

파이썬 프로그래밍

---

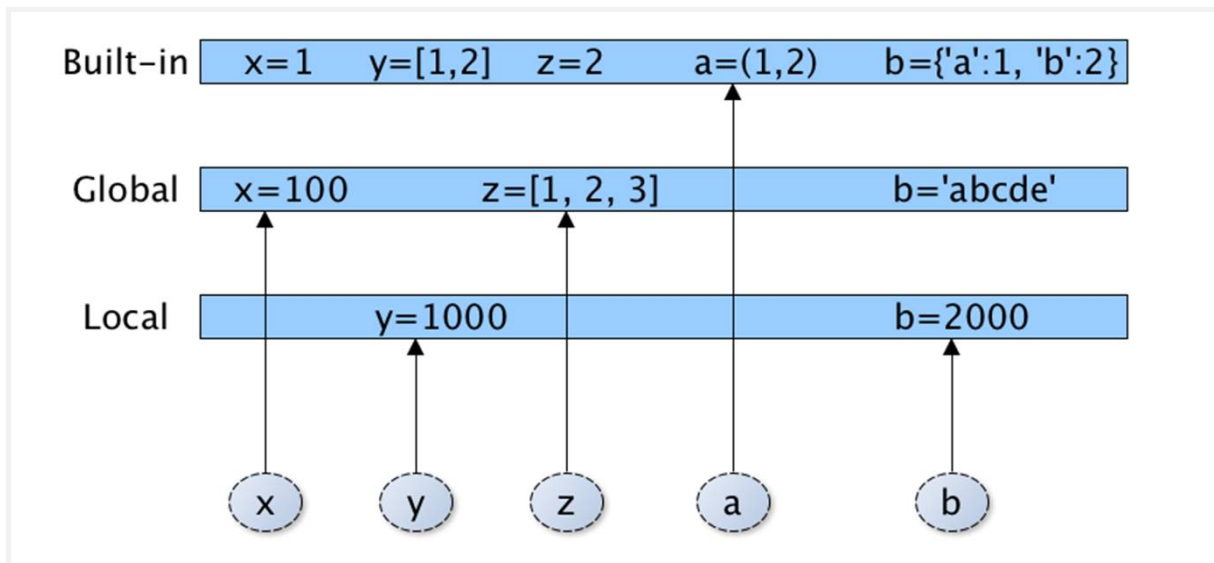
# 파이썬 모듈



한국기술교육대학교  
온라인평생교육원

## ■ 이름 공간

- 이름 공간 또는 스코프 (Naming Space or Scope): 이름이 존재하는 장소.  
파이썬은 실행 시간에 각 이름들을 적절한 이름 공간에 넣어 관리한다.
- 이름 공간(스코프)의 종류
  - 지역(Local): 각 함수 내부
  - 전역(Global): 모듈 (파일) 내부
  - 내장(Built-in): 파이썬 언어 자체에서 정의한 영역
- 변수가 정의되는 위치에 의해 변수의 스코프가 정해짐
  - 파이썬에서 변수의 정의
    - 변수가 l-value로 사용될 때
- 변수가 r-value로 사용될 때 해당 변수의 값을 찾는 순서 규칙
  - L --> G --> B



- 지역(Local) : 일반적으로 각 함수 내부, 클래스 메소드 안. 객체 내부
- 일정한 공간 내에 존재하는 것 = local
- 모듈의 물리적인 단위 = 파일
- 내장(built-in) : 파이썬 언어 자체에서 정의한 영역
- `a = 1` 에서 a라고 하는 것이 =의 왼쪽에 존재 시 a는 l-value
- `b = a` 를 하게 되면 b는 l-value, a는 r-value
- a 변수와 b 변수는 동일한 파일 내 존재하는 전역(global) 변수가 됨
- L → G → B : a 값을 local, global, built-in 순서로 찾음
- b는 local 영역에서 정의가 되었기 때문에 local 영역 정의 사용

## ■ 이름 공간

### 1. 지역변수와 전역변수

- 변수의 스코프는 해당 변수가 l-value로서 정의되는 위치에 따라 달라짐
- 변수가 함수 내에서 정의되면 해당 함수의 지역 변수가 된다.

```
# g, h는 전역 변수
```

```
g = 10
```

```
h = 5
```

```
def f(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 f(h) # 함수 호출시에 사용되는 변수는 해당 위치의 스코프에서 값을 찾음  
          - 전역 변수
```

```
print h # 전역 변수 h는 변함 없음
```

```
30
```

```
5
```

- g, h 두 변수는 함수 바깥에 위치하므로 전역 변수
- a 인자는 f 함수 안에 존재하는 지역(local) 변수
- h 변수는 f 함수 내에 있으므로 지역변수가 됨
- 지역 변수 안 a는 r-value로 기존에 존재하는 것 활용 → 지역변수a
- b도 f 함수 내 지역변수
- h + a + g는 전부 r-value로 기존에 존재하는 것 활용
- g 값의 경우 로컬 내 존재하지 않으므로 전역 변수 활용
- h는 같은 전역 변수 내에서 찾아 값 기입
- 지역 변수 내 h와 전역 변수 h는 다른 값

## ■ 이름 공간

### 1. 지역변수와 전역변수

- 함수 내부에서 전역 변수를 직접 사용하고자 할 때
  - global 키워드 활용

```
h = 5

def f(a):    # a는 지역
    global h # h 변수를 전역이라고 미리 선언함
    h = a + 10 # h는 l-value로 정의되더라도 미리 선언된 내용 때문에 전역 변수
    return h

print f(10)
print h      # 전역 변수 h 값이 함수 내에서 변경되었음
```

```
20
20
```

- f 함수 내 h는 지역변수
- global h 키워드를 적으면 h가 전역 변수 사용으로 선언됨
- h = 5 가 사라지고 새로운 값 20이 들어감

## ■ 이름 공간

### 1. 지역변수와 전역변수

- [주의] 동일 함수 내에서 동일한 변수가 지역변수와 전역변수로 동시에 활용될 수 없음
  - 함수 내에서 정의되는 변수는 지역 변수로 간주
  - 지역 변수로 선언되기 이전에 해당 변수를 사용할 수 없음

```
g = 10
```

```
def f():  
    a = g    # l-value로 사용되는 g는 전역 변수  
    g = 20   # r-value로 정의되는 g는 지역 변수  
    return a
```

```
print f()
```

```
-----  
UnboundLocalError                                Traceback (most recent call last)  
<ipython-input-68-e323361344da> in <module>()  
      6     return a  
      7  
----> 8 print f()  
  
<ipython-input-68-e323361344da> in f()  
      2  
      3 def f():  
----> 4     a = g    # l-value로 사용되는 g는 전역 변수  
      5     g = 20   # r-value로 정의되는 g는 지역 변수  
      6     return a
```

```
UnboundLocalError: local variable 'g' referenced before assignment
```

- a라는 변수는 함수 내에서 새로 만들어지는 것
- g는 local 변수 내에 없으므로 전역변수 값 사용
- global g 와 같은 내용이 없기 때문에 l-value의 g는 새롭게 선언
- 그러면 g는 함수 안에 존재하는 지역변수가 됨
- g가 지역변수이면서 전역변수이기 때문에 error 발생

---

## ■ 이름 공간

### 1. 지역변수와 전역변수

```
g = 10

def f():
    global g    # g는 전역 변수로 선언됨
    a = g       # a는 지역 변수, g는 전역 변수
    g = 20      # g는 전역 변수
    return a

print f()
```

10

- a=g 가 아래 쓰이면 error 발생하지 않음
- a=g의 g는 지역변수

## ■ 이름 공간

### 2. 특정 공간의 이름 목록 얻기

- 이름(Name) : 변수 (객체) 이름
  - 함수 이름
  - 클래스 이름
  - 모듈 이름
- dir(): 함수가 호출된 스코프에서 정의되어 있는 모든 Name들을 문자열 리스트로 반환한다.
- dir(object): object이 지니고 있는 모든 Name들을 문자열 리스트로 반환한다.

```
l = []  
print dir(l)
```

```
['_add_', '__class__', '__contains__', '__delattr__', '__delitem__', '__delslice__', '__doc__',  
'_eq_', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getslice__', '__gt__',  
'_hash_', '__iadd__', '__imul__', '__init__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',  
'_ne_', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__',  
'_setattr__', '__setitem__', '__setslice__', '__sizeof__', '__str__', '__subclasshook__',  
'append', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

- dir() 내장함수 → 특정개체, 모듈 내 존재하는 이름들을 리스트로 반환
- 파이썬의 모든 것은 object이 될 수 있음
- object = 함수, 특정객체, 클래스, 인스턴스, 모듈의 이름 등
- '\_'로 시작되는 것들은 특수한 변수 또는 메소드
- 이 에 존재하는 각각의 식별자

## ■ 이름 공간

### 3. 함수의 중첩 영역(Nested Scopes) 지원

- Nested Scope: 함수 안에 정의되어 있는 함수 내부
  - 가장 안쪽의 스코프부터 바깥쪽의 스코프쪽으로 변수를 찾는다.

```
x = 2
def F():
    x = 1
    def G():
        print x
    G()

F()
```

1

- 아래 x는 위 x와 다른 f 함수의 지역변수
- G() → 중첩영역을 지원하는 함수
- 함수 안에 print x를 하면 가까운 x를 먼저 찾음