Script Programming 2

Functions

Merxhan Bajrami

2023 March 23
Week 3

Contents

Functions

Functions in Python are blocks of code that are designed to perform a specific task. They are created using the def keyword followed by the function name, parentheses, and a colon. The code to be executed is then indented under the function declaration.

def Statements with Parameters:

Parameters are variables that are passed into a function when it is called. They are listed inside the parentheses after the function name, separated by commas. The values of these variables can be used within the function to perform the desired task.

Here is an example of a function that takes two parameters and returns their sum:

```
def add_numbers(x, y):
    result = x + y
    return result
```

Return Values and return Statements:

Functions can also return a value when they are called. This is done using the return statement. The value to be returned is placed after the return keyword. If a function is called without a return statement, it will return None by default.

Here is an example of a function that returns the square of a number:

```
def square_number(x):
    return x**2
```

The None Value

The None value is a special object in Python that represents the absence of a value. It is often used as a default return value for functions that do not return anything else.
Here is an example of a function that does not return anything:

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

Keyword Arguments and print()

Functions can also be called using keyword arguments. These are arguments that are preceded by their parameter name, separated by an equal sign. This can make it easier to understand what each argument represents.

Here is an example of a function that takes a name and age as keyword arguments

```
def introduce(name, age):
    print("My name is " + name + " and I am " + str(age) + " years old.")

# Call the function using keyword arguments
introduce(name="Arben", age=25)
```

Local Scope

A variable defined inside a function has local scope. It can only be accessed within the function where it was defined. Once the function finishes execution, the variable is destroyed and cannot be accessed anymore.

```
def my_function():
  x = 10  # x has local scope
  print("Inside the function, x = ", x)

my_function()
print("Outside the function, x = ", x)  # This will raise an error
```

```
Inside the function, x = 10
NameError: name 'x' is not defined
```

Global Scope

A variable defined outside a function has global scope. It can be accessed from anywhere in the program, including inside functions. To modify a global variable inside a function, you need to use the global keyword.

```
x = 10  # x has global scope

def my_function():
  global x
  x = 20  # Modify the global variable x
  print("Inside the function, x = ", x)

my_function()
print("Outside the function, x = ", x)
```

```
Inside the function, x = 20
Outside the function, x = 20
```

global statement

The global keyword in Python is used to indicate that a variable is a global variable, which means it can be accessed from anywhere in the program, including inside functions.

```
global x = 10  # x has global scope

def my_function():
  global x
  x = 20  # Modify the global variable x
  print("Inside the function, x = ", x)

my_function()
print("Outside the function, x = ", x)
```

```
Inside the function, x = 20
Outside the function, x = 20
```

Exception handling

In Python, exceptions are errors that occur during the execution of a program. Exception handling is a way to handle these errors and prevent the program from crashing. Python provides a try-except block for handling exceptions.

```
try:
  x = int(input("Enter a number: "))
  print("The number is ", x)
except ValueError:
  print("Invalid input. Please enter a valid number.")
```

```
Enter a number: hello
Invalid input. Please enter a valid number.
```

Streamlit

- Streamlit is an open-source Python library that makes it easy to build beautiful custom web apps for machine learning and data science.
- Streamlit is designed to be easy to use, allowing developers to create and share web apps with just a few lines of Python code.

Streamlit features

- Streamlit offers a variety of features to help developers build powerful web apps, including:
- Automatic widget generation
- Customizable layouts
- Real-time collaboration
- Built-in caching
- Integration with popular machine learning libraries

Getting started

- To get started with Streamlit, you need to install the library using pip: pip install streamlit
- Once installed, you can create a new Streamlit app by creating a new Python file and importing the Streamlit library.
- Then, you can use Streamlit's built-in widgets to create interactive elements in your app.

Why Streamlit?

- Streamlit is a powerful tool for building custom web apps in Python.
- With its easy-to-use widgets and customizable layouts, Streamlit makes it easy for developers to create and share their work.
- Whether you're building machine learning models or exploring data, Streamlit is a great tool to have in your toolbox.

Exercices

Problem 1

1. Write a function called calculate_sum that takes two arguments, x and y, and returns their sum.

Problem 2

1. Write a function called convert_to_upper that takes a string as input and returns the string in all uppercase letters.

Problem 3

1. Define a function called get_average that takes a list of numbers as input and returns their average.

Problem 4

- Create a function called print_table that takes a keyword argument header and a list of rows, and prints a table with the header and rows.

Problem 5

- Write a function called calculate_total that takes a list of numbers and returns their sum. Use a global variable called tax_rate to calculate the total with tax.

Problem 6

- Define a function called get_factorial that takes an integer as input and returns its factorial. Use exception handling to handle invalid input.

Problem 7

- Create a function called greet_many that takes a list of names and a keyword argument greeting, and returns a list of greeting messages for each name.

Problem 8

- Write a function called find_max that takes a list of numbers as input and returns the maximum value. Use the built-in function max() to find the maximum value, and return None if the list is empty.

Script Programming 2

Questions?

Merxhan Bajrami

2023 March 23
Week 3