

ASSIGNMENT 4 – PYTHON FUNCTIONS

1. What does the len() function do in Python?

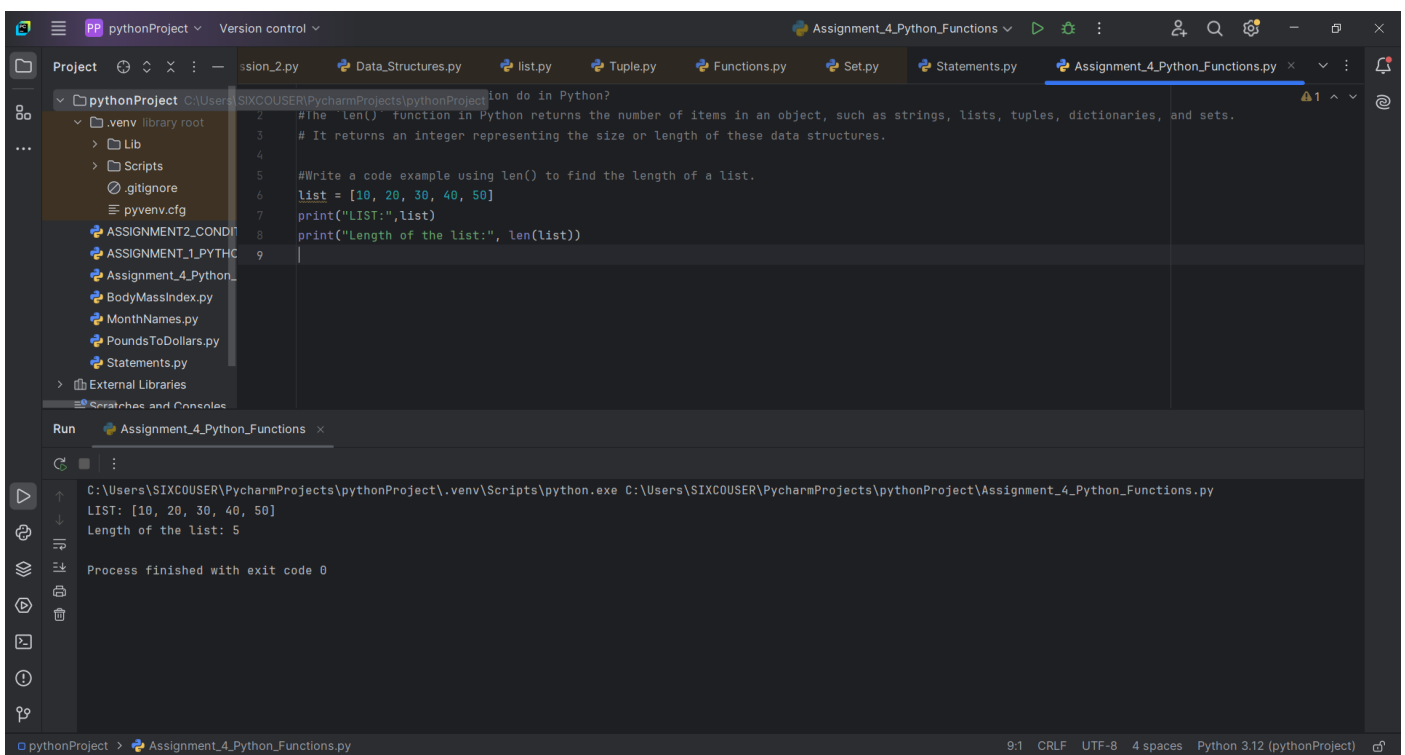
The `len()` function in Python returns the number of items in an object, such as strings, lists, tuples, dictionaries, and sets. It returns an integer representing the size or length of these data structures.

2. Write a code example using len() to find the length of a list.

```
list = [10, 20, 30, 40, 50]

print("LIST:",list)

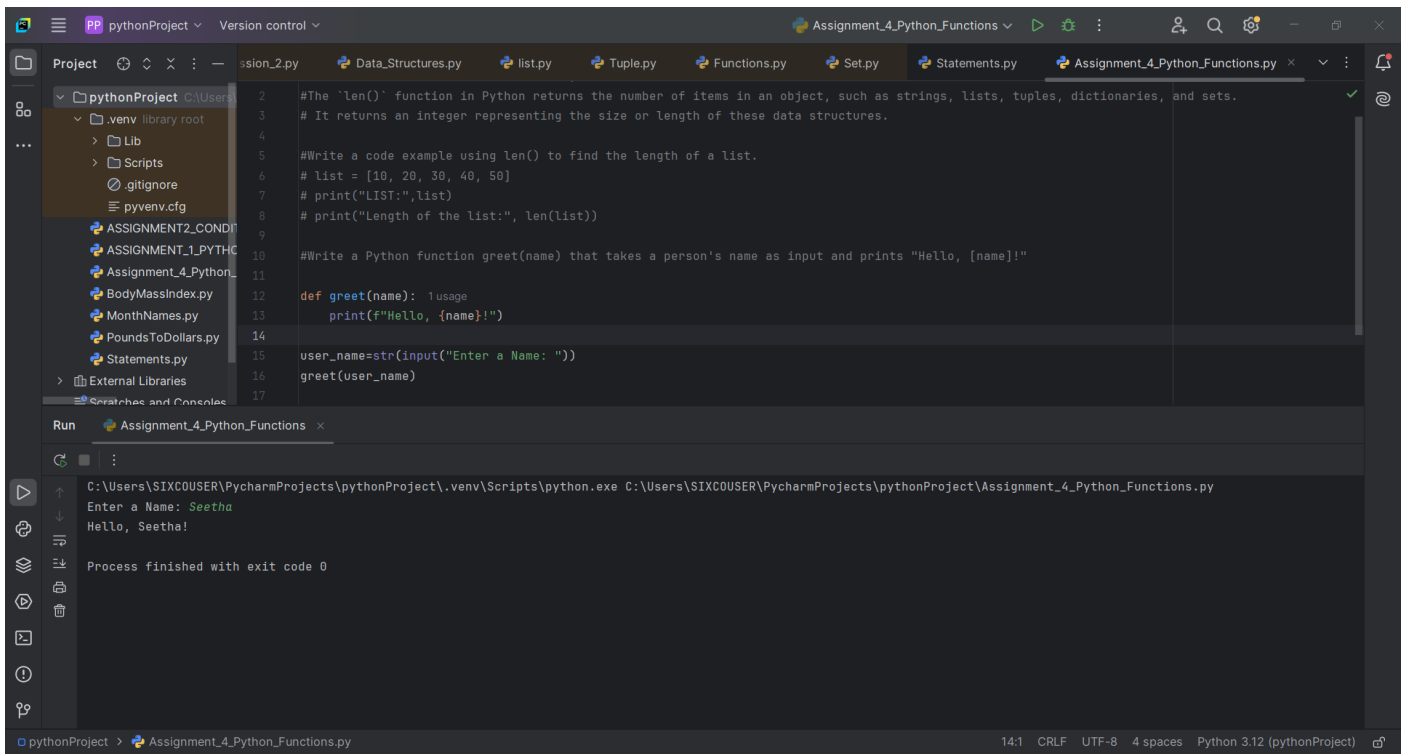
print("Length of the list:", len(list))
```



3. Write a Python function greet(name) that takes a person's name as input and prints "Hello, [name]!"

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

user_name=str(input("Enter a Name: "))
greet(user_name)
```



4. Write a Python function `find_maximum(numbers)` that takes a list of integers and returns the maximum value without using the built-in `max()` function. Use a loop to iterate through the list and compare values.

```
def find_maximum(numbers):  
    if not numbers:  
        return None  
    maximum = numbers[0]  
    for i in numbers:  
        if i > maximum:  
            maximum = i  
  
    return maximum
```

```
numbers_list = [14, 29, 51, 8, 30]  
print(numbers_list)  
print("Maximum value:", find_maximum(numbers_list))
```

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Assignment_4_Python_Functions.py'. The main editor window displays the code for 'Assignment_4_Python_Functions.py'. The code defines a function 'find_maximum' that takes a list of numbers and returns the maximum value. It then calls this function with a list of numbers [14, 29, 51, 8, 30] and prints the result. The Run window at the bottom shows the output: '[14, 29, 51, 8, 30]' and 'Maximum value: 51'.

```
22 def find_maximum(numbers):
23     if not numbers:
24         return None
25     maximum = numbers[0]
26     for i in numbers: # Start iteration from the second element
27         if i > maximum:
28             maximum = i
29
30     return maximum
31
32 numbers_list = [14, 29, 51, 8, 30]
33 print(numbers_list)
34 print("Maximum value:", find_maximum(numbers_list))
35
36
```

Run: Assignment_4_Python_Functions.py

C:\Users\SIXCOUSER\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\SIXCOUSER\PycharmProjects\pythonProject\Assignment_4_Python_Functions.py

[14, 29, 51, 8, 30]
Maximum value: 51

Process finished with exit code 0

5. Explain the difference between local and global variables in a Python function.

Local Variables

#Scope: Defined inside a function; accessible only within that function.

#Lifetime: Exists only during the function's execution.

#Example:

```
def my_function():
```

```
    local_var = 10          # Local variable
```

```
    print(local_var)
```

```
my_function()              # This will print 10
```

```
print(local_var)           # This will raise an error, local_var is not accessible here
```

The screenshot shows the same PyCharm IDE environment as the first screenshot, but with additional comments added to the code to explain local variables. The code is the same as before, but with comments explaining the scope and lifetime of the local variable 'local_var'. The Run window at the bottom shows the output: '10'.

```
32 # numbers_list = [14, 29, 51, 8, 30]
33 # print(numbers_list)
34 # print("Maximum value:", find_maximum(numbers_list))
35
36 #Explain the difference between local and global variables in a Python function.
37 #Local Variables
38 #Scope: Defined inside a function; accessible only within that function.
39 #Lifetime: Exists only during the function's execution.
40 #Example:
41 def my_function():
42     local_var = 10 # Local variable
43     print(local_var)
44
45 my_function() # This will print 10
46 # print(local_var) # This will raise an error, local_var is not accessible here
47
```

Run: Assignment_4_Python_Functions.py

C:\Users\SIXCOUSER\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\SIXCOUSER\PycharmProjects\pythonProject\Assignment_4_Python_Functions.py

10

Process finished with exit code 0

Global Variables

#Scope: Defined outside any function; accessible throughout the entire program.

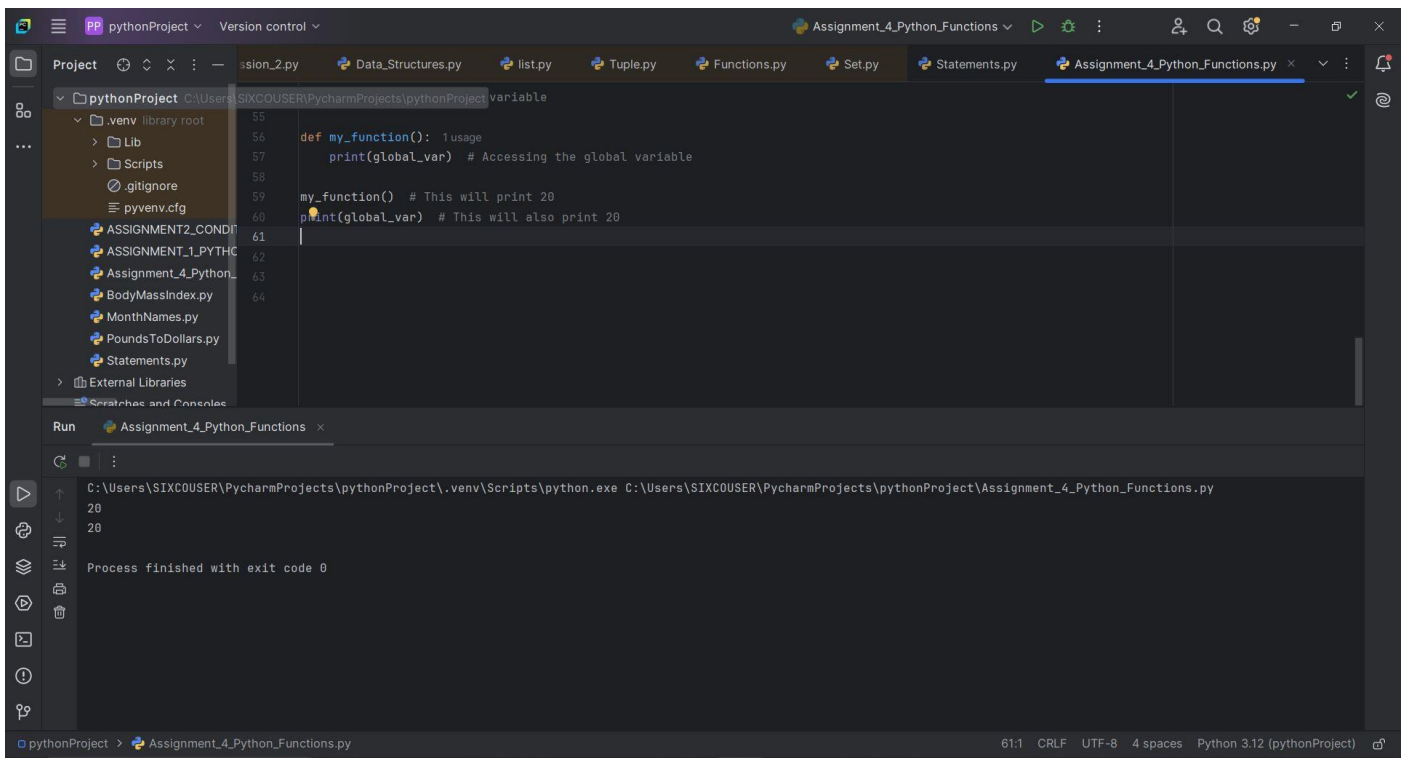
#Lifetime: Exists for the duration of the program.

#Example:

```
global_var = 20          # Global variable
```

```
def my_function():  
    print(global_var)    # Accessing the global variable
```

```
my_function()           # This will print 20  
print(global_var)       # This will also print 20
```



6. Write a program where a global variable and a local variable have the same name and show how Python differentiates between them.

```
value = 10    # Global variable
```

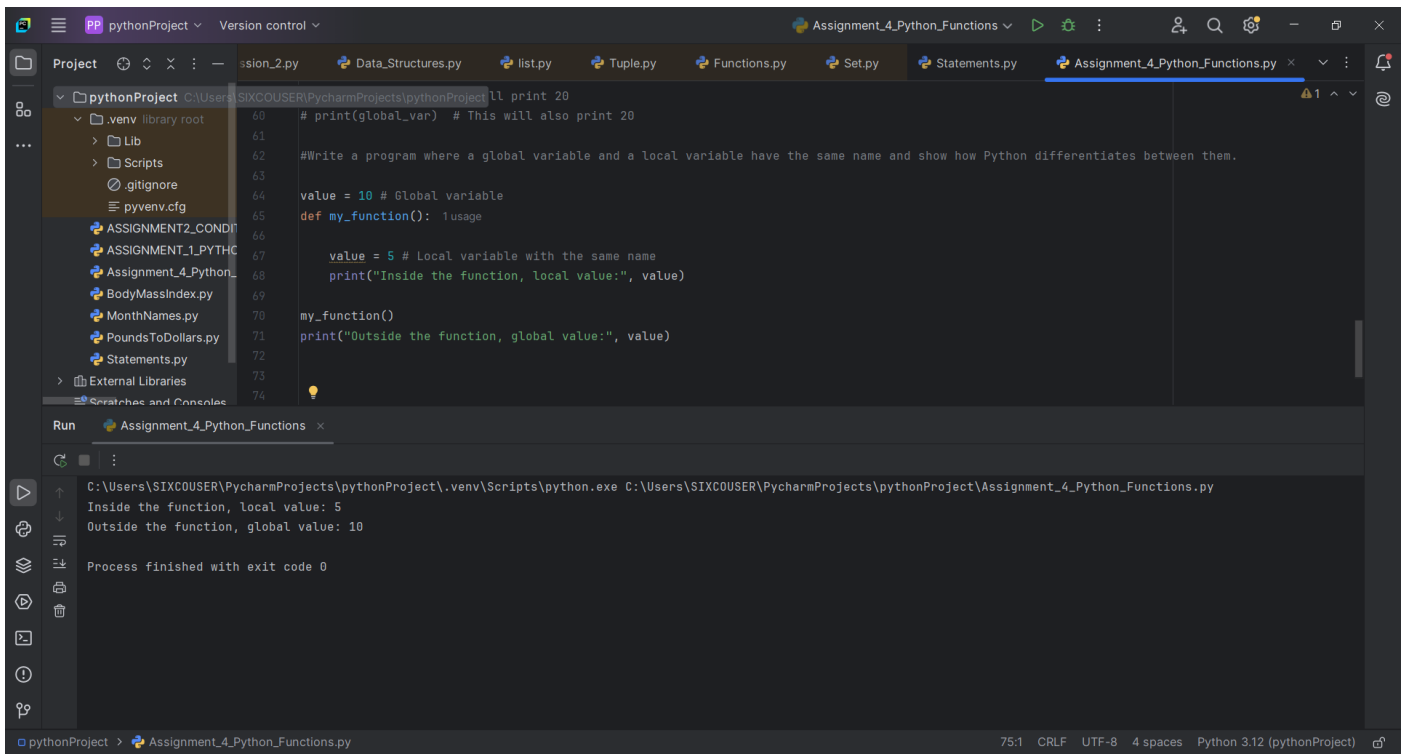
```
def my_function():
```

```
    value = 5    # Local variable with the same name
```

```
    print("Inside the function, local value:", value)
```

```
my_function()
```

```
print("Outside the function, global value:", value)
```



7. Create a function `calculate_area(length, width=5)` that calculates the area of a rectangle. If only the length is provided, the function should assume the width is 5. Show how the function behaves when called with and without the width argument.

```
def calculate_area(length, width=5):
```

```
    area = length * width
```

```
    return area
```

```
area_with_width = calculate_area(10, 3)
```

```
print("Area with length and width:", area_with_width)
```

```
area_with_default_width = calculate_area(10)
```

```
print("Area with length and default width 5:", area_with_default_width)
```

pythonProject

Version control

Assignment_4_Python_Functions

pythonProject

ssion_2.py

Data_Structures.py

list.py

Tuple.py

Functions.py

Set.py

Statements.py

Assignment_4_Python_Functions.py

pythonProject

.venv

library root

Lib

Scripts

.gitignore

pyvenv.cfg

ASSIGNMENT2_CONDITIONAL.py

ASSIGNMENT1_PYTHON_FUNCTIONS.py

Assignment_4_Python_Functions.py

BodyMassIndex.py

MonthNames.py

PoundsToDollars.py

Statements.py

External Libraries

Scratches and Consoles

75

Show how the function behaves when called with and without the width argument.

76

77

def calculate_area(length, width=5): 2 usages

78

area = length * width

79

return area

80

81

area_with_width = calculate_area(length=10, width=3)

82

print("Area with length and width:", area_with_width)

83

84

area_with_default_width = calculate_area(10)

85

print("Area with length and default width 5:", area_with_default_width)

86

87

88

89

90

Run

Assignment_4_Python_Functions

C:\Users\SIXCOUSER\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\SIXCOUSER\PycharmProjects\pythonProject\Assignment_4_Python_Functions.py

Area with length and width: 30

Area with length and default width 5: 50

Process finished with exit code 0

pythonProject

Assignment_4_Python_Functions.py

89:1

CRLF

UTF-8

4 spaces

Python 3.12 (pythonProject)