Python’s scopes—the places where variables are defined and looked up

variables may be assigned in three different places, corresponding to three different scopes:

• If a variable is assigned inside a def, it is local to that function.

• If a variable is assigned in an enclosing def, it is nonlocal to nested functions.

• If a variable is assigned outside all defs, it is global to the entire file.

```
X = 99
def func():
    X = 88
```

In the following module file, the X = 99 assignment creates a global variable named X (visible everywhere in this file), but the X = 88 assignment creates a local variable X (visible only within the def statement).

```
# Global scope
X = 99 # X and func assigned in module: global
def func(Y): # Y and Z assigned in function: locals
    # Local scope
    Z = X + Y # X is a global
    return Z
func(1) # func in module: result=100
```

```
100
```

Start coding or generate with AI.