

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and running code. The file explorer on the left shows a project named 'entri\_d41\_python\_project' with several files: 'Session1.py', 'Tuple.py', 'Statement.py', 'functions.py', 'filehandling.py', and 'Assignment4.py'. The 'Assignment4.py' file is open in the editor, showing a Python script with comments and code. The script includes a comment about the 'len()' function, a code example for a string, a code example for a list, and a function definition 'greet()'. The 'Run' toolbar at the bottom left shows a play button and a terminal icon. The terminal window at the bottom displays the command used to run the script: 'C:\Users\user\PycharmProjects\entri\_d41\_python\_project\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\entri\_d41\_pytho'. The output of the script is shown in the terminal, with the number '5' on the first line and '4' on the second line.

```
# 1 Python-Functions What does the len() function do in Python?
# The len() function is a built in function returns the number of items in an object
# (strings,list,tuples,sets,dictionaries)
# When the object is a string, the len() function returns the number of characters in the string.
a='hello'
print(len(a))
# Write a code example using len() to find the length of a list.
list=[32,'hi',True,2]
print(len(list))

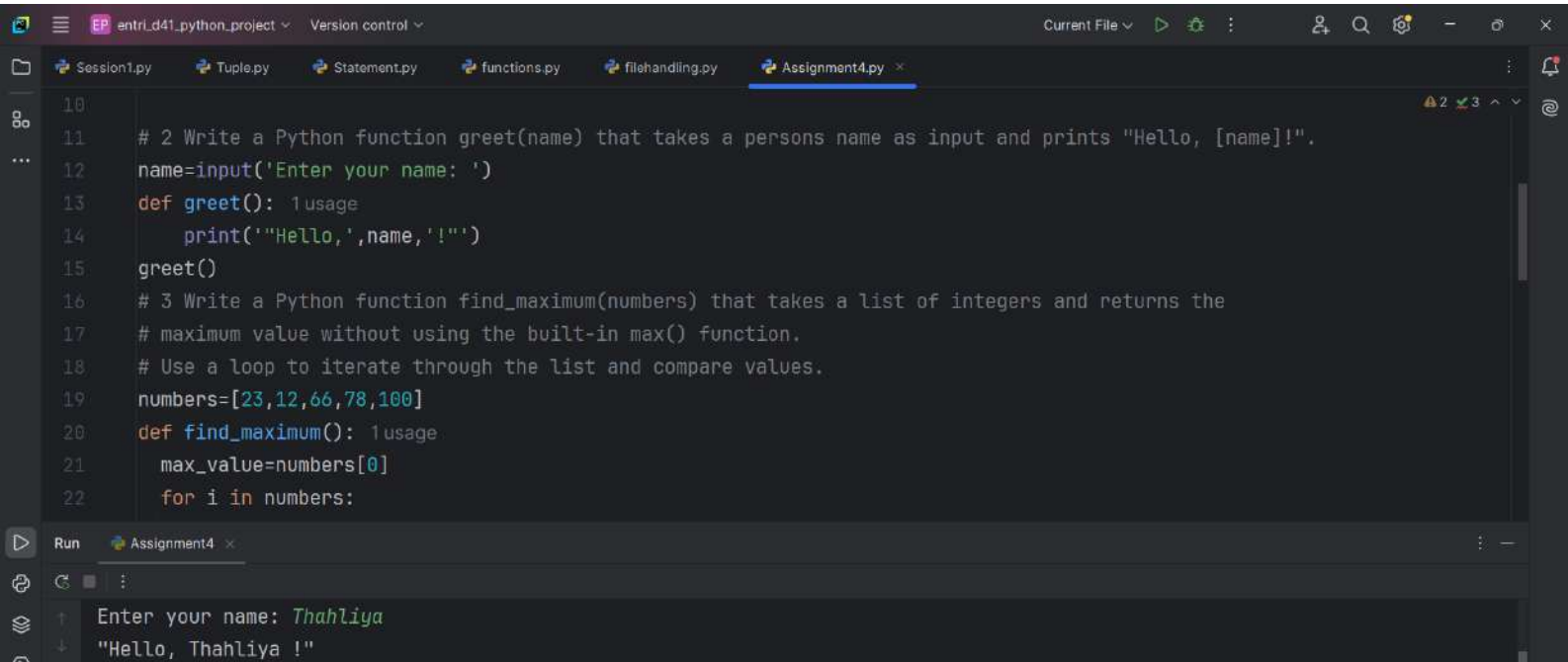
# 2 Write a Python function greet(name) that takes a persons name as input and prints "Hello, [name]!".
name=input('Enter your name: ')
def greet(): 1 usage
```

Run Assignment4 x

C:\Users\user\PycharmProjects\entri\_d41\_python\_project\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\entri\_d41\_pytho

5

4



```
10
11 # 2 Write a Python function greet(name) that takes a persons name as input and prints "Hello, [name]!".
12 name=input('Enter your name: ')
13 def greet(): 1 usage
14     print('Hello,',name, '!')
15 greet()
16 # 3 Write a Python function find_maximum(numbers) that takes a list of integers and returns the
17 # maximum value without using the built-in max() function.
18 # Use a loop to iterate through the list and compare values.
19 numbers=[23,12,66,78,100]
20 def find_maximum(): 1 usage
21     max_value=numbers[0]
22     for i in numbers:
```

Run Assignment4 x

Enter your name: *Thahliya*

"Hello, Thahliya !"

The screenshot shows a code editor window with a dark theme. The top bar includes a file explorer icon, a menu icon, and the project name 'entri\_d41\_python\_project'. Below the top bar is a tab bar with several open files: 'Session1.py', 'Tuple.py', 'Statement.py', 'functions.py', 'filehandling.py', and 'Assignment4.py'. The 'Assignment4.py' tab is selected. The main editor area displays Python code for a function 'find\_maximum' that iterates through a list of numbers to find the maximum value. The code is as follows:

```
16 # 3 Write a Python function find_maximum(numbers) that takes a list of integers and returns the
17 # maximum value without using the built-in max() function.
18 # Use a loop to iterate through the list and compare values.
19 numbers=[23,12,66,78,100]
20 def find_maximum():
21     max_value=numbers[0]
22     for i in numbers:
23         if i > max_value:
24             max_value=i
25     print('Maximum value of the list is',max_value)
26 find_maximum()
27
28 # 4 Explain the difference between local and global variables in a Python function.
```

At the bottom of the editor is a 'Run' panel with a sub-tab 'Assignment4'. It shows the output of the code execution: 'Maximum value of the list is 100'.

The screenshot shows a code editor with a dark theme. The top bar includes a file explorer on the left, a project name 'entri\_d41\_python\_project' with a 'Version control' dropdown, and a 'Current File' dropdown. The file explorer shows a list of files: 'Session1.py', 'Tuple.py', 'Statement.py', 'functions.py', 'filehandling.py', and 'Assignment4.py'. The 'Assignment4.py' file is open in the editor. The code in the editor is as follows:

```
28 # 4 Explain the difference between local and global variables in a Python function.
29 # Write a program where a global variable and a local variable have the same name and show how
30 # Python differentiates between them.
31 # [ Local variables are defined and can only be accessed from within the function and are
32 # destroyed when the function returns where as Variables that are defined outside of any function are
33 # global variables and these variables can be accessed from anywhere in the program,
34 # including from within functions.
35 # To explicitly define a variable as global, you can use the global keyword.]
36 abc= 10 #global variable
37 def my_function(): 1usage
38     abc = 20 # local variable
39     print("Value of x inside the function (local):", abc)
40 my_function()
41 print("Value of x outside the function (global):", abc)
42
```

Below the code editor is a 'Run' panel. It shows the output of the program:

```
Maximum value of the list is 100
Value of x inside the function (local): 20
Value of x outside the function (global): 10
```

The image shows a code editor window with a dark theme. The top bar displays the project name 'entri\_d41\_python\_project' and 'Version control'. The file explorer on the left shows several files: 'Session1.py', 'Tuple.py', 'Statement.py', 'functions.py', 'filehandling.py', and 'Assignment4.py'. The main editor area contains the following Python code:

```
39     print("Value of x inside the function (local):", abc)
40 my_function()
41 print("Value of x outside the function (global):", abc)
42
43 # 5 Create a function calculate_area(length, width=5) that calculates the area of a rectangle.
44 # If only the length is provided, the function should assume the width is 5.
45 # Show how the function behaves when called with and without the width argument.
46 def calculate_area(length, width=5): #default argument 1 usage
47     area=length * width
48     print('Area of the rectangle= ',area)
49 calculate_area(10)
```

Below the code editor is a 'Run' panel showing the output of the code execution:

```
value of x outside the function (global): 10
Area of the rectangle= 50
```

The image shows a code editor window with a dark theme. The top bar displays the project name 'entri\_d41\_python\_project' and 'Version control'. The file explorer on the left shows a project structure with files: 'on1.py', 'Tuple.py', 'Statement.py', 'functions.py', 'filehandling.py', and 'Assignment4.py'. The main editor area shows the following Python code:

```
39     print("Value of x inside the function (local):", abc)
40 my_function()
41 print("Value of x outside the function (global):", abc)
42
43 # 5 Create a function calculate_area(length, width=5)that calculates the area of a rectangle.
44 # If only the length is provided, the function should assume the width is 5.
45 # Show how the function behaves when called with and without the width argument.
46 def calculate_area(length, width=5): #default argument 1usage
47     area=length * width
48     print('Area of the rectangle= ',area)
49 calculate_area( length: 10, width: 3)
```

Below the editor is a 'Run' panel for 'Assignment4'. It shows the output of the code execution:

```
Area of the rectangle=  30
Process finished with exit code 0
```