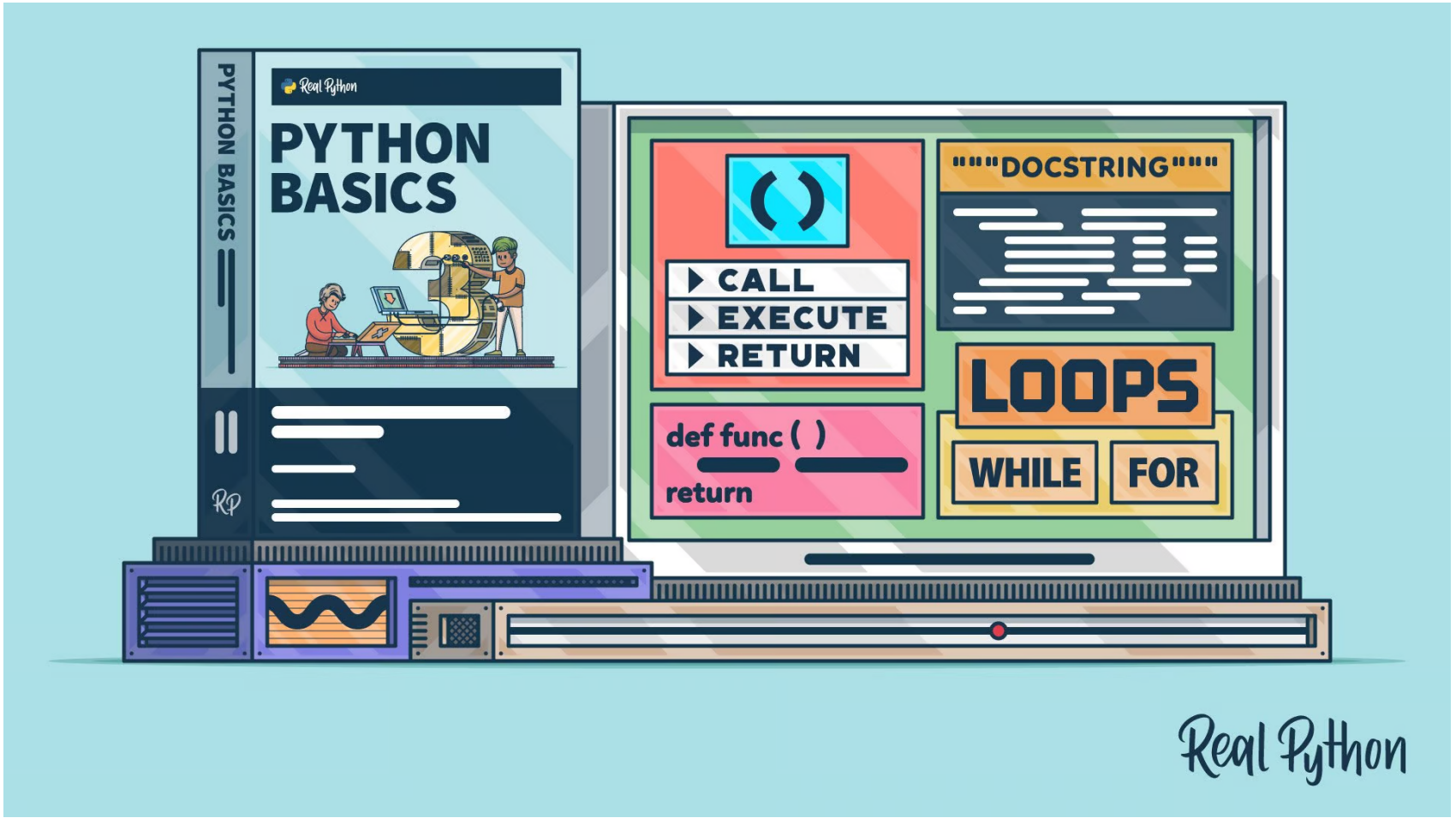
Python Basics Exercises: Functions and Loops

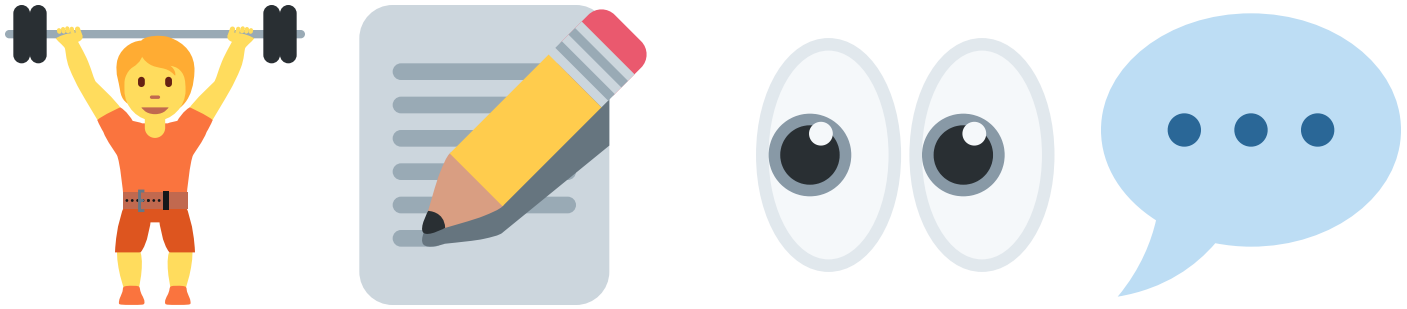


Real Python Exercises Course

Real Python Exercises Course

Real Python Exercises Course

Real Python Exercises Course



Real Python Exercises Course

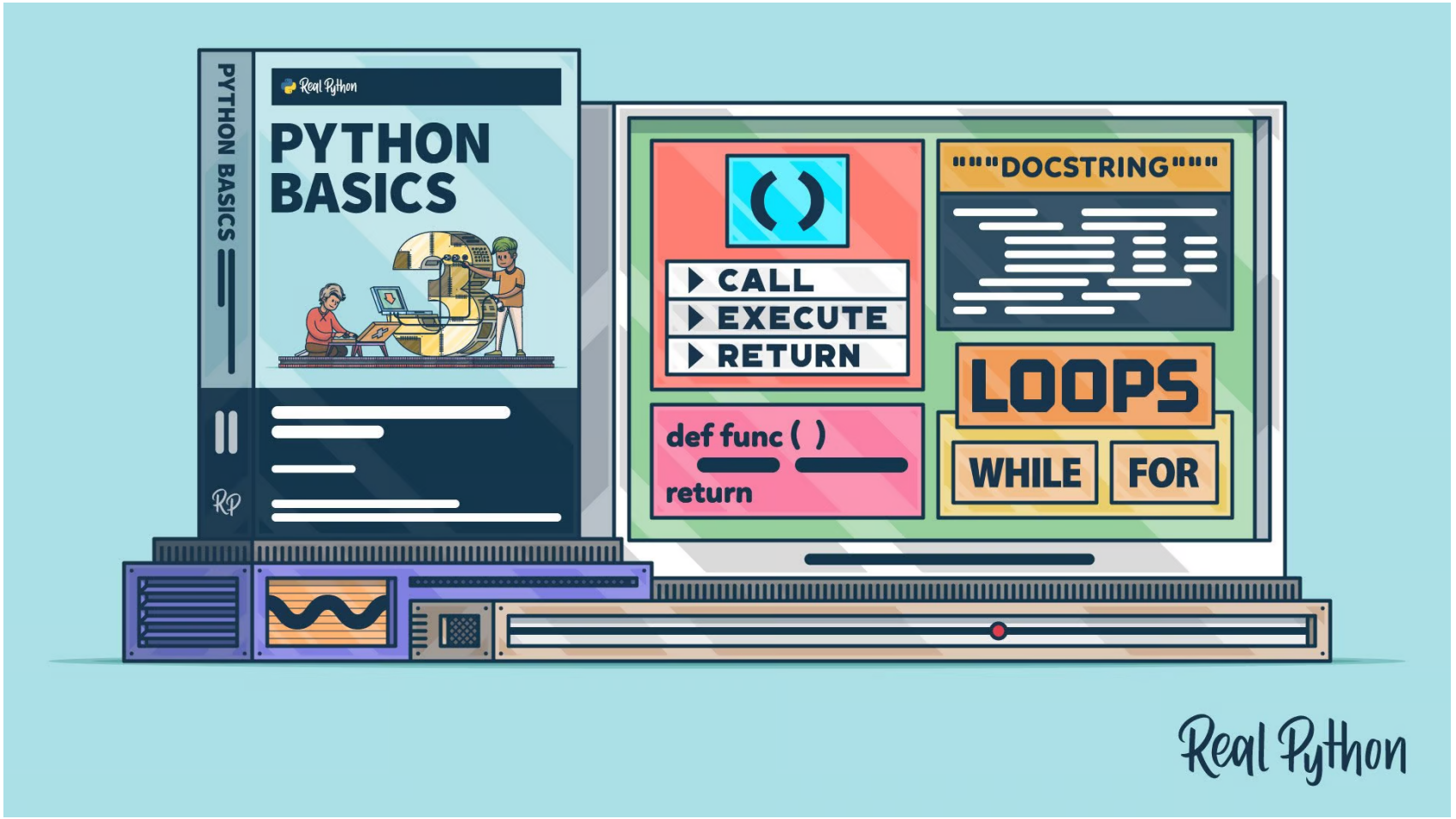
The three steps for each task:

1. 👀 Learn about the exercise
2. 👨‍💻 Code your solution
3. ↔️ Compare your solution

Python Basics Exercises: Functions and Loops

1. Review Exercises
2. Challenge
3. More Review Exercises
4. Another Challenge

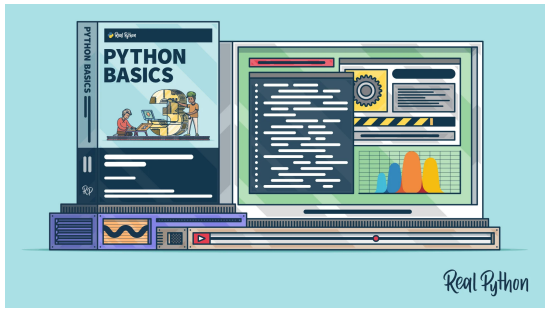
Background - Python Basics: Functions and Loops



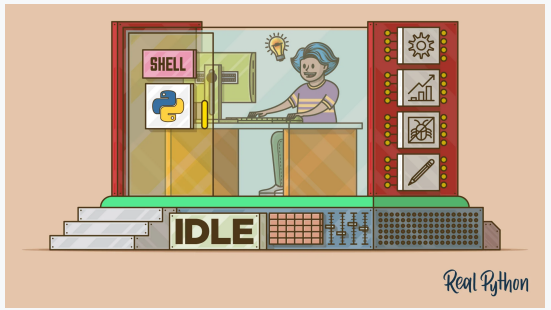
Background - Python Basics: Functions and Loops

- Define a Function
- Ask for User Input
- Do Basic Arithmetic
- Run `for` Loops

Background - Using IDLE



Python Basics: Setting Up Python



Getting Started With Python IDLE

Ready to Get Started?

🤔 Exercise: Greet Someone

Write a function called `greet()` that takes one string parameter called `name` and displays the text `"Hello, <name>!"`, where `<name>` is replaced by the value of the `name` parameter.

💡 Solution: Greet Someone

```
def greet(name):
    print(f"Hello, {name}!")
```

🤔 Exercise: Cube

Write a function called `cube()` that takes one number parameter and returns the value of that number raised to the third power. Test the function by calling your `cube()` function on a few different numbers and displaying the results.

💡 Solutions: Cube

```
def cube(num):
    cube_num = num**3
    return cube_num
```

```
def cube(num):
    cube_num = pow(num, 3)
    return cube_num
```

Tips

- Use code comments to help you get organized
- Break exercises into smaller tasks
- Use descriptive variable names
- Test repeatedly to see whether the code does what you expect it to do

Tips

- Use code comments to help you get organized
- Break exercises into smaller tasks
- Use descriptive variable names
- Test repeatedly to see whether the code does what you expect it to do

Next Up: Tackle Your First Challenge!

🤔 Challenge: Convert Temperatures

Write a program called `temperature.py` that defines two functions:

1. `convert_cel_to_far()`, which takes one `float` parameter representing degrees Celsius and returns a `float` representing the same temperature in degrees Fahrenheit using the following formula:

```
F = C * 9/5 + 32
```

2. `convert_far_to_cel()`, which takes one `float` parameter representing degrees Fahrenheit and returns a `float` representing the same temperature in degrees Celsius using the following formula:

```
C = (F - 32) * 5/9
```

🤔 Challenge: Convert Temperatures

The program should do the following:

1. Prompt the user to enter a temperature in degrees Fahrenheit and then display the temperature converted to Celsius
2. Prompt the user to enter a temperature in degrees Celsius and then display the temperature converted to Fahrenheit
3. Display all converted temperatures rounded to two decimal places

🤔 Challenge: Convert Temperatures

Here's a sample run of the program:

```
Enter a temperature in degrees F: 72
72 degrees F = 22.22 degrees C

Enter a temperature in degrees C: 37
37 degrees C = 98.60 degrees F
```

💡 Challenge: Convert Temperatures

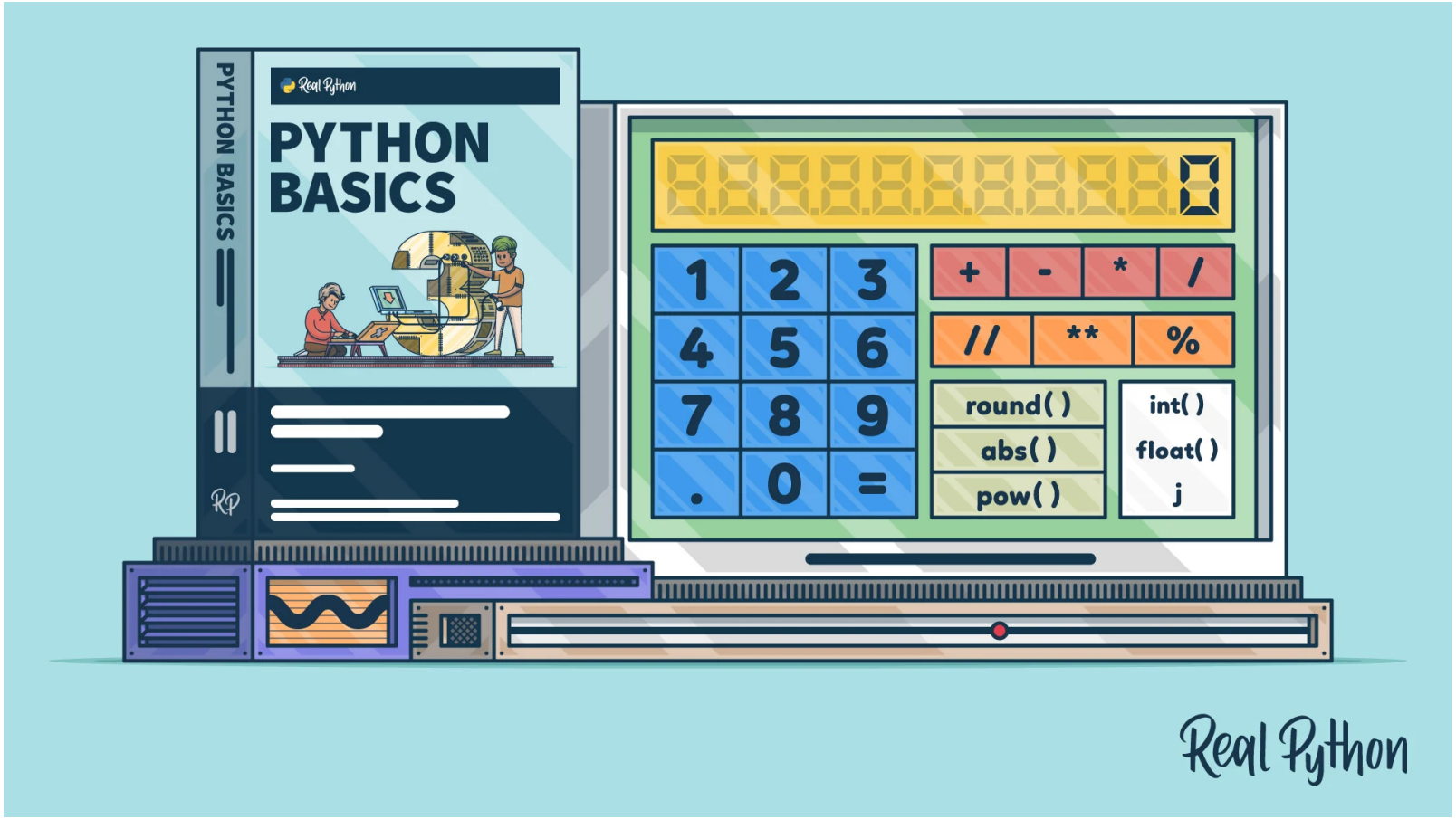
```
def convert_cel_to_far(temp_cel):
    temp_far = temp_cel * (9 / 5) + 32
    return temp_far

def convert_far_to_cel(temp_far):
    temp_cel = (temp_far - 32) * (5 / 9)
    return temp_cel

temp_far = input("Enter a temperature in degrees F: ")
temp_cel = convert_far_to_cel(float(temp_far))
print(f"{temp_far} degrees F = {temp_cel:.2f} degrees C")

temp_cel = input("\nEnter a temperature in degrees C: ")
temp_far = convert_cel_to_far(float(temp_cel))
print(f"{temp_cel} degrees C = {temp_far:.2f} degrees F")
```

Python Basics: Numbers and Math



🤔 Exercise: Print Integers

Write a `for` loop that prints out the integers `2` through `10`, each on a new line, using `range()`.

💡 Solution: Print Integers

```
for i in range(2, 11):
    print(i)
```

🤔 Exercise: Display the Doubles

Write a function called `doubles()` that takes a number as its input and doubles it. Then use `doubles()` in a loop to double the number `2` three times, displaying each result on a separate line.

Here's some sample output:

```
4
8
16
```

💡 Solution: Display the Doubles

```
def doubles(num):
    return num * 2

my_num = 2
for i in range(0, 3):
    my_num = doubles(my_num)
    print(my_num)
```

🤔 Challenge: Track Your Investment

Write a program called `invest.py` that tracks the growing amount of an investment over time.

The initial deposit for an investment is called the principal amount. Each year, the amount increases by a fixed percentage, called the annual rate of return.

🤔 Challenge: Track Your Investment

Write a function called `invest()` with three parameters: the principal amount, the annual rate of return, and the number of years to calculate. The function signature might look something like this:

```
def invest(amount, rate, years):
```

The function should then print out the amount of the investment, rounded to two decimal places, at the end of each year for the specified number of years.

🤔 Challenge: Track Your Investment

For example, calling `invest(100, .05, 4)` should print the following:

```
year 1: $105.00
year 2: $110.25
year 3: $115.76
year 4: $121.55
```

To finish the program, prompt the user to enter an initial amount, an annual percentage rate, and a number of years. Then call `invest()` to display the calculations for the values entered by the user.

💡 Challenge: Track Your Investment

```
def invest(amount, rate, years):
    for year in range(1, years + 1):
        amount = amount * (1 + rate)
        print(f"year {year}: ${amount:,.2f}")


amount = float(input("Enter a principal amount: "))
rate = float(input("Enter an anual rate of return: "))
years = int(input("Enter a number of years: "))

invest(amount, rate, years)
```

Summary and Additional Resources

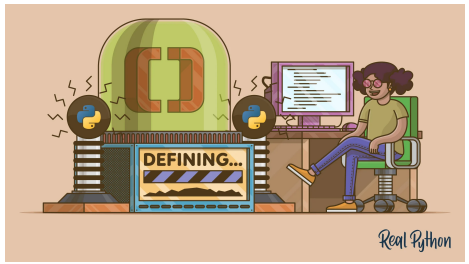
In this course, you **practiced** using:

- Functions
- User Input
- Basic Arithmetic
- `for` Loops

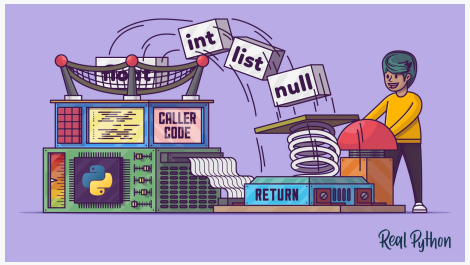
Tips

- Use code comments to help you get organized
- Break exercises into smaller tasks
- Use descriptive variable names
- Test repeatedly to see whether the code does what you expect it to do

Additional Resources



Defining Your Own Python Function



The Python return Statement: Usage and Best Practices

Python “for” Loops (Definite Iteration)

Congratulations and Thanks!