

17. Python 이름 공간

1. Python 이름공간

1] 이름공간 정의

- 파이썬은 실행 시간에 각 이름들을 적절한 이름 공간에 넣어 관리
- 이름 공간(스코프)의 종류
 - 지역(Local): 각 함수 내부, 변수가 함수 내에서 정의되면 해당 함수의 지역 변수
 - 전역(Global): 모듈 (파일) 내부, 함수 내부에서 전역 변수를 직접 사용하고자 할 때, global 키워드 활용
 - 동일 함수 내에서 동일한 변수가 지역변수와 전역변수로 동시에 활용될 수 없음
 - 내장(Built-in): 파이썬 언어 자체에서 정의한 영역

예시 1 (전역 변수, 지역 변수)

g, h는 전역 변수

g = 10

h = 5

def f1(a): # a는 지역 변수

 h = a + 10 # h는 지역, 새로 l-value로 정의했음

 b = h + a + g # b도 지역, g는 r-value이므로 기존 값을 참조 - 전역 변수

return b

print ("f1(h)->", f1(h)) # 함수 호출시에 사용되는 변수는 해당 위치의 스코프에서 값을 찾음

전역 변수

print ("h->" , h) # 전역 변수 h는 변함 없음

x = 20 # 가장 안쪽의 스코프부터 바깥쪽의 스코프쪽으로 변수를 찾는다

def F():

 x = 5 # 아래 x는 위 x와 다른 f 함수의 지역변수

def G():

 print ("x->" , x) # 함수 안에 print x를 하면 가까운 x를 먼저 찾음

 G() # G() → 중첩영역을 지원하는 함수

F()

결과

f1(h)-> 30

h-> 5

x-> 5

1. Python 모듈/함수

2] 모듈/함수의 정의

- 함수: 파일 내에서 일부 코드를 묶는 것
- 모듈: 파일 단위로 코드들을 묶는 것, 비슷하거나 관련된 일을 하는 함수나상수값들을 모아서 하나의 파일에 저장하고 추후에 재사용하는 단위(모듈을 사용하는 측에서는 모듈에 정의된 함수나 변수 이름을 사용)

3] 모듈 장점

- 코드의 재사용
- 프로그램 개발시에 전체 코드들을 여러 모듈 단위로 분리하여 설계함으로써 작업의 효율을 높일 수 있음
- 별도의 이름 공간(스코프)를 제공함으로써 동일한 이름의 여러 함수나 변수들이 각 모듈마다 독립적으로 정의.
- 모듈은 하나의 독립된 이름 공간을 확보하면서 정의

예시 1

```
import ch17.mymath
print ("dir(ch17.mymath)->",dir(ch17.mymath) )
print ("ch17.mymath.mypi ->",ch17.mymath.mypi )
print ("ch17.mymath.area(5)->",ch17.mymath.area(5))
```

mymath 에 정의된 이름들 확인하기
mymath 안에 정의된 mypi 를 사용한다
mymath 안에 정의된 area 를 사용한다

결과

```
-----
dir(ch17.mymath)-> ['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'area', 'mypi']
ch17.mymath.mypi -> 3.14
ch17.mymath.area(5)-> 78.5
```

1. Python class 이름공간

예시 1

```
class C: # 클래스도 독립적인 이름 공간
    a = 2 # 클래스 이름 공간 내에 변수 선언
    pass # 클래스 정의
c = C() # 클래스 인스턴스 객체 생성
c.a = 1 # 클래스에서 생성된 인스턴스 객체도 별도의 이름 공간
print("c.a->", c.a)
print("c.__class__.a->", c.__class__.a)
print("C.a->", C.a)
```

결과

```
-----
c.a-> 1
c.__class__.a-> 2
C.a-> 2
```