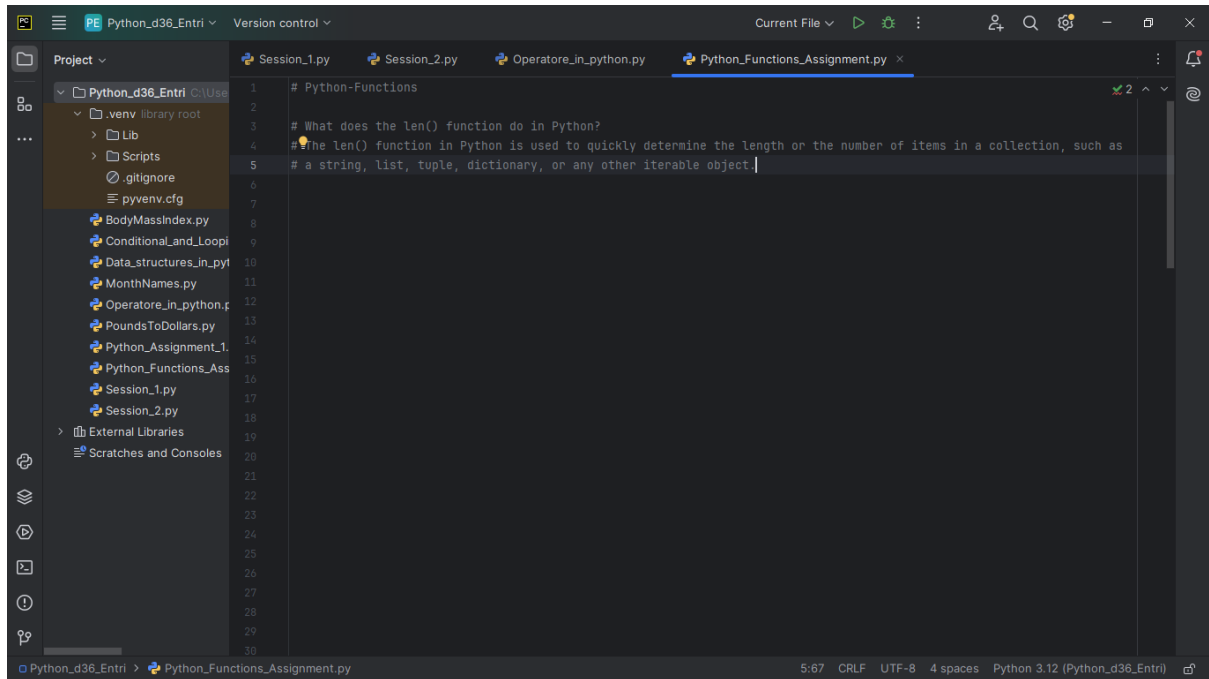


PYTHON ASSIGNMENT: PYTHON - FUNCTIONS

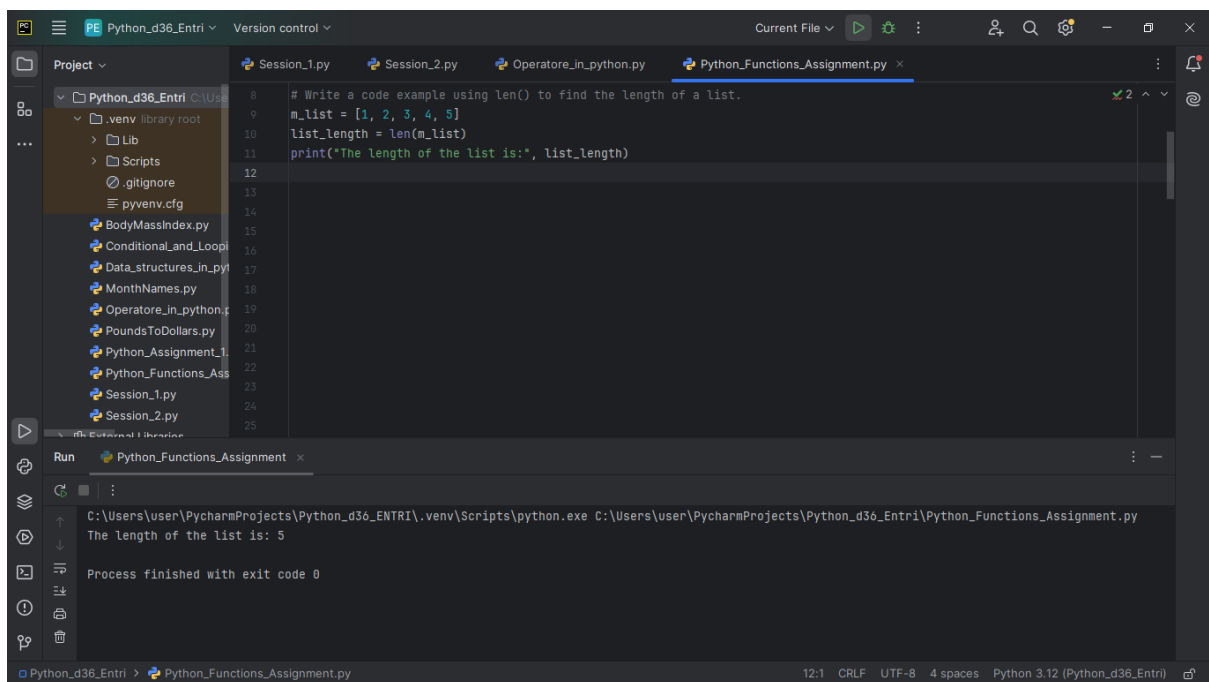
What does the len() function do in Python?



The screenshot shows a code editor with a project named 'Python_d36_Entri'. The file 'Python_Functions_Assignment.py' is open, displaying the following code:

```
1 # Python-Functions
2
3 # What does the len() function do in Python?
4 # The len() function in Python is used to quickly determine the length or the number of items in a collection, such as
5 # a string, list, tuple, dictionary, or any other iterable object
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

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



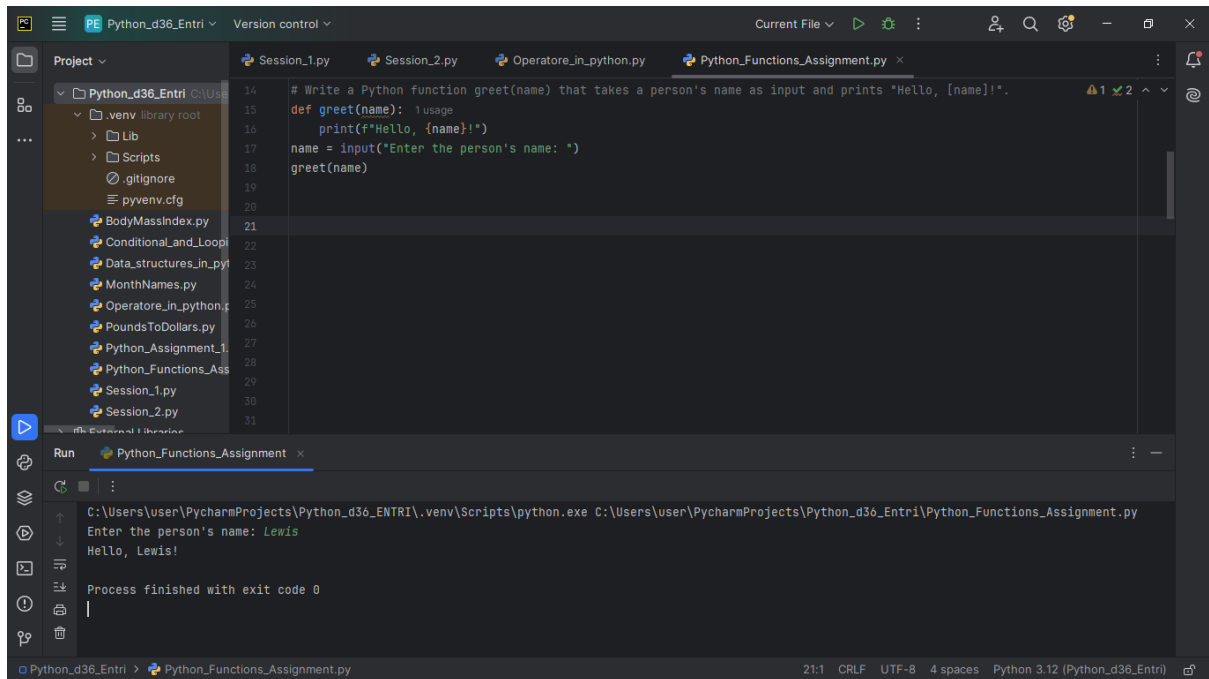
The screenshot shows the same code editor with the file 'Python_Functions_Assignment.py' open, displaying the following code:

```
8 # Write a code example using len() to find the length of a list.
9 m_list = [1, 2, 3, 4, 5]
10 list_length = len(m_list)
11 print('The length of the list is:', list_length)
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

Below the code editor, the 'Run' console shows the output of the code:

```
C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_Entri\Python_Functions_Assignment.py
The length of the list is: 5
Process finished with exit code 0
```

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



The screenshot shows the PyCharm IDE with the file `Python_Functions_Assignment.py` open. The code defines a function `greet(name)` that takes a name as input and prints "Hello, [name]!". The function is called with the input "Lewis". The Run window shows the output "Hello, Lewis!" and "Process finished with exit code 0".

```
14 # Write a Python function greet(name) that takes a person's name as input and prints "Hello, [name]!".
15 def greet(name):
16     print(f"Hello, {name}!")
17     name = input("Enter the person's name: ")
18     greet(name)
19
20
21
22
23
24
25
26
27
28
29
30
31
```

Run: Python_Functions_Assignment

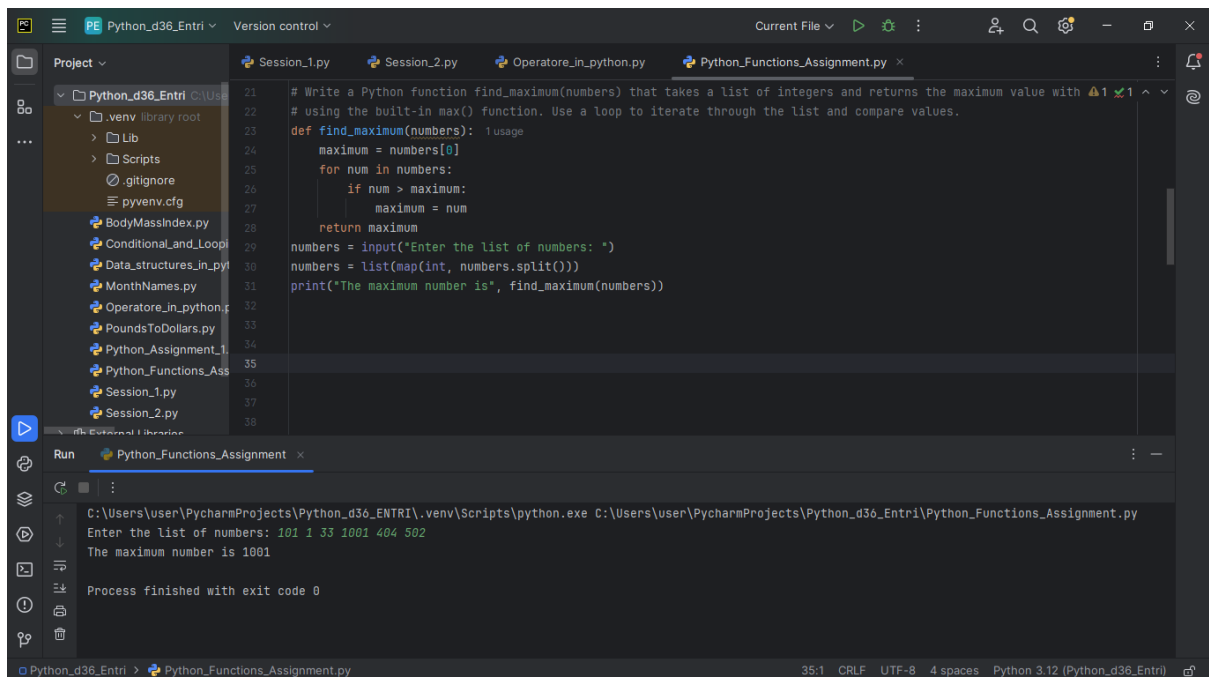
C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_ENTRI\Python_Functions_Assignment.py

Enter the person's name: Lewis

Hello, Lewis!

Process finished with exit code 0

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.



The screenshot shows the PyCharm IDE with the file `Python_Functions_Assignment.py` open. The code defines a function `find_maximum(numbers)` that takes a list of integers and returns the maximum value. The function is called with the input list [101, 1, 33, 1001, 404, 502]. The Run window shows the output "The maximum number is 1001" and "Process finished with exit code 0".

```
21 # Write a Python function find_maximum(numbers) that takes a list of integers and returns the maximum value with
22 # using the built-in max() function. Use a loop to iterate through the list and compare values.
23 def find_maximum(numbers):
24     maximum = numbers[0]
25     for num in numbers:
26         if num > maximum:
27             maximum = num
28     return maximum
29
30 numbers = input("Enter the list of numbers: ")
31 numbers = list(map(int, numbers.split()))
32 print("The maximum number is", find_maximum(numbers))
33
34
35
36
37
38
```

Run: Python_Functions_Assignment

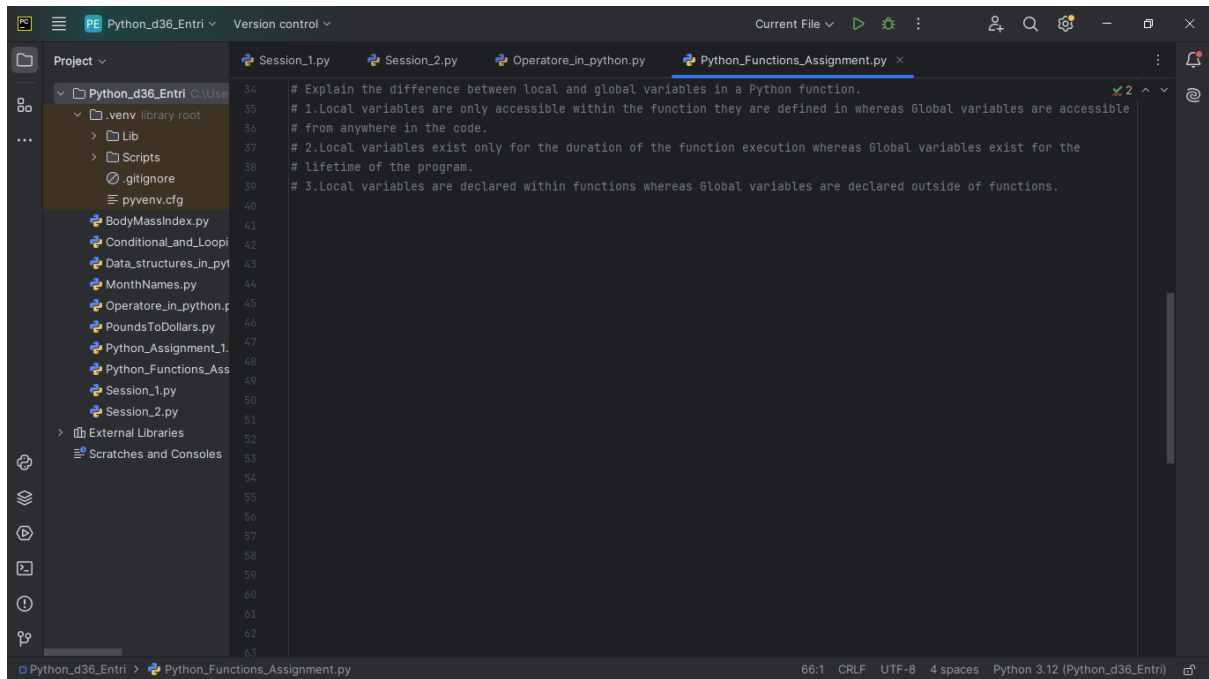
C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_ENTRI\Python_Functions_Assignment.py

Enter the list of numbers: 101 1 33 1001 404 502

The maximum number is 1001

Process finished with exit code 0

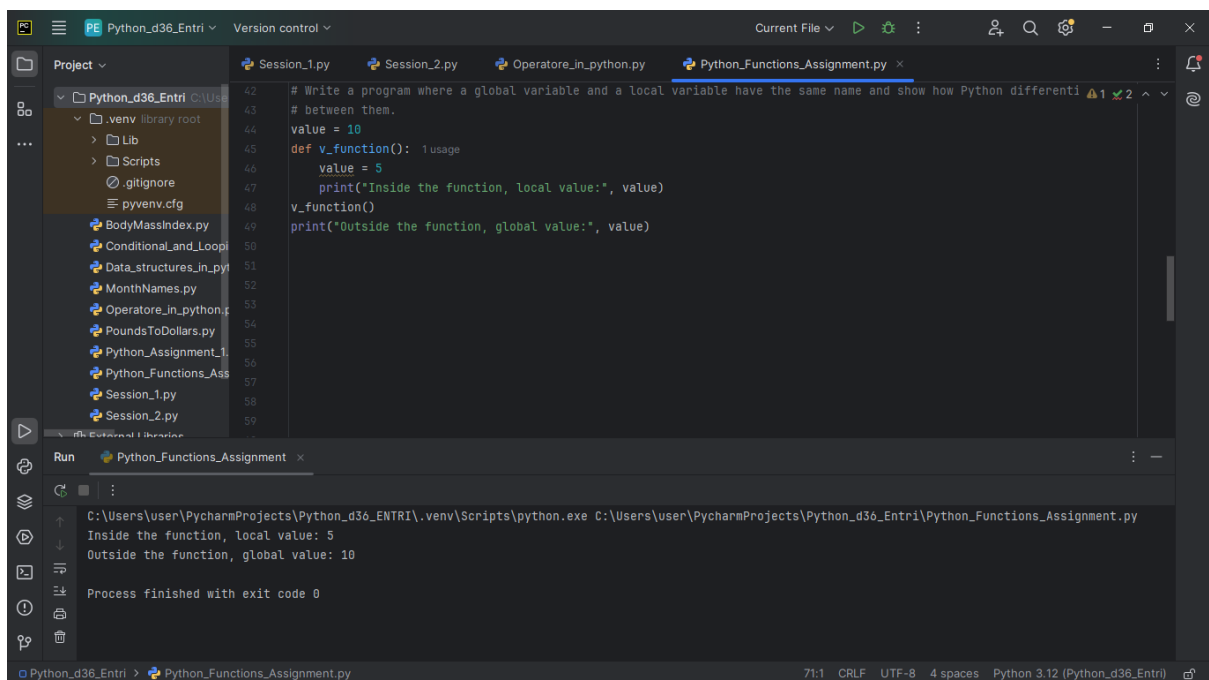
Explain the difference between local and global variables in a Python function.



The screenshot shows the PyCharm IDE with a project named 'Python_d36_Entri'. The file 'Python_Functions_Assignment.py' is open, displaying the following comments:

```
34 # Explain the difference between local and global variables in a Python function.
35 # 1. Local variables are only accessible within the function they are defined in whereas Global variables are accessible
36 # from anywhere in the code.
37 # 2. Local variables exist only for the duration of the function execution whereas Global variables exist for the
38 # lifetime of the program.
39 # 3. Local variables are declared within functions whereas Global variables are declared outside of functions.
```

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



The screenshot shows the PyCharm IDE with the same project. The file 'Python_Functions_Assignment.py' is open, displaying the following code:

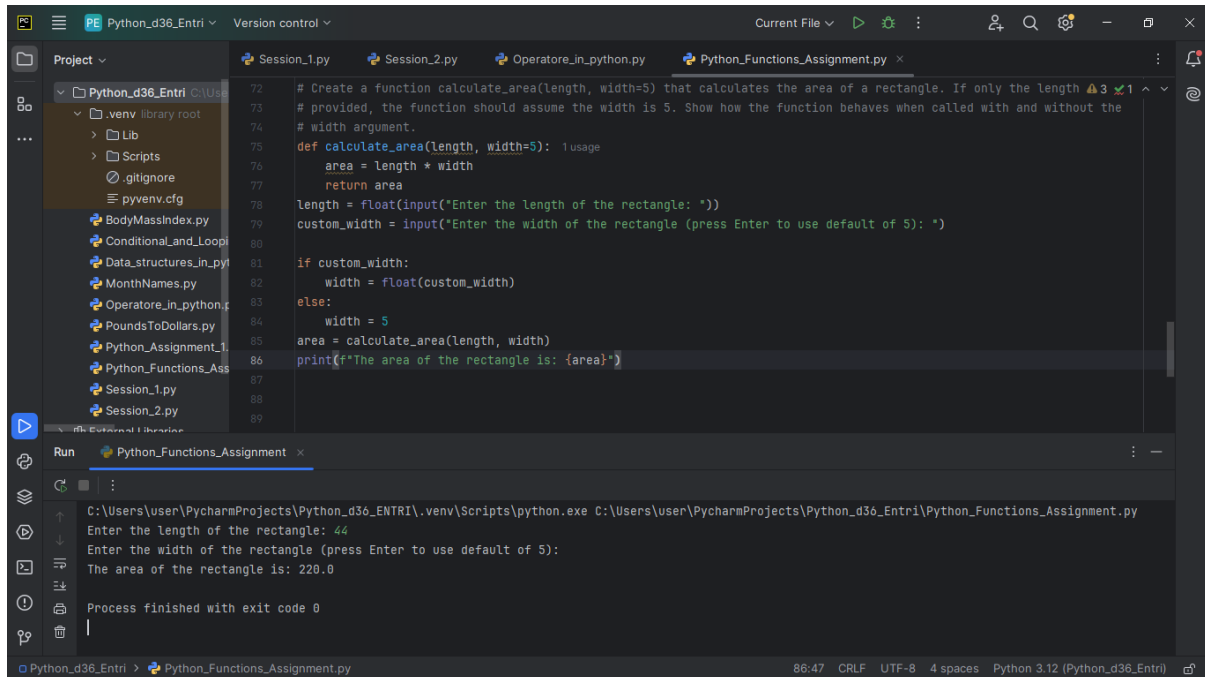
```
42 # Write a program where a global variable and a local variable have the same name and show how Python differentiates
43 # between them.
44 value = 10
45 def v_function():
46     value = 5
47     print("Inside the function, local value:", value)
48 v_function()
49 print("Outside the function, global value:", value)
```

The Run window shows the output of the program:

```
C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_ENTRI\Python_Functions_Assignment.py
Inside the function, local value: 5
Outside the function, global value: 10
Process finished with exit code 0
```

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.

Width = 5:



The screenshot shows the PyCharm IDE with the file `Python_Functions_Assignment.py` open. The code defines a function `calculate_area` with a default width of 5. The `Run` console shows the program's execution: it prompts for the length (44) and width (pressing Enter to use the default of 5), and outputs the area as 220.0.

```
72 # Create a function calculate_area(length, width=5) that calculates the area of a rectangle. If only the length
73 # provided, the function should assume the width is 5. Show how the function behaves when called with and without the
74 # width argument.
75 def calculate_area(length, width=5):
76     area = length * width
77     return area
78 length = float(input("Enter the length of the rectangle: "))
79 custom_width = input("Enter the width of the rectangle (press Enter to use default of 5): ")
80
81 if custom_width:
82     width = float(custom_width)
83 else:
84     width = 5
85 area = calculate_area(length, width)
86 print(f"The area of the rectangle is: {area}")
87
88
89
```

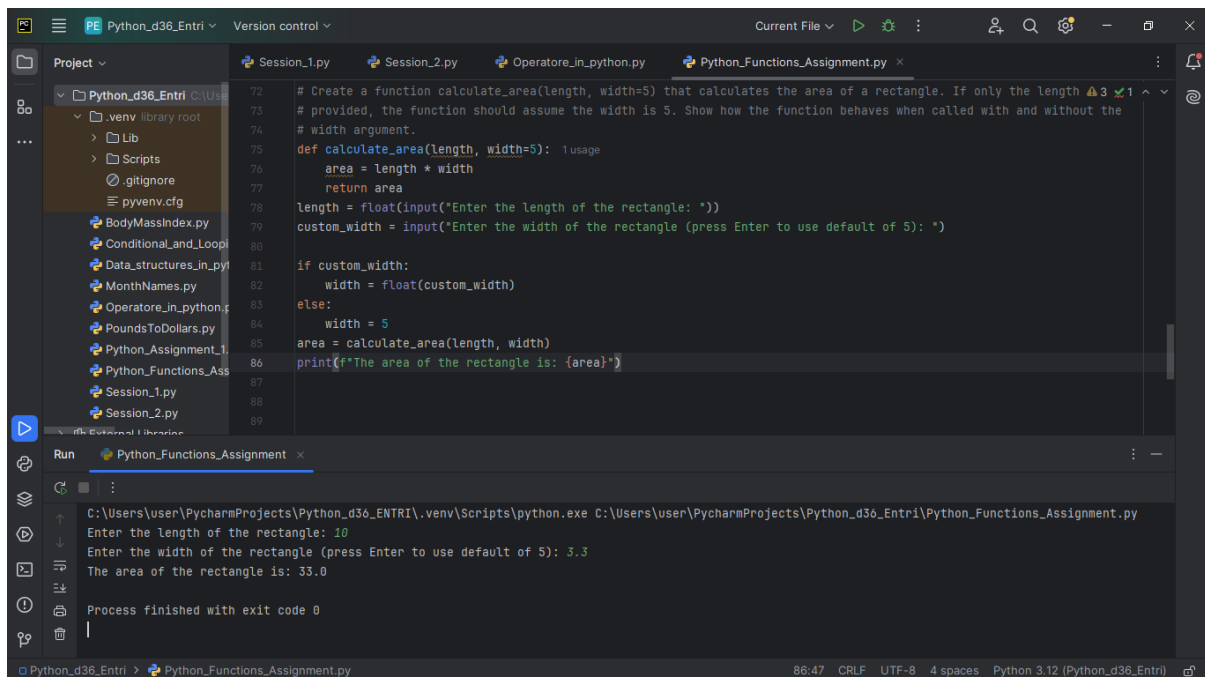
Run Python_Functions_Assignment

C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_ENTRI\Python_Functions_Assignment.py

Enter the length of the rectangle: 44
Enter the width of the rectangle (press Enter to use default of 5):
The area of the rectangle is: 220.0

Process finished with exit code 0

Custom Width:



The screenshot shows the same PyCharm IDE setup as the first image. In this run, the user provides a custom width of 3.3. The `Run` console shows the program's execution: it prompts for the length (10) and width (3.3), and outputs the area as 33.0.

```
72 # Create a function calculate_area(length, width=5) that calculates the area of a rectangle. If only the length
73 # provided, the function should assume the width is 5. Show how the function behaves when called with and without the
74 # width argument.
75 def calculate_area(length, width=5):
76     area = length * width
77     return area
78 length = float(input("Enter the length of the rectangle: "))
79 custom_width = input("Enter the width of the rectangle (press Enter to use default of 5): ")
80
81 if custom_width:
82     width = float(custom_width)
83 else:
84     width = 5
85 area = calculate_area(length, width)
86 print(f"The area of the rectangle is: {area}")
87
88
89
```

Run Python_Functions_Assignment

C:\Users\user\PycharmProjects\Python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\Python_d36_ENTRI\Python_Functions_Assignment.py

Enter the length of the rectangle: 10
Enter the width of the rectangle (press Enter to use default of 5): 3.3
The area of the rectangle is: 33.0

Process finished with exit code 0