Welcome to Python! 🚀

What is Python?

Python is a powerful and beginner-friendly programming language with clean, readable syntax that's almost like plain English.

Why Choose Python?

- **Easy to Learn:** Simple syntax makes it perfect for beginners
- **Versatile:** Used in web development, data science, AI, and more
- **Popular:** High demand in the job market
- **Community:** Large, supportive community with extensive resources

Let's start your coding journey together!

Welcome to Python! 🚀

1. Your First Python Program

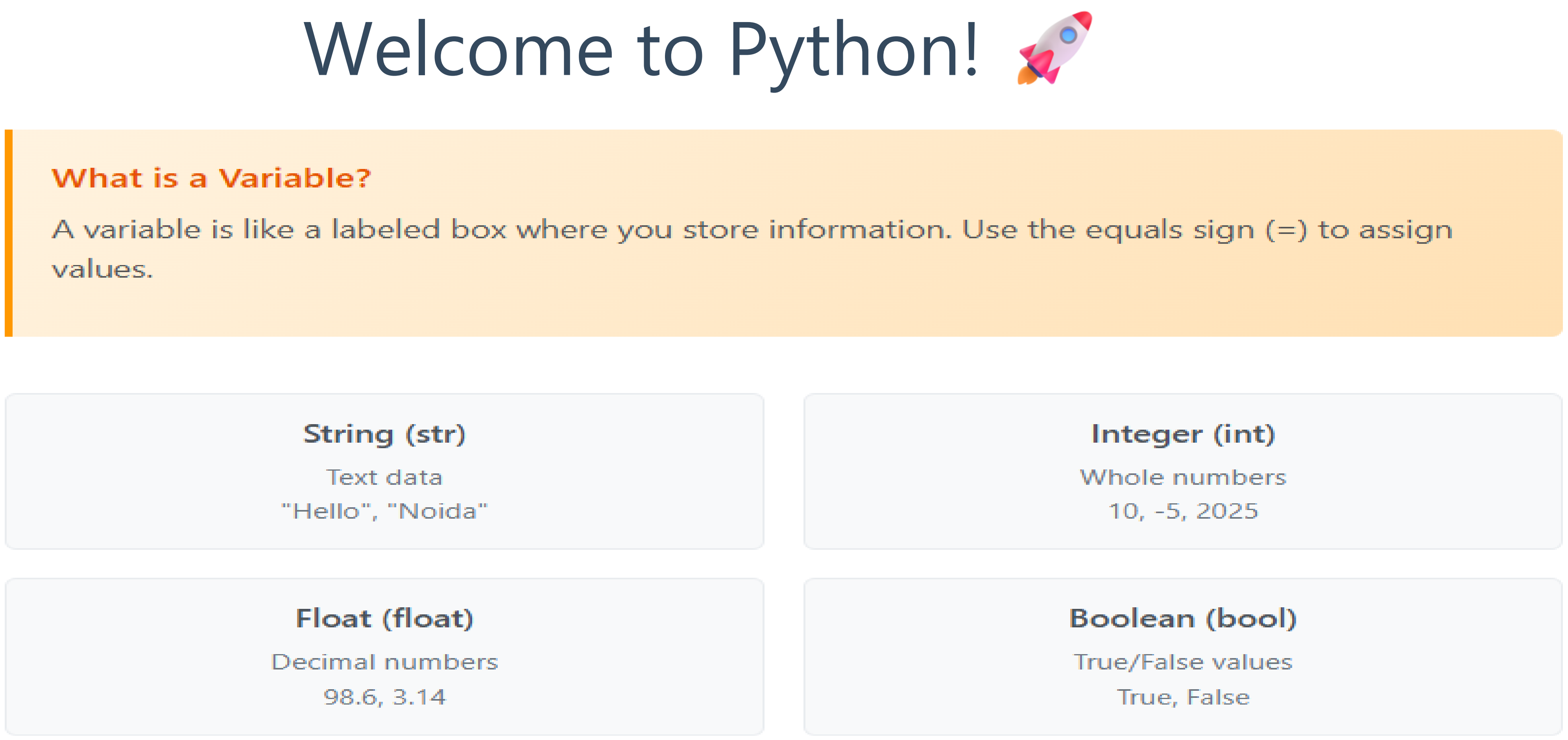
The print() Function

The `print()` function displays output to the screen. Text must be enclosed in single (') or double (") quotes.

```
print("Hello, Python World!")
print('This also works.')
```

✍️ Exercise 1

Write a Python program that prints your name on the first line and your favorite hobby on the second line.



Welcome to Python! 🚀

What is a Variable?

A variable is like a labeled box where you store information. Use the equals sign (=) to assign values.

String (str)
Text data
"Hello", "Noida"

Integer (int)
Whole numbers
10, -5, 2025

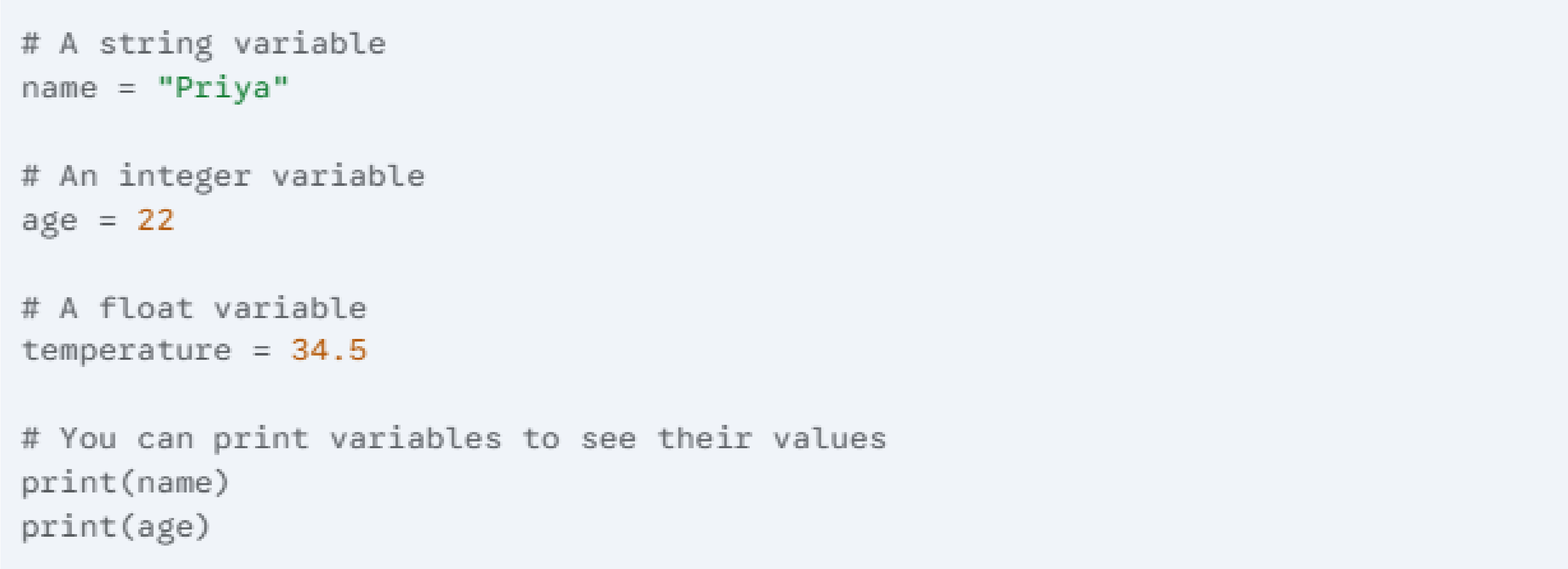
Float (float)
Decimal numbers
98.6, 3.14

Boolean (bool)
True/False values
True, False

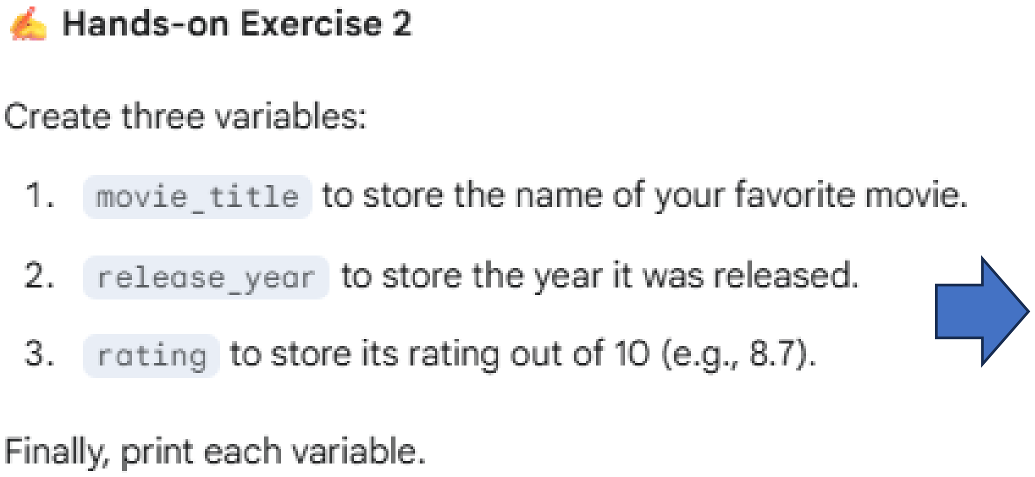
✍️ Exercise 2

Create three variables: movie_title (favorite movie), release_year (year released), rating (out of 10). Print each variable.

Welcome to Python! 🚀



```
# A string variable
name = "Priya"

# An integer variable
age = 22

# A float variable
temperature = 34.5

# You can print variables to see their values
print(name)
print(age)
```

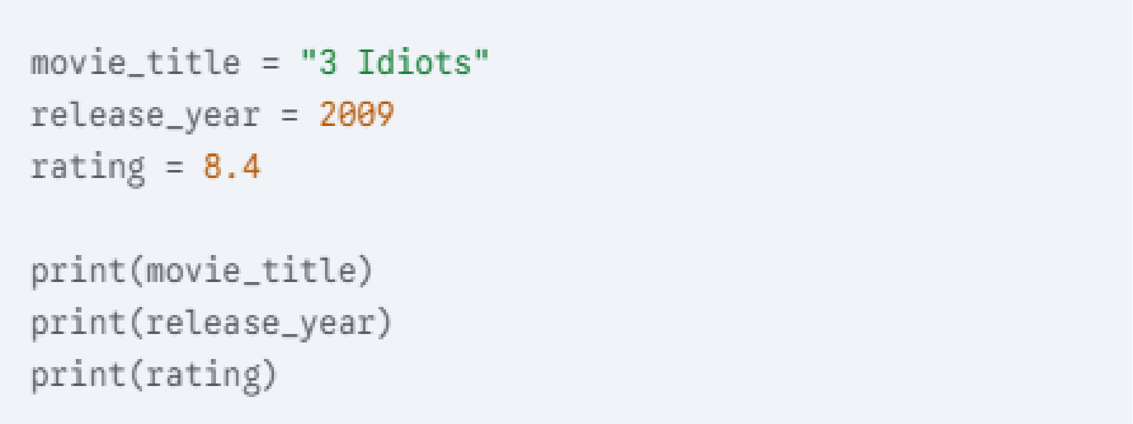


✍️ **Hands-on Exercise 2**

Create three variables:

1. `movie_title` to store the name of your favorite movie.
2. `release_year` to store the year it was released.
3. `rating` to store its rating out of 10 (e.g., 8.7).

Finally, print each variable.



```
movie_title = "3 Idiots"
release_year = 2009
rating = 8.4

print(movie_title)
print(release_year)
print(rating)
```

Welcome to Python! 🚀

3. Getting User Input

The input() Function

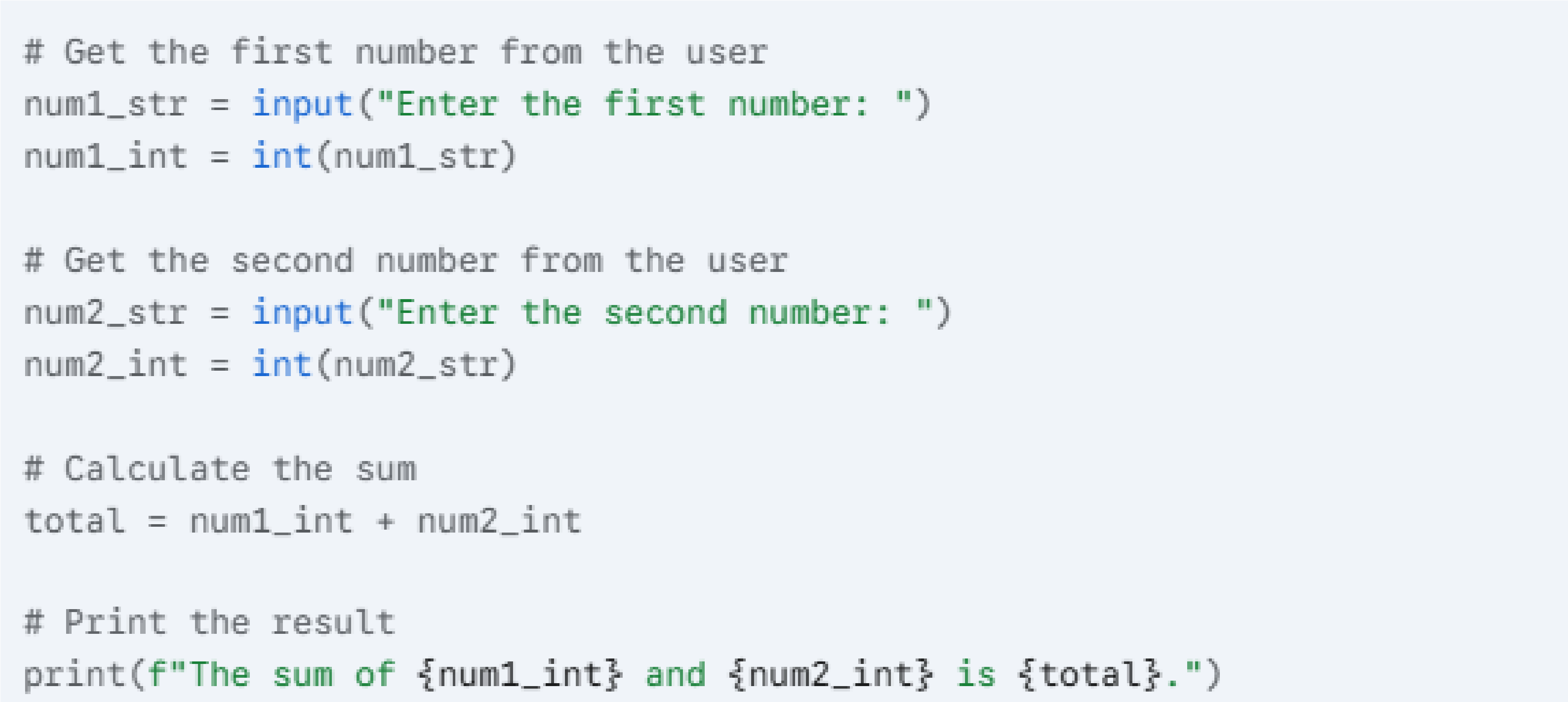
Makes your programs interactive! Always returns data as a `string`. Use `int()` or `float()` to convert to numbers.

```python
# Ask for the user's name
user_name = input("Please enter your name: ")

# Ask for the user's age and convert it to an integer
user_age_str = input("Please enter your age: ")
user_age_int = int(user_age_str)

# Using an f-string to easily combine strings and variables
print(f"Hello, {user_name}! You are {user_age_int} years old.")
```

User Input

✍️ Hands-on Exercise 3

Write a program that asks the user for two numbers, converts them to integers, adds them together, and prints the result.



```python
# Get the first number from the user
num1_str = input("Enter the first number: ")
num1_int = int(num1_str)

# Get the second number from the user
num2_str = input("Enter the second number: ")
num2_int = int(num2_str)

# Calculate the sum
total = num1_int + num2_int

# Print the result
print(f"The sum of {num1_int} and {num2_int} is {total}.")
```

4. Making Decisions with if-elif-else

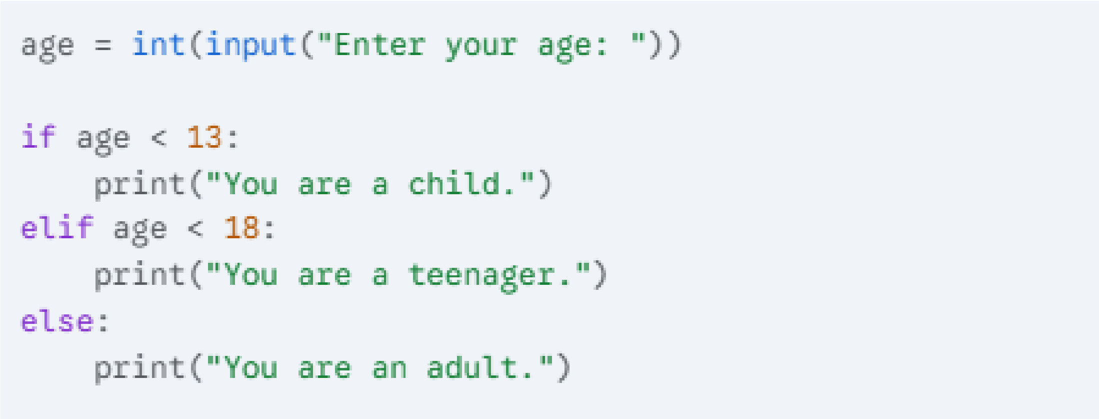
Conditional Logic

if: Runs code if condition is true

elif: Checks another condition

else: Runs if no conditions are true

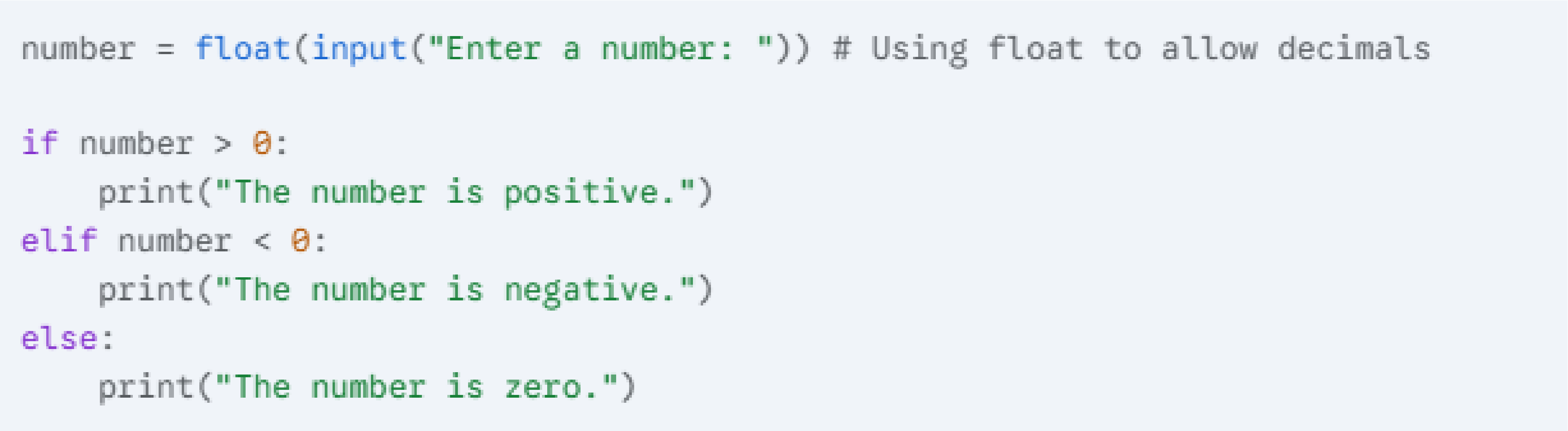
Comparison Operators: == (equal), != (not equal), > (greater), < (less), >= (greater/equal), <= (less/equal)



```
age = int(input("Enter your age: "))

if age < 13:
    print("You are a child.")
elif age < 18:
    print("You are a teenager.")
else:
    print("You are an adult.")
```

✍️ Hands-on Exercise 4

Write a program that asks for a number and then prints whether the number is **positive**, **negative**, or **zero**.



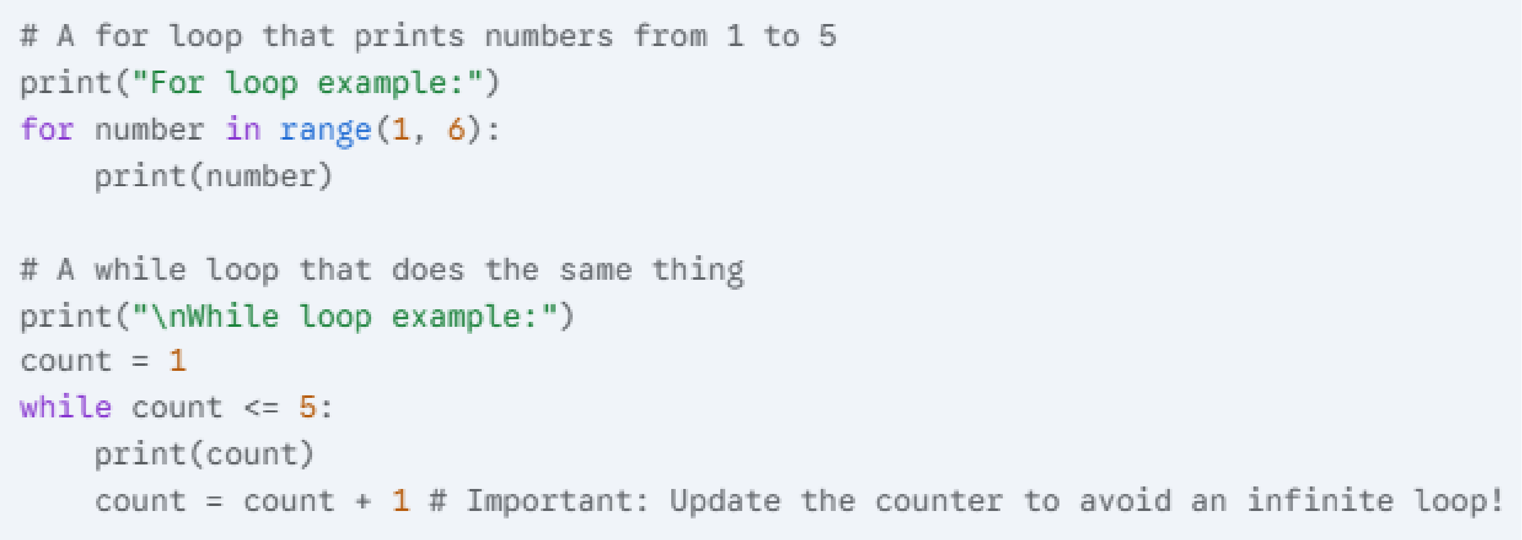
```
number = float(input("Enter a number: ")) # Using float to allow decimals

if number > 0:
    print("The number is positive.")
elif number < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```

5. Repeating Actions with Loops 🔄

Types of Loops

for loop: Iterate over sequences (lists, ranges)

while loop: Repeat while condition is true

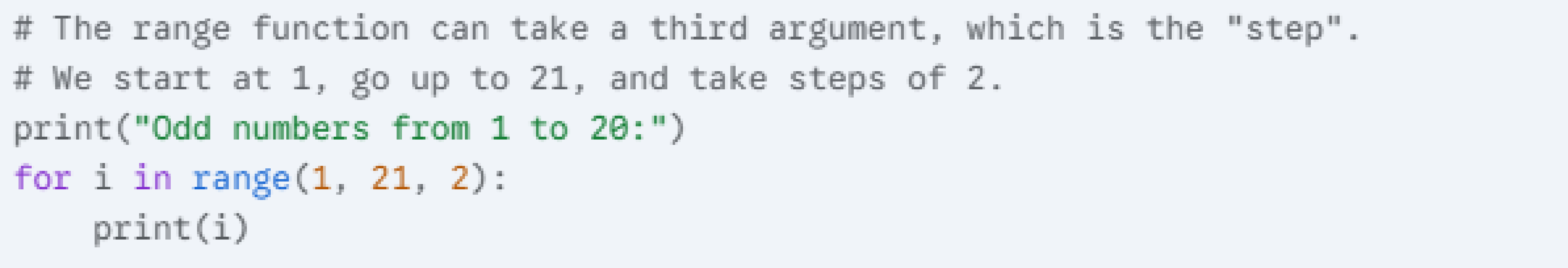


```python
# A for loop that prints numbers from 1 to 5
print("For loop example:")
for number in range(1, 6):
    print(number)

# A while loop that does the same thing
print("\nWhile loop example:")
count = 1
while count <= 5:
    print(count)
    count = count + 1 # Important: Update the counter to avoid an infinite loop!
```

5. Repeating Actions with Loops 🔄

✍️ Hands-on Exercise 5

Use a `for` loop and the `range()` function to print all the **odd numbers** from 1 to 20.

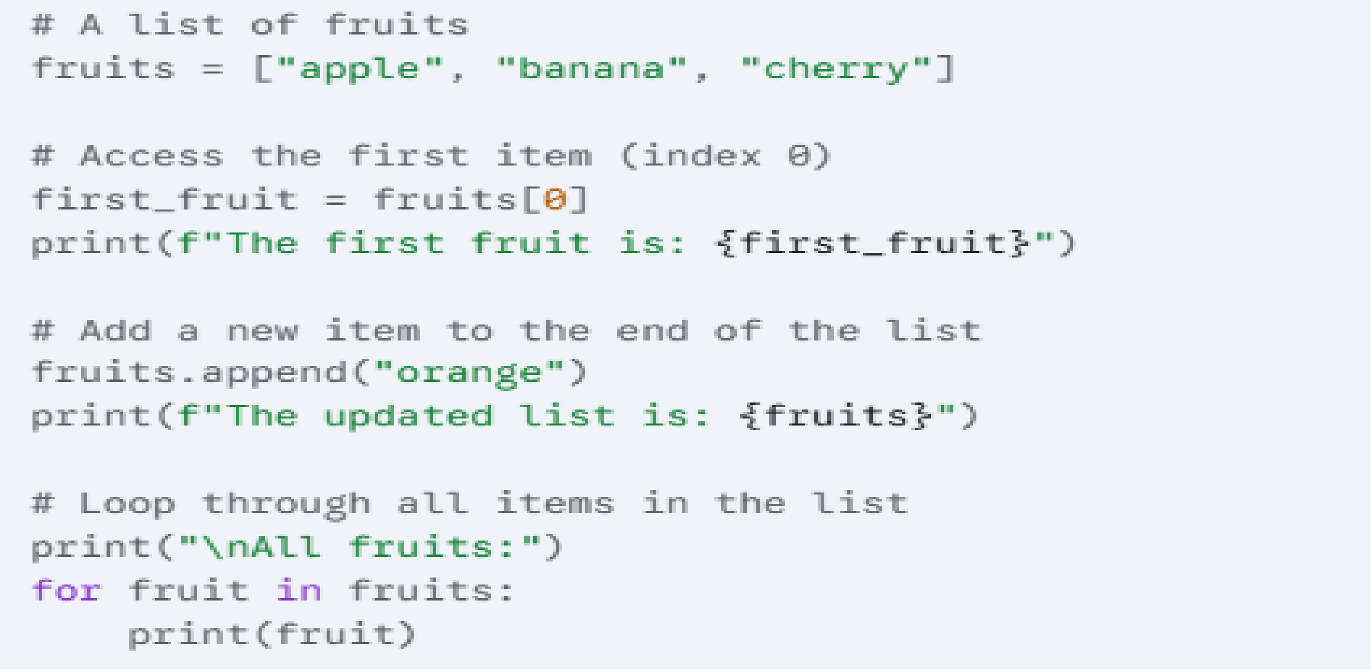


```python
# The range function can take a third argument, which is the "step".
# We start at 1, go up to 21, and take steps of 2.
print("Odd numbers from 1 to 20:")
for i in range(1, 21, 2):
    print(i)
```

6. Working with Lists 📝

What is a List?

An ordered collection of items. Created with square brackets []. **Important:** Indexing starts at 0!

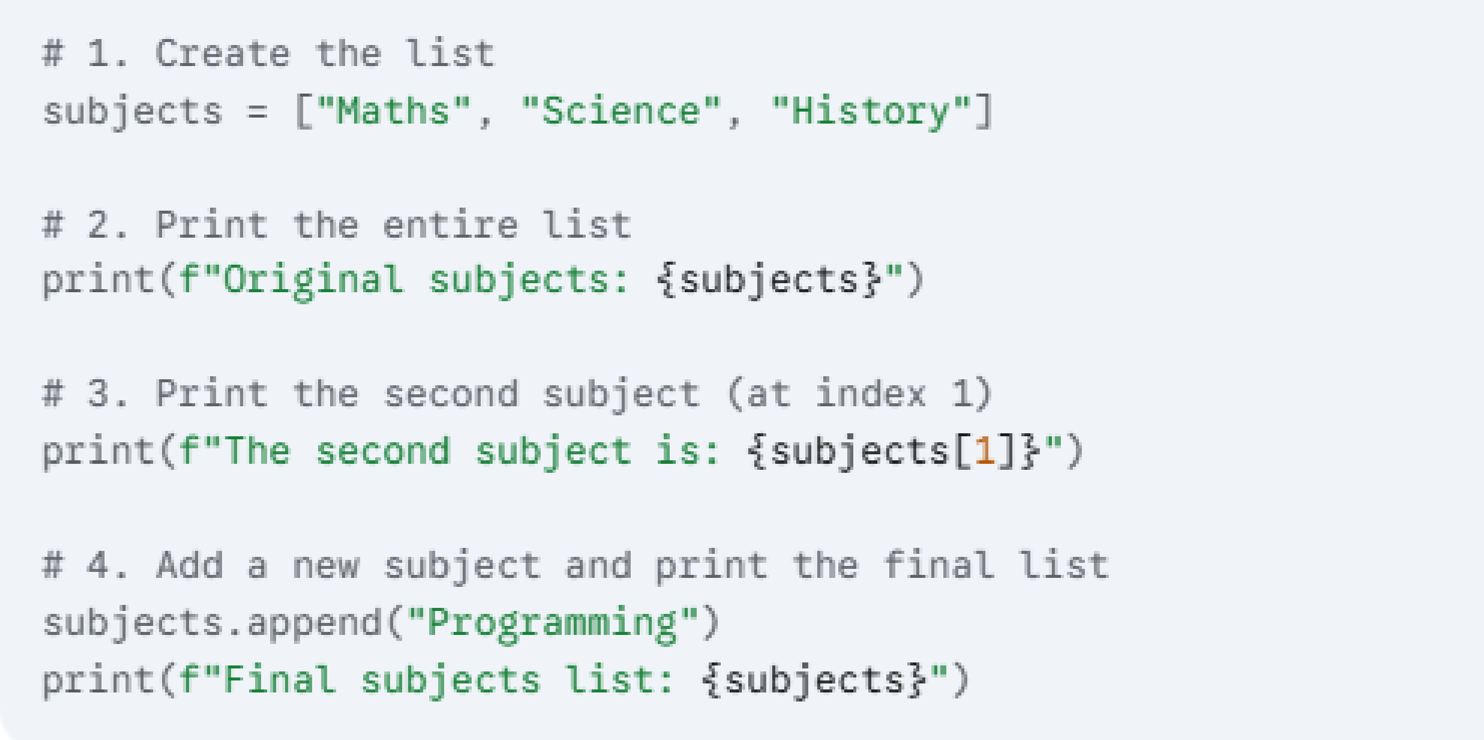


```
# A list of fruits
fruits = ["apple", "banana", "cherry"]

# Access the first item (index 0)
first_fruit = fruits[0]
print(f"The first fruit is: {first_fruit}")

# Add a new item to the end of the list
fruits.append("orange")
print(f"The updated list is: {fruits}")

# Loop through all items in the list
print("\nAll fruits:")
for fruit in fruits:
    print(fruit)
```

6. Working with Lists 📝

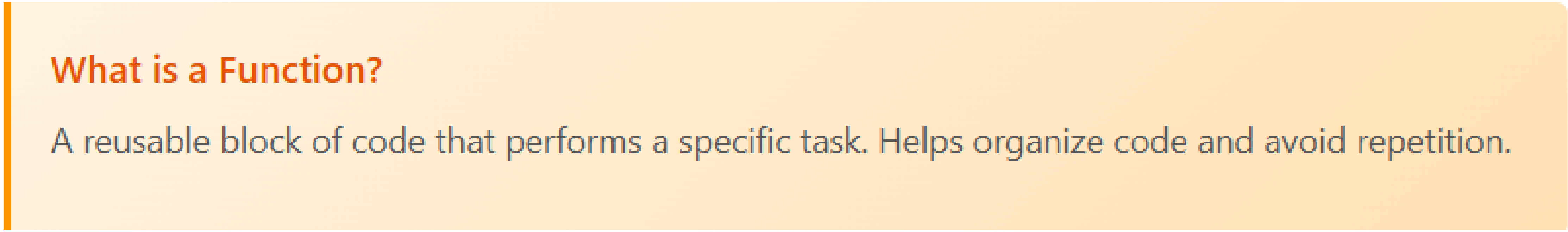
✍️ Exercise 4

Create a list of school subjects, print the list, access the second subject, and add "Programming" to the list.



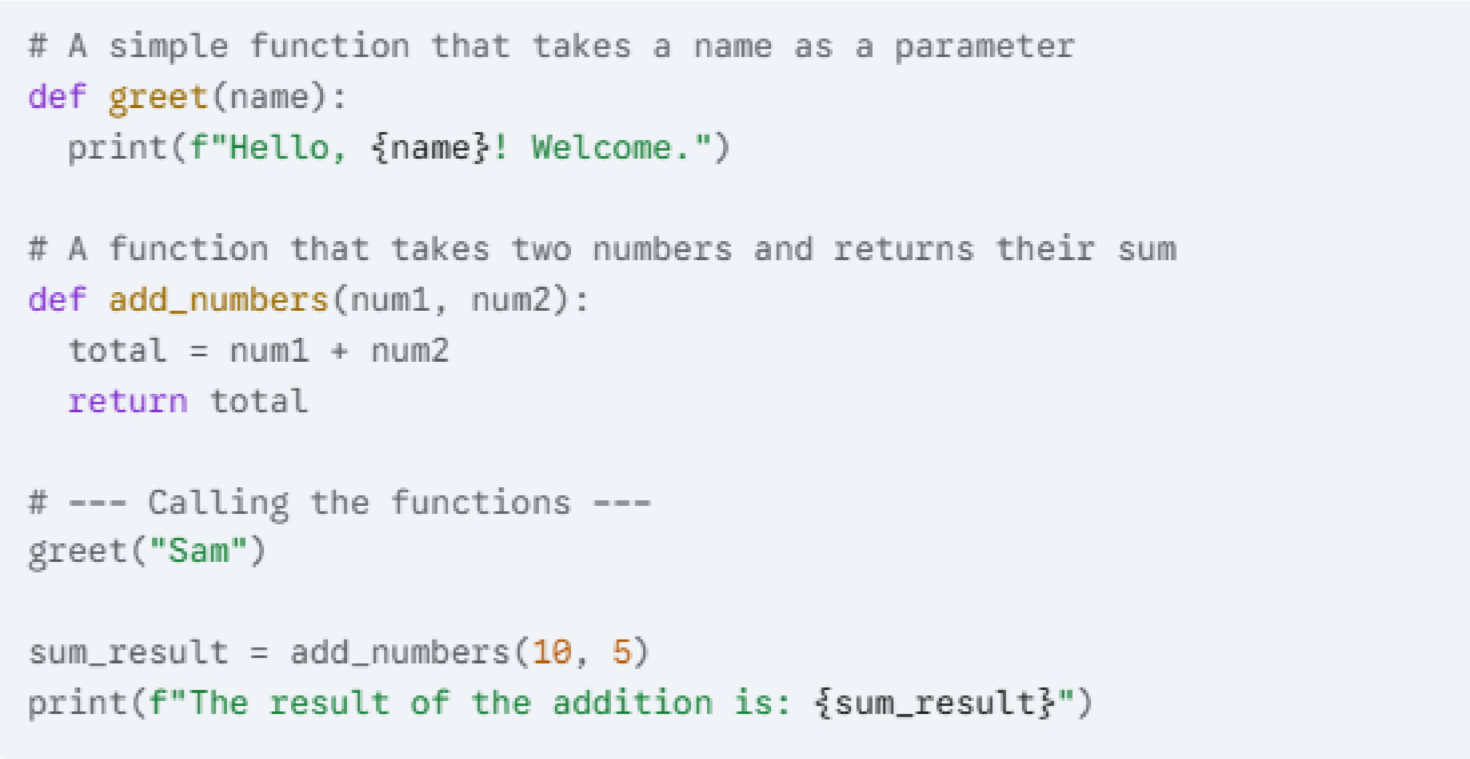
```python
# 1. Create the list
subjects = ["Maths", "Science", "History"]

# 2. Print the entire list
print(f"Original subjects: {subjects}")

# 3. Print the second subject (at index 1)
print(f"The second subject is: {subjects[1]}")

# 4. Add a new subject and print the final list
subjects.append("Programming")
print(f"Final subjects list: {subjects}")
```

7. Organizing Code with Functions



What is a Function?

A reusable block of code that performs a specific task. Helps organize code and avoid repetition.

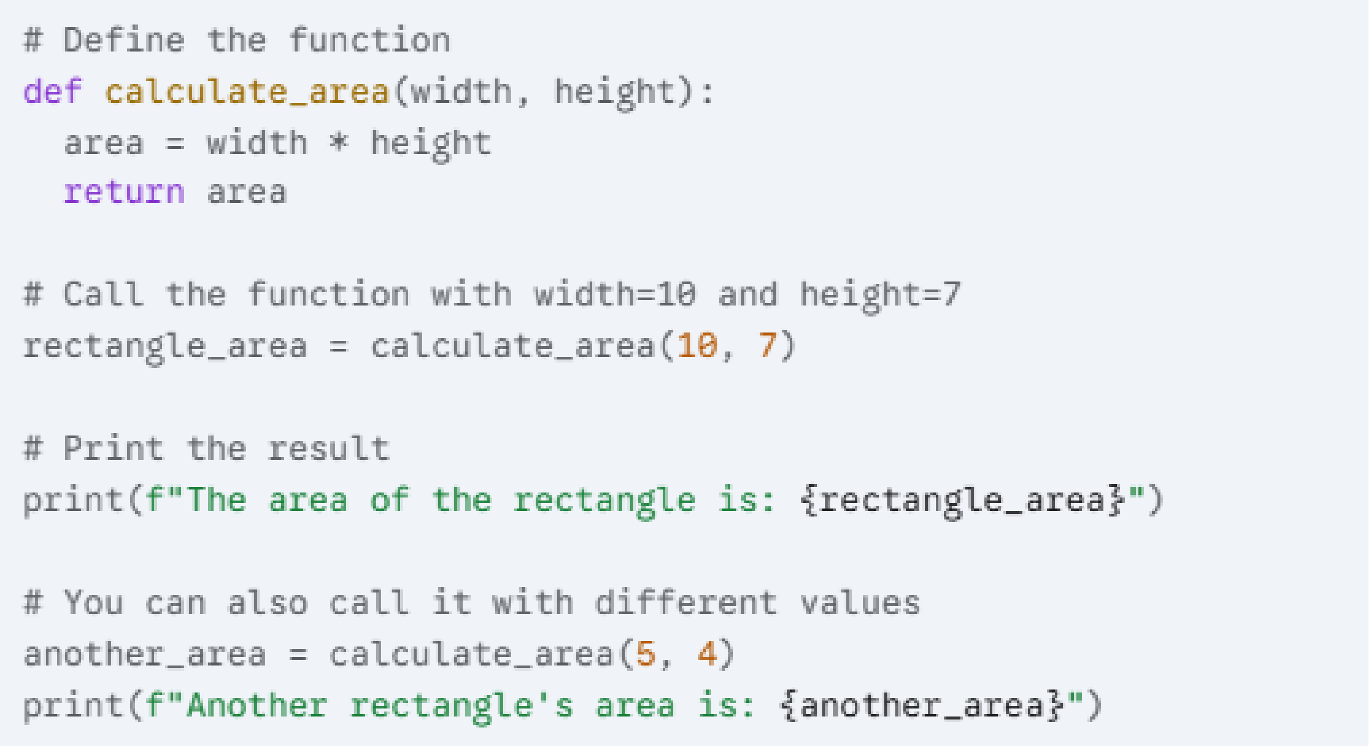


```python
# A simple function that takes a name as a parameter
def greet(name):
  print(f"Hello, {name}! Welcome.")

# A function that takes two numbers and returns their sum
def add_numbers(num1, num2):
  total = num1 + num2
  return total

# --- Calling the functions ---
greet("Sam")

sum_result = add_numbers(10, 5)
print(f"The result of the addition is: {sum_result}")
```

7. Organizing Code with Functions

✍️ Hands-on Exercise 7

Write a function named `calculate_area` that takes two parameters: `width` and `height`. The function should calculate the area of a rectangle (`width * height`) and `return` the result. Then, call this function with some numbers and print the returned area.



```
# Define the function
def calculate_area(width, height):
  area = width * height
  return area

# Call the function with width=10 and height=7
rectangle_area = calculate_area(10, 7)

# Print the result
print(f"The area of the rectangle is: {rectangle_area}")

# You can also call it with different values
another_area = calculate_area(5, 4)
print(f"Another rectangle's area is: {another_area}")
```

Congratulations! 🎉

You've learned the fundamental concepts of Python programming:

- Printing output with print()
- Variables and data types
- Getting user input
- Conditional statements
- Loops (for and while)
- Working with lists
- Creating functions

Keep practicing and you'll be building amazing things!