수 실행이 종료되면 해당 지역 변수에 할당된 메모리는 해제됩니다. 지역 변수는 선언된 함수 내에서만 접근 가능합니다. 이 코드 예제에는 명시적인 지역변수가 포함되어 있지 않으나, 지역 변수를 선언했다면 각 함수 호출 시 해당 함수의 스택 프레임 내에 할당되었을 것입니다.

```
1 # incrementCount 함수 정의:
2 # - 전역 변수 count의 값을 1 증가
3 def incrementCount():
4     global count
5     count = count + 1
6
7 # decrementCount 함수 정의:
8 # - 전역 변수 count의 값을 1 감소
9 def decrementCount():
10    global count
11    count = count - 1
12
13 # 전역 변수 count를 1로 초기화
14 count = 1
15
16 incrementCount()
17 print(count)
18
19 decrementCount()
20 print(count)
21
22 incrementCount()
23 print(count)
24 incrementCount()
25 print(count)
```

함수 호출 스택 변화 시각화

- 아래에 주어진 파이썬 코드를 실행했을 때, 프로그램의 스택 내 실행 문맥이 어떻게 변화하는지 단계별로 그림으로 그려 보세요.
- 2번 라인에서 'name' 변수에 접근할 때 스코프 체이닝(scope chaining)이 발생합니다. 스코프 체이닝의 작동 방식을 자세히 살펴보시기 바랍니다

```
1 def bar():
2     msg1 = "<< " + name
3
4     msg2 = foo(msg1)
5     msg2 += " "
6
7     msg3 = pos(msg2)
8     msg3 += " "
9
10    return msg3
11
12 def foo(argF):
13     msg = argF + "님"
14     return msg
15
16 def pos(argP):
17     msg = argP + "안녕하세요"
18     return msg
19
20 name = "홍길동"
21
22 result = bar()
23
24 print(result)
```

Q/A

감사합니다



주문식교육의 산실
영진전문대학교