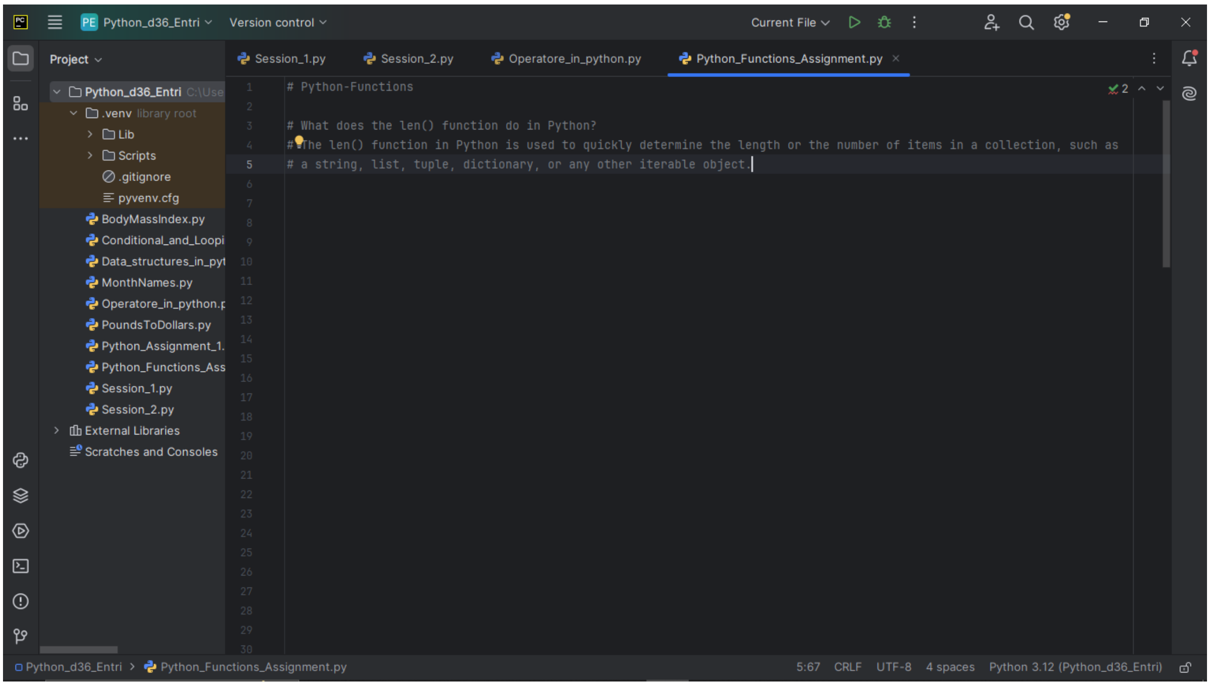
PYTHON ASSIGNMENT: PYTHON - FUNCTIONS

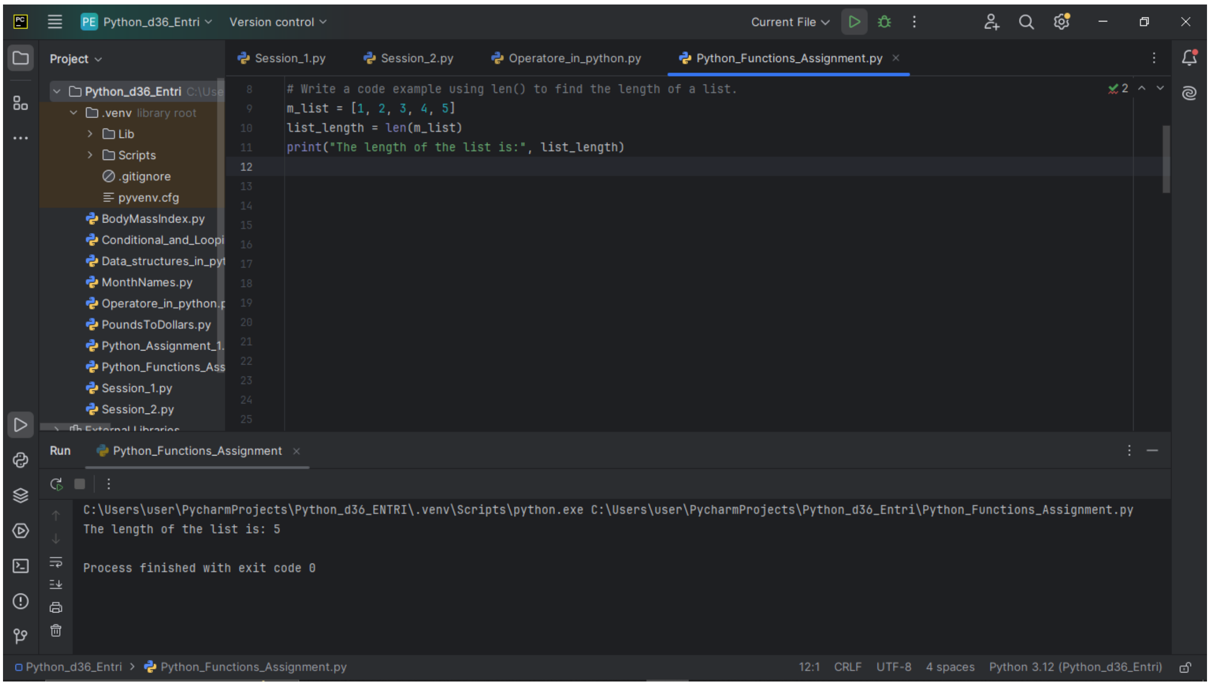
What does the len() function do in Python?



```python
# Python-Functions

# What does the len() function do in Python?
#The len() function in Python is used to quickly determine the length or the number of items in a collection, such as
# a string, list, tuple, dictionary, or any other iterable object.
```

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

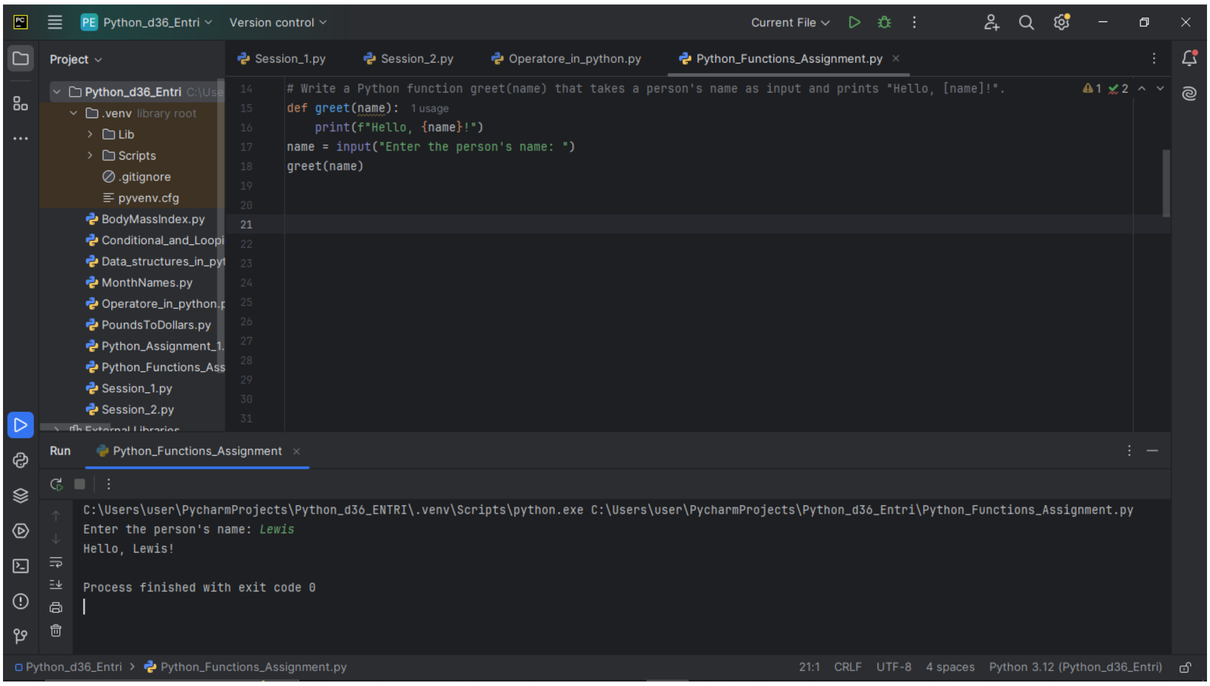


```python
# Write a code example using len() to find the length of a list.
m_list = [1, 2, 3, 4, 5]
list_length = len(m_list)
print("The length of the list is:", list_length)
```

```
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]!".

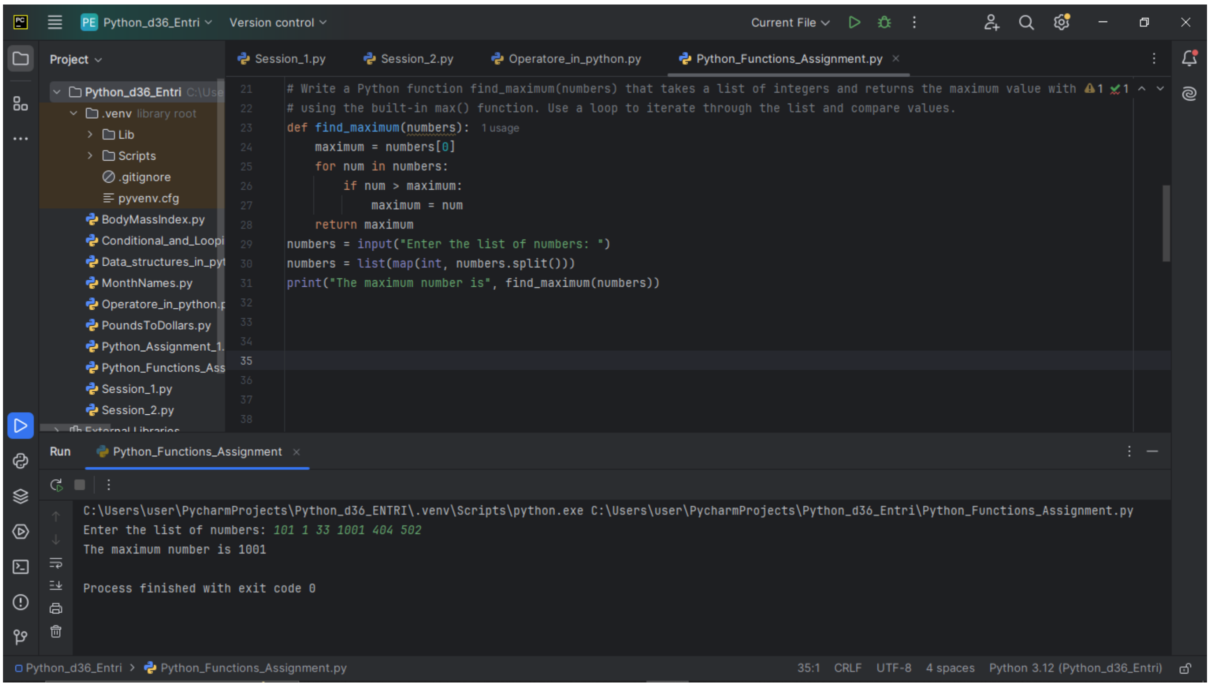


```
# Write a Python function greet(name) that takes a person's name as input and prints "Hello, [name]!".
def greet(name):
    print(f"Hello, {name}!")
name = input("Enter the person's name: ")
greet(name)
```

```
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.



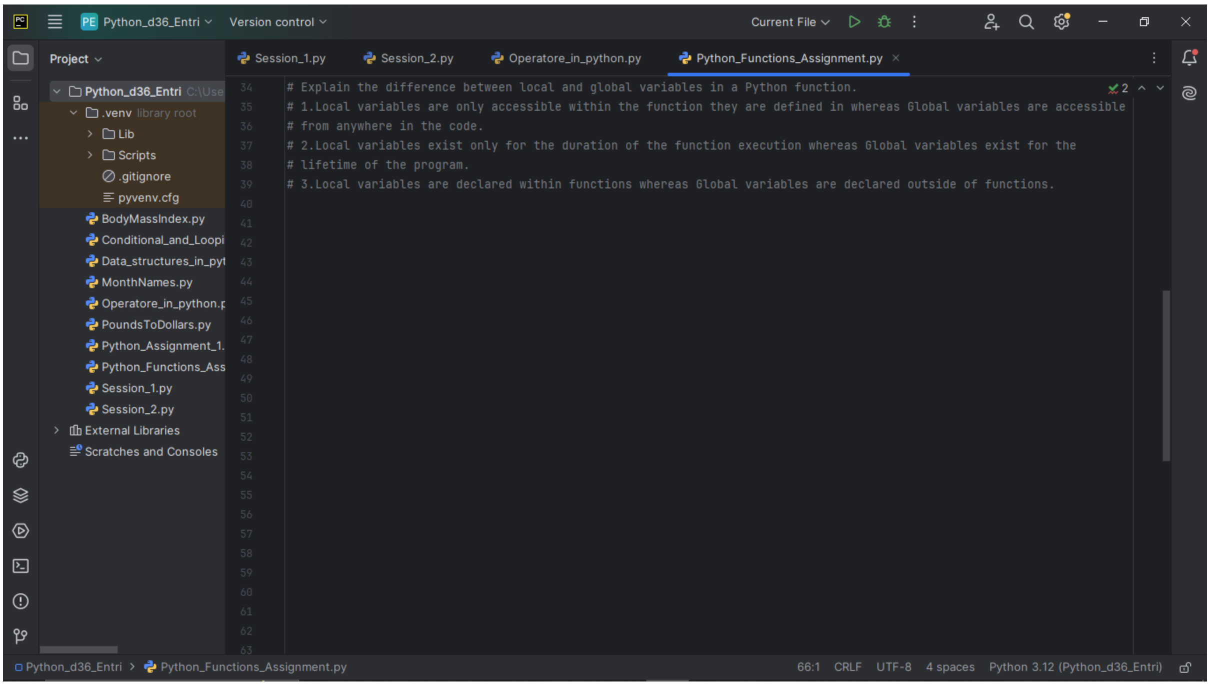
```
# Write a Python function find_maximum(numbers) that takes a list of integers and returns the maximum value with
# using the built-in max() function. Use a loop to iterate through the list and compare values.
def find_maximum(numbers):
    maximum = numbers[0]
    for num in numbers:
        if num > maximum:
            maximum = num
    return maximum
numbers = input("Enter the list of numbers: ")
numbers = list(map(int, numbers.split()))
print("The maximum number is", find_maximum(numbers))
```

```
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
```

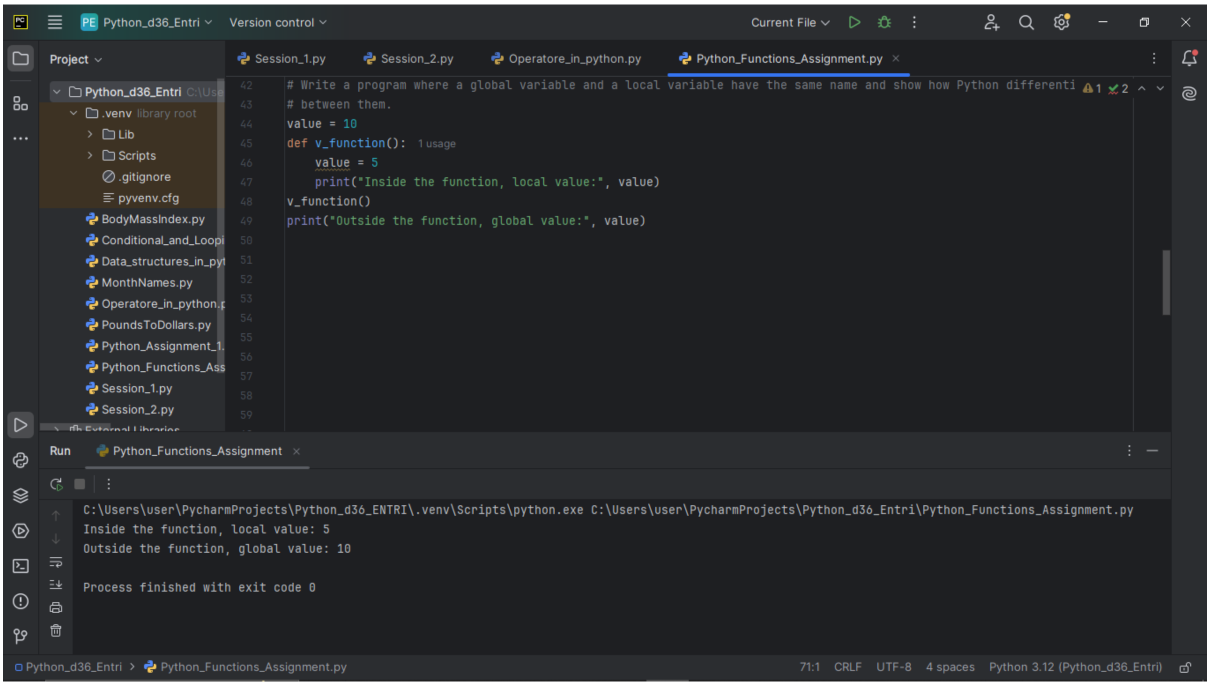
Python_d36_Entri
Version control
Current File
Project
Session_1.py
Session_2.py
Operatore_in_python.py
Python_Functions_Assignment.py
Run
Python_Functions_Assignment
35:1 CRLF UTF-8 4 spaces Python 3.12 (Python_d36_Entri)

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



```python
# Explain the difference between local and global variables in a Python function.
# 1.Local variables are only accessible within the function they are defined in whereas Global variables are accessible
# from anywhere in the code.
# 2.Local variables exist only for the duration of the function execution whereas Global variables exist for the
# lifetime of the program.
# 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.



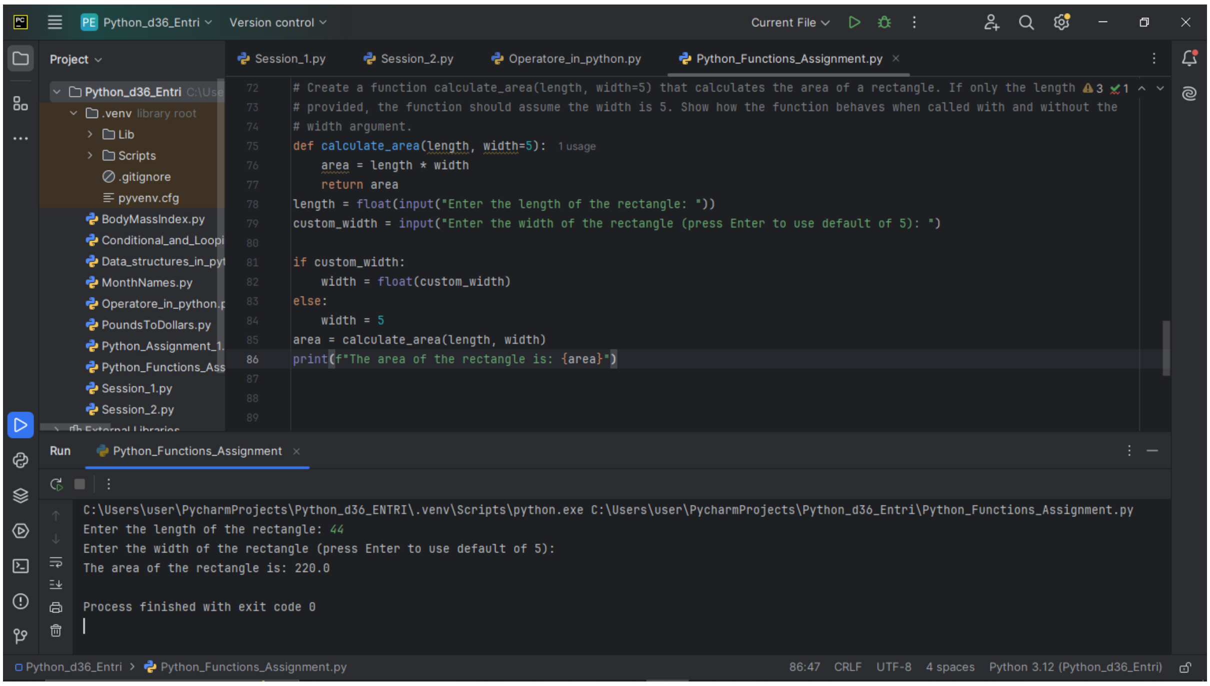
```python
# Write a program where a global variable and a local variable have the same name and show how Python differenti
# between them.
value = 10
def v_function():
    value = 5
    print("Inside the function, local value:", value)
v_function()
print("Outside the function, global value:", value)
```

```
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:



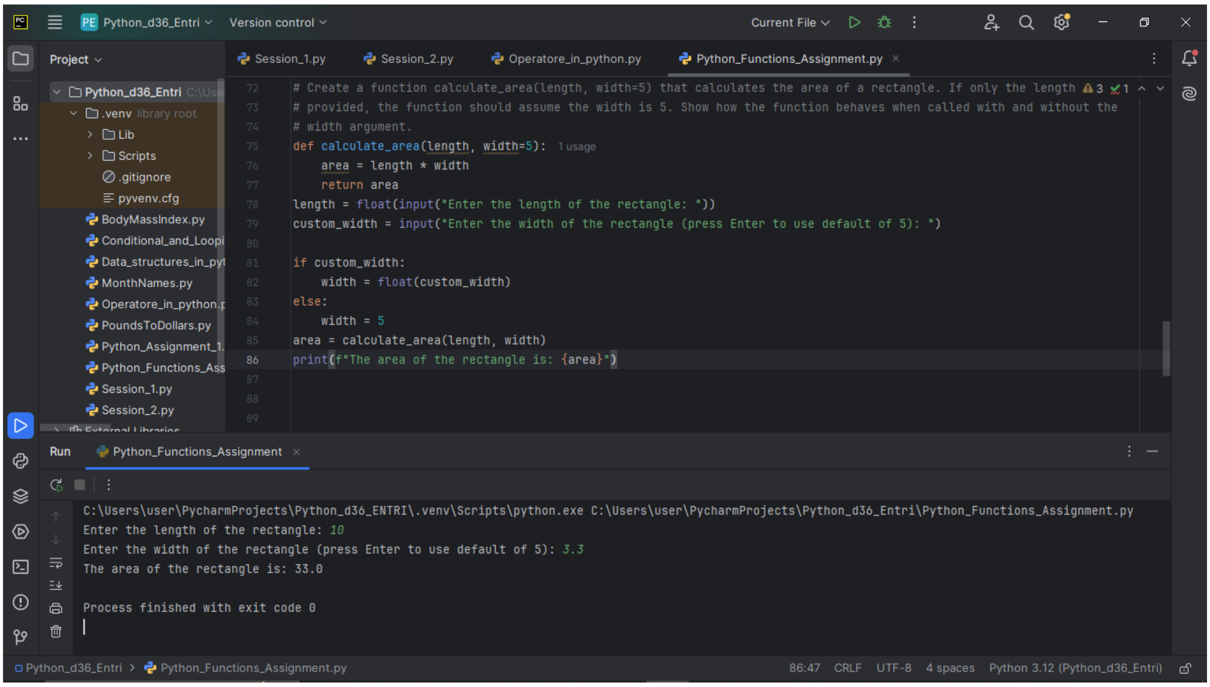
```python
# Create a function calculate_area(length, width=5) that calculates the area of a rectangle. If only the length
# provided, the function should assume the width is 5. Show how the function behaves when called with and without the
# width argument.
def calculate_area(length, width=5):
    area = length * width
    return area
length = float(input("Enter the length of the rectangle: "))
custom_width = input("Enter the width of the rectangle (press Enter to use default of 5): ")

if custom_width:
    width = float(custom_width)
else:
    width = 5
area = calculate_area(length, width)
print(f"The area of the rectangle is: {area}")
```

```
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:



```python
# Create a function calculate_area(length, width=5) that calculates the area of a rectangle. If only the length
# 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
length = float(input("Enter the length of the rectangle: "))
custom_width = input("Enter the width of the rectangle (press Enter to use default of 5): ")

if custom_width:
    width = float(custom_width)
else:
    width = 5
area = calculate_area(length, width)
print(f"The area of the rectangle is: {area}")
```

```
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
```