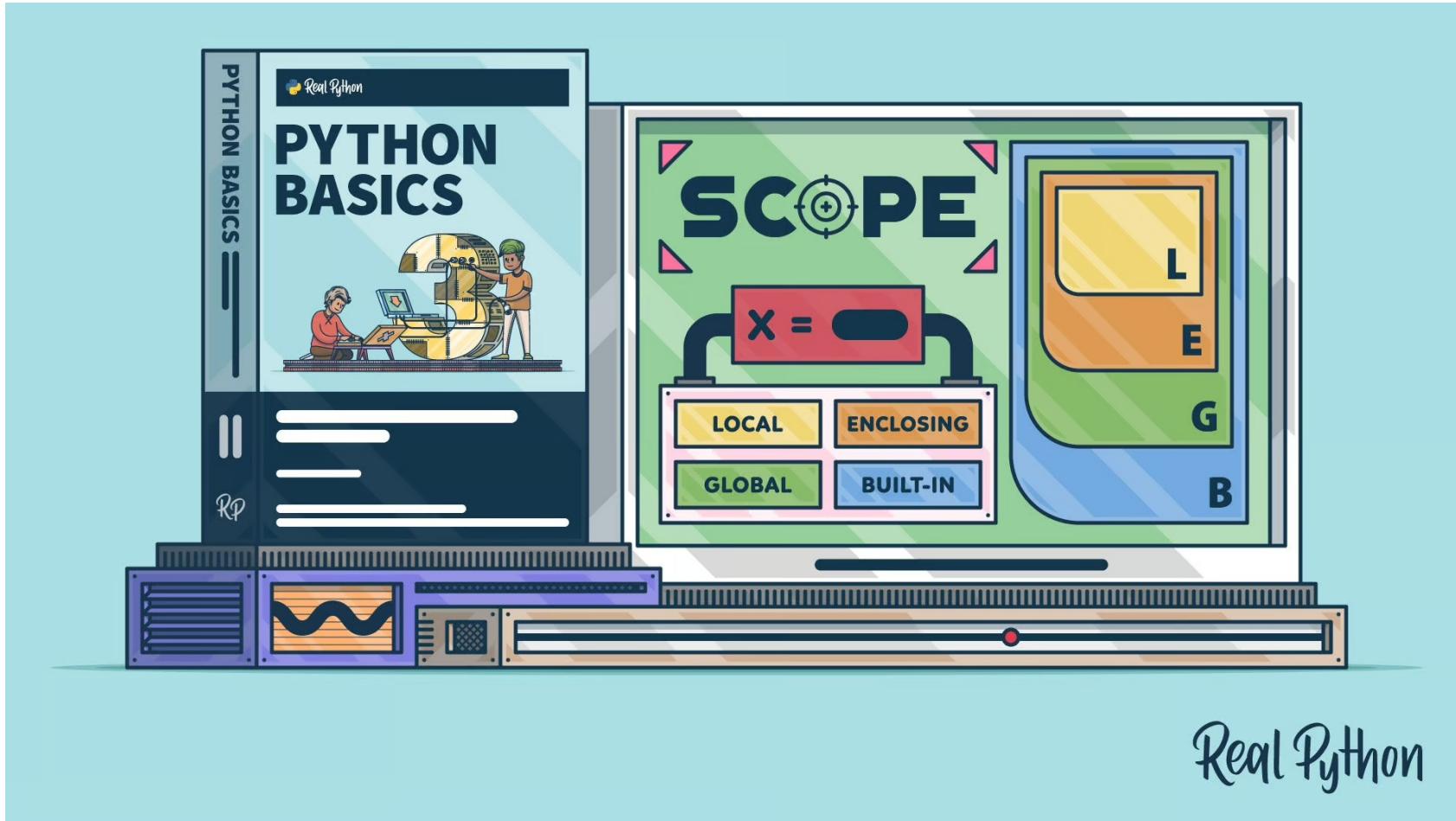


Python Basics Exercises: Scopes



Real Python

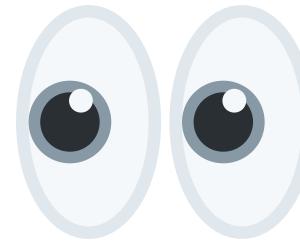
Real Python Exercises Course



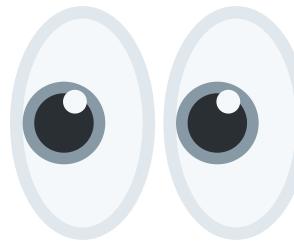
Real Python Exercises Course



Real Python Exercises Course



Real Python Exercises Course

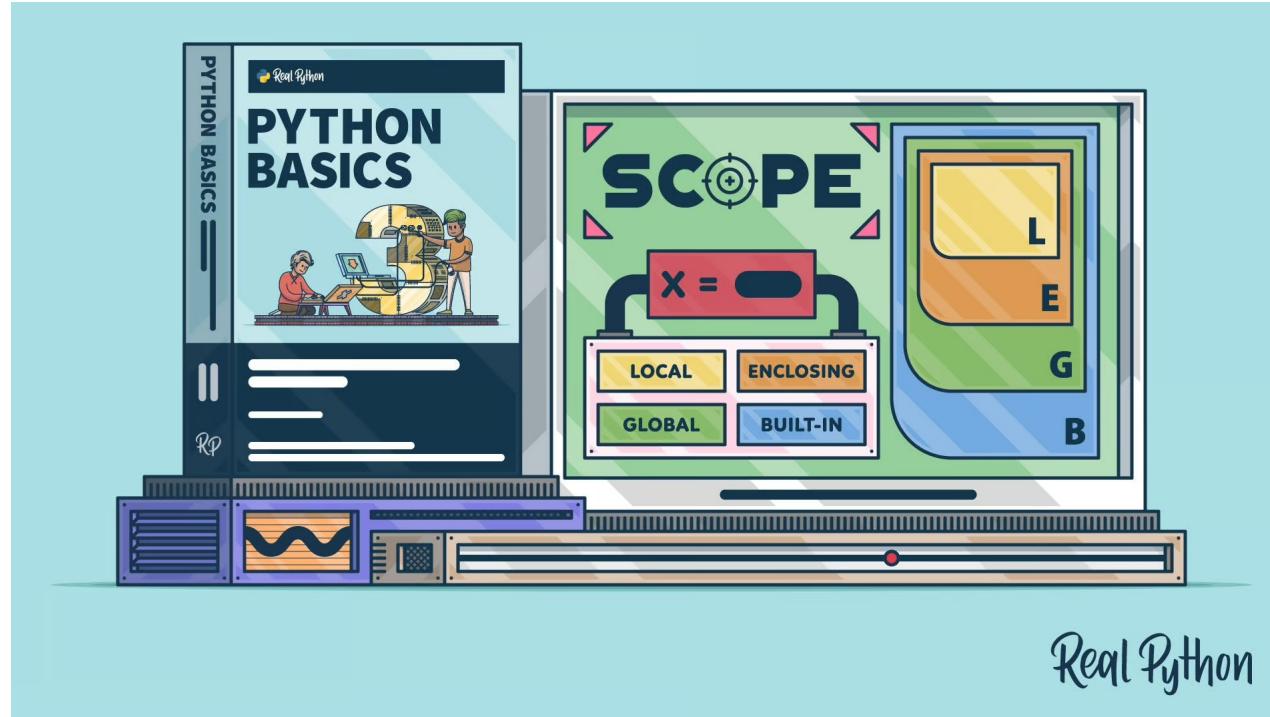


Real Python Exercises Course

The three steps for each task:

1.  Learn about the exercise
2.  Code your solution
3.  Compare your solution

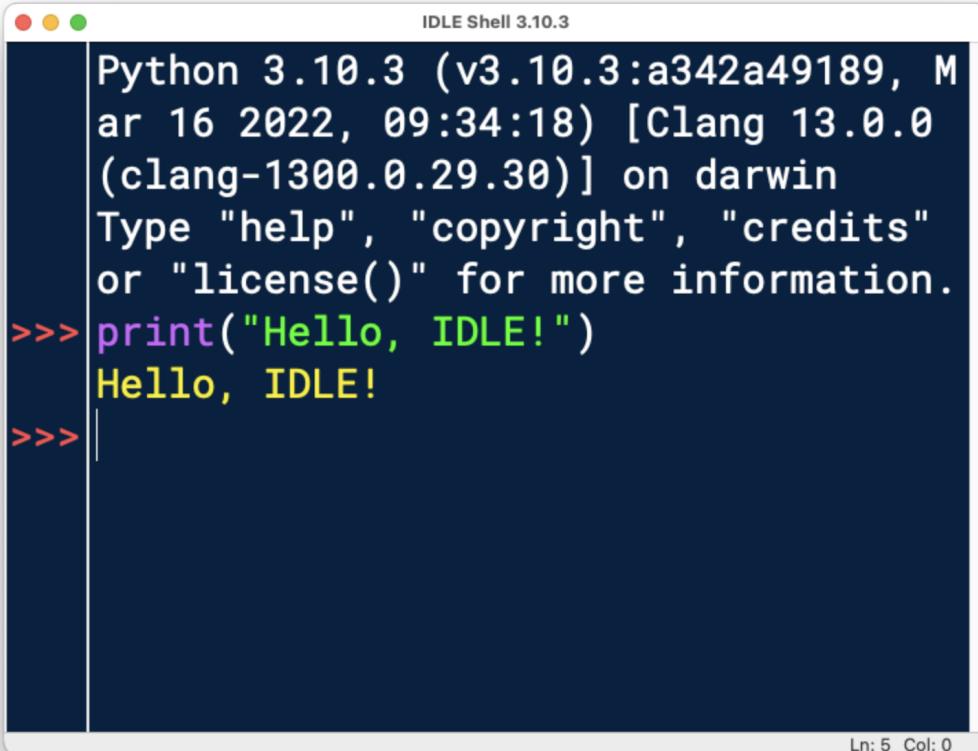
Background - Python Basics: Scopes



Background - Python Basics: Scopes

- Spotting **Scopes**
- Working With the `global` Keyword
- Exploring `locals()`
- Using the `return` Statement
- Organizing a Fun Party 🎉

Background - Using IDLE



IDLE Shell 3.10.3

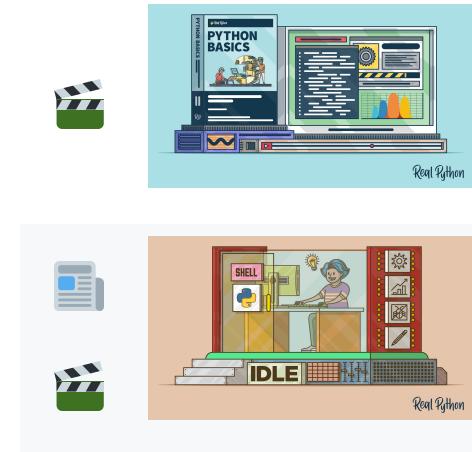
```
Python 3.10.3 (v3.10.3:a342a49189, Mar 16 2022, 09:34:18) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits"
or "license()" for more information.

>>> print("Hello, IDLE!")
Hello, IDLE!

>>>
```

Ln: 5 Col: 0

Additional Resources



- Python Basics:
Setting Up Python
- Getting Started With
Python IDLE

Ready to Get Started?



Exercise: Mouse in the Cabinet

```
def explore_basement():
    def explore_cabinet():
        """"""
        cabinet_items = ["keys", "sunglasses"]
        print(cabinet_items)

        basement_items = ["bed", "cabinet"]
        explore_cabinet()
        print(basement_items)

address = "Python Palace"
explore_basement()
print(address)
```

In the program on the left, you can find these variables and functions:

- `address`
- `cabinet_items`
- `basement_items`
- `print()`

From the perspective of the mouse in the cabinet, order them into the four scopes:

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



Solution: Mouse in the Cabinet

```
def explore_basement():
    def explore_cabinet():
        """"""
        cabinet_items = ["keys", "sunglasses"]
        print(cabinet_items)

    basement_items = ["bed", "cabinet"]
    explore_cabinet()
    print(basement_items)

address = "Python Palace"
explore_basement()
print(address)
```

Scope	Variable or Function
Local	cabinet_items
Enclosing	basement_items
Global	address
Built-in	print()



Exercise: Ask For the Addresses

You decide to throw a party with the mouse. After cleaning up all the items that are lying around, you want to hang up some signs so the guests can find their way into the mouse's cabinet.



Exercise: Ask For the Addresses

Have a look at the program below. Can you guess how the output of the program looks?

```
def explore_basement():
    def explore_cabinet():
        """ """
        address = "Cookie Cabinet"
        print(address)

    address = "Mouse House"
    explore_cabinet()
    print(address)

address = "Python Palace"
explore_basement()
print(address)
```



Solution: Ask For the Addresses

Cookie Cabinet

Mouse House

Python Palace



Exercise: Adjust the Address

You want to invite other animals to the party.

However, "Python Palace" may sound a bit scary to other rodents.



Exercise: Adjust the Address

Add one line to the function body of `explore_cabinet()` to set the global `address` to "Cookie Cabinet".

```
def explore_basement():
    def explore_cabinet():
        """
        address = "Cookie Cabinet"
        print(address)

        address = "Mouse House"
        explore_cabinet()
        print(address)

    address = "Python Palace"
    explore_basement()
    print(address)
```



Solution: Adjust the Address

```
def explore_basement():
    def explore_cabinet():
        """"""
        global address
        address = "Cookie Cabinet"
        print(address)

    address = "Mouse House"
    explore_cabinet()
    print(address)

address = "Python Palace"
explore_basement()
print(address)
```



Exercise: Deliver the Invitation

You also want to invite a bear to the party. To make sure the bear receives the invitation, you decide to hand it over personally.



Exercise: Deliver the Invitation

Adjust the function definition of `visit_woods()` to accept a string as an argument:

```
def visit_woods():
    """
    if "my_invitation" in locals():
        print(my_invitation)

invitation = "Let's have a party!"
visit_woods()
```

Then, update the function call of `visit_woods()` so the program prints the value of `my_invitation`.



Solution: Deliver the Invitation

```
def visit_woods(my_invitation):
    """Bear"""
    if "my_invitation" in locals():
        print(my_invitation)

invitation = "Let's have a party!"
visit_woods(invitation)
```



Exercise: Collect All Cookies

There's no party without cookies!

Look on the shelf and under to sofa to collect all the cookies.

To avoid crumbling any precious cookies, you're not allowed to throw them. Instead, you need to carefully hand over every cookie list with a `return` statement.



Exercise: Collect All Cookies

Update the program below so the output looks like this:

```
[ "Peanut", "Chocolate", "Oat", "Salted Caramel"]
```

```
def on_the_shelf():
    shelf_cookies = [ "Peanut", "Chocolate"]

def under_the_sofa():
    sofa_cookies = [ "Oat", "Salted Caramel"]

cookies = []
print(cookies)
```



Solution: Collect All Cookies

```
def on_the_shelf():
    shelf_cookies = ["Peanut", "Chocolate"]
    return shelf_cookies

def under_the_sofa():
    sofa_cookies = ["Oat", "Salted Caramel"]
    return sofa_cookies

cookies = []
cookies += on_the_shelf()
cookies += under_the_sofa()
print(cookies)
```



Exercise: Rename the Palace

The party is over and it was a great success!

You decide to rename the "Python Palace" into "Mouse House".
For nostalgic reasons, you want to keep the old address sign around.



Exercise: Rename the Palace

Adjust the code below, so the last line of the program prints "Mouse House".

Keep the line `address = "Python Palace"` untouched.

```
def explore_basement():
    def explore_cabinet():
        """"""
        global address
        address = "Cookie Cabinet"
        print(address)

    address = "Mouse House"
    explore_cabinet()
    print(address)

address = "Python Palace"
explore_basement()
print(address)
```



Solution: Rename the Palace

```
def explore_basement():
    def explore_cabinet():
        """"""
        # Remove: global address
        address = "Cookie Cabinet"
        print(address)

        address = "Mouse House"
        explore_cabinet()
        print(address)
        return address

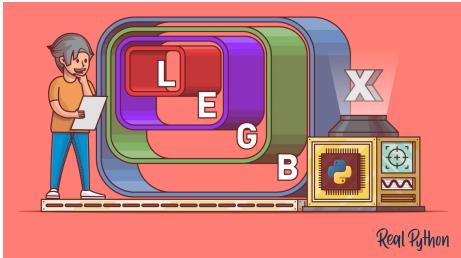
address = "Python Palace"
address = explore_basement()
print(address)
```

Summary and Additional Resources

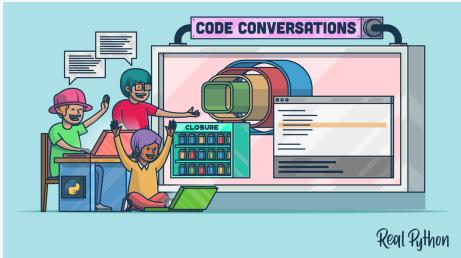
In this course, you practiced how to:

- Spot **Scopes**
- Work With the `global` **Keyword**
- Explore `locals()`
- Use the `return` **Statement**
- Organize a Fun **Party** 

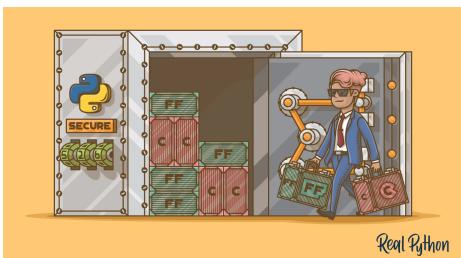
Additional Resources



Namespaces and Scope in Python



Exploring Scopes and Closures in Python



Python Inner Functions

Congratulations and Thanks!

