PYTHON BASICS: SCOPES

Python Basics: Scopes

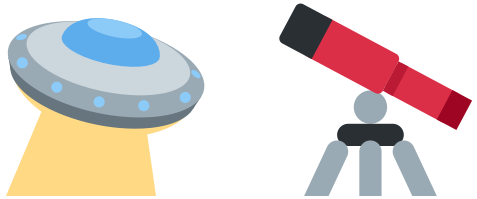


Table of Contents

Understanding Scope

```
total = 5
```

Using the LEGB Rule

1. Local scope
2. Enclosing scope
3. Global scope
4. Built-in scope

Honey, have you seen my sunglasses?

Scope	🕶❓
Local	😎
Enclosing	💼
Global	🏘
Built-in	😆

Exploring the Local, Enclosing, and Global Scope

1. **Local scope**
2. **Enclosing scope**
3. **Global scope**
4. Built-in scope

Local Scope

- Contains the names that you define inside a function
- Every time you call a function, you’re also creating a new local scope

Enclosing Scope

- Is a special scope that only exists for nested functions
- Names that you define in the enclosing scope are commonly known as nonlocal names

Global Scope

- Is the top-most scope in a Python program, script, or module
- Names are visible from everywhere in your Python code
- There’s only one global Python scope per program execution

Next Up: Built-in Scope

1. Local scope
2. Enclosing scope
3. Global scope
4. **Built-in scope**

Inspecting the Built-in Scope

1. Local scope
2. Enclosing scope
3. Global scope
4. **Built-in scope**

Built-In Scope

- Is automatically loaded by Python when you run a program or script
- Contains names that are built into Python
- Is implemented as a standard library module named `builtins`

The LEGB Rule

1. Local scope
2. Enclosing scope
3. Global scope
4. Built-in scope

Using the global Statement

```
counter = 0

def update_counter():
    counter = counter + 1

update_counter()
```

Preventing Pitfalls

Good Programming Practices

- Use local names rather than global names.
- Write self-contained functions.
- Try to use unique objects names, no matter what scope you’re in.
- Avoid global name modifications throughout your programs.

```
counter = 0

def update_counter():
    global counter
    counter = counter + 1

update_counter()
print(counter)
```

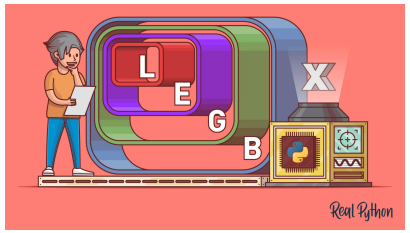
```
counter = 0

def update_counter(current_counter):
    return current_counter + 1

counter = update_counter(counter)
print(counter)
```

Summary and Additional Resources

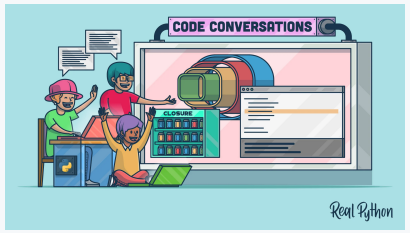
In this course you learned:

1. What scopes are and how they work in Python
2. Why it's important to know about Python scope
3. What the LEGB rule is and how Python uses it to resolve names
4. How to use the `global` statement
5. How to implement good programming practices

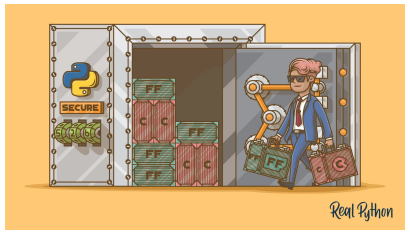
Additional Resources



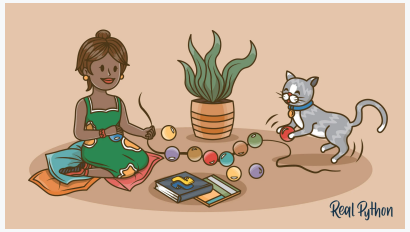
Namespaces and Scope in Python



Exploring Scopes and Closures in Python



Python Inner Functions



Python 3’s f-Strings