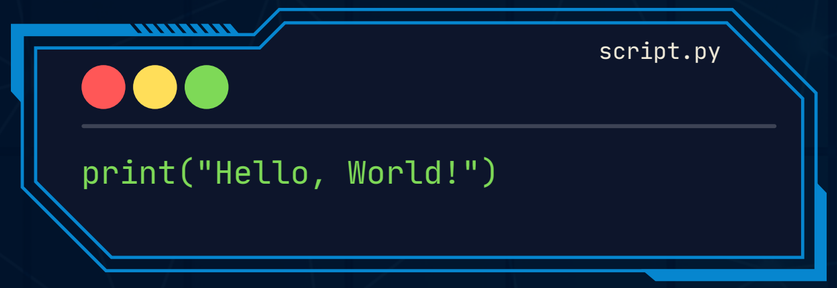
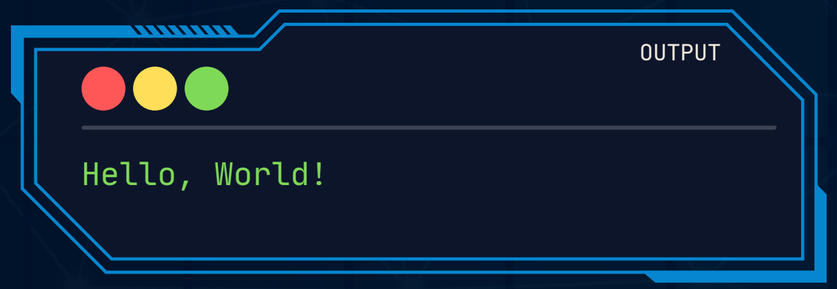
PYTHON FOUNDATIONS: FROM BASICS TO PRACTICE

SESSION 4: PLAYING WITH DATA AND FUNCTIONS

TABLE OF CONTENT

01 UNDERSTANDING FUNCTIONS





A function in Python is a reusable block of code with a specific purpose, and it can be given a name. When you call a function, you execute the code it contains. Functions allow you to input parameters to execute the same code with different values.

There are different types of functions in Python:

1. Built-in functions that come with Python.

2. User-defined functions that developers (like you!) create.

02 DEFINE AND USE A FUNCTION

```
def greet():
    print("Hello, world!")

greet() # Output: Hello, world!
```

In Python, a function is defined using the def keyword, followed by the function name and a pair of parentheses.

Functions may or may not take parameters (also called arguments) and can return values .

To call (use) a function without parameters, you simply use its name followed by parentheses.

FUNCTION PARAMETERS AND RETURN VALUES

Functions often need input data, which are represented by parameters defined within the parentheses when defining the function. You can have multiple parameters separated by commas. Functions can also return values using the `return` keyword, allowing data to be passed back to the calling code. Functions can have various combinations of parameters and return values or none at all, depending on their purpose.

```
def greet(name):
    print("Hello, " + name + "!")

greet("Alice") # Output: Hello, Alice!
```

```
def calculate_total(price, quantity):
    return price * quantity

calculate_total(200, 10)
```

04 SCOPE OF VARIABLES

```
a = 5

def f1():
  a = 2
  print(a)

print(a) # Prints 5
f1()   # Prints 2
```

Variables defined inside a function are called local variables. They can only be accessed within the scope of that function and are not visible outside of it.

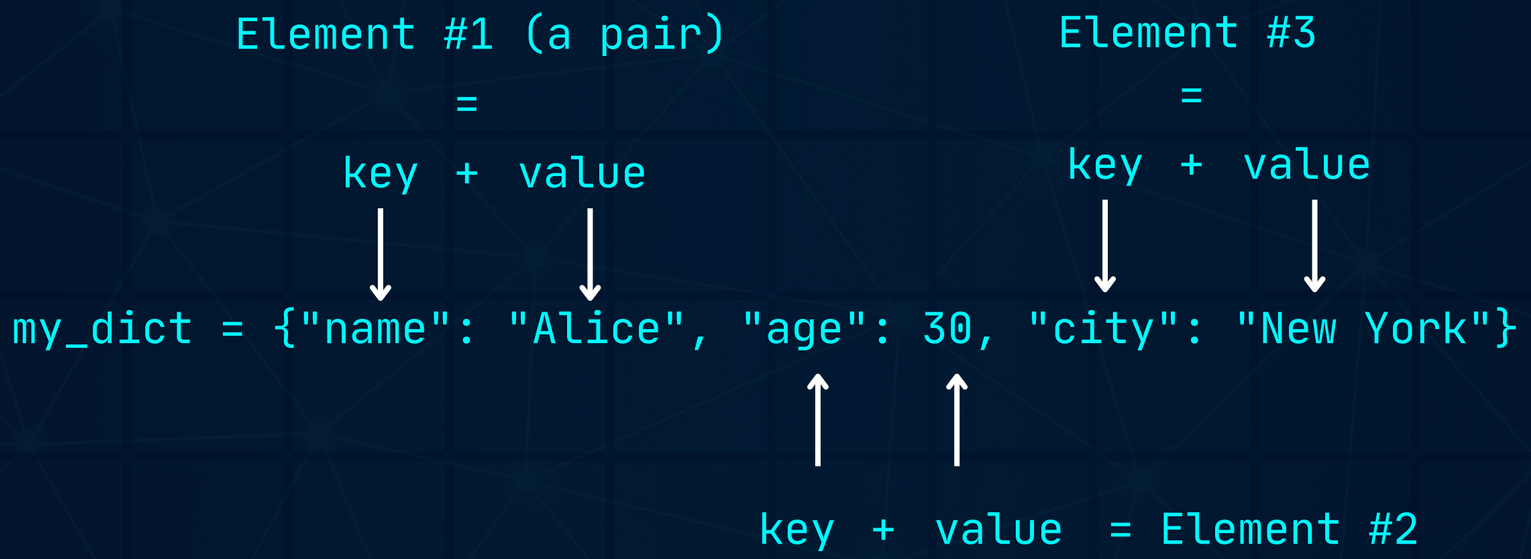
Global variables, on the other hand, are defined outside of functions and can be accessed anywhere in the code.

05 BEST PRACTICES

1. Use meaningful names for your functions and parameters to make your code self-explanatory.
2. Functions should have a single, well-defined purpose.
3. Aim for functions to be concise and not overly long.
4. Comment your functions to explain their purpose, expected inputs, and returned values.

06 DICTIONARIES

In Python, a dictionary is a powerful data structure used to store and manage collections of data. Unlike other data structures, such as lists, dictionaries organize data into key-value pairs. Each key in a dictionary is unique and associated with a specific value.



HOMEWORK ASSIGNMENT: AVERAGE GRADES CALCULATION WITH FUNCTIONS AND DICTIONARY

In our previous session, we laid the foundation for calculating a student's average grades in math, physics, and science and provided encouraging messages based on their performance. Now, it's time to take your coding skills to the next level by implementing functions and dictionaries in your program.

Your Task:

1. **Write a Python program that calculates the average grades of a student in math, physics, and science using functions and dictionaries.**
2. **Create two functions:**
   - **Function 1: calculate_average(grades) that takes a dictionary of grades as input, calculates the average, and returns it.**
   - **Function 2: evaluate_student(average) that takes the calculated average as input and provides an encouraging message based on the average.**
3. **Create a dictionary grades to store the student's grades in math, physics, and science.**
4. **Calculate the average grade using the calculate_average function.**
5. **Evaluate and encourage the student using the evaluate_student function.**
6. **Display the following information:**
   - **Individual grades in math, physics, and science.**
   - **Average grade.**
   - **The encouraging message based on the average.**

HAPPY CODING AHEAD!

Every line of code you write is a step towards mastering Python. Keep practicing, keep exploring, and keep pushing your boundaries.

Thank you for joining us on this journey!

Sarah Haddad
haddads686@gmail.com