**Shyam Verma**

**Write a Function to Calculate the Area of a Rectangle**: Create a function called calculate\_area that takes two arguments: the **length** and **width** of a rectangle. The function should return the area of the rectangle.



Function Definition: The function calculate\_area was defined to take two parameters, length and width, associated with the rectangle's dimension.

Return Statement: It calculates the area using the formula of length times width and returns the value of the result.

Function Call: This function is invoked with argument values as 33 and 11 and its return value was being assigned to a variable named area.

Output Formatted The output was printed using an f-string for a readable message: area of the rectangle: 363.

**2.Write a Function to Check if a Number is Prime**: Create a function called is\_prime that takes one argument, a positive integer n, and returns True if the number is prime, otherwise False.



Function Explanation:

The function is\_prime(n) checks whether a number n is prime.

It returns False for any numbers less than or equal to 1.

Logic for Checking Prime:

For any number greater than 1, it loops through from 2 up to the square root of n (inclusive).

If any of those numbers evenly divide n, then it returns False.

If no divisors are found, it returns True.

Sample Usage:

We call the function with n=7, and it prints the result (True), because 7 is a prime number.

1. **Write a Function to Count the Occurrences of Each Character in a String**: Create a function called char\_count that takes a string s and returns a dictionary where the keys are the characters, and the values are the number of times each character appears in the string.



**Flow with "hello world":**

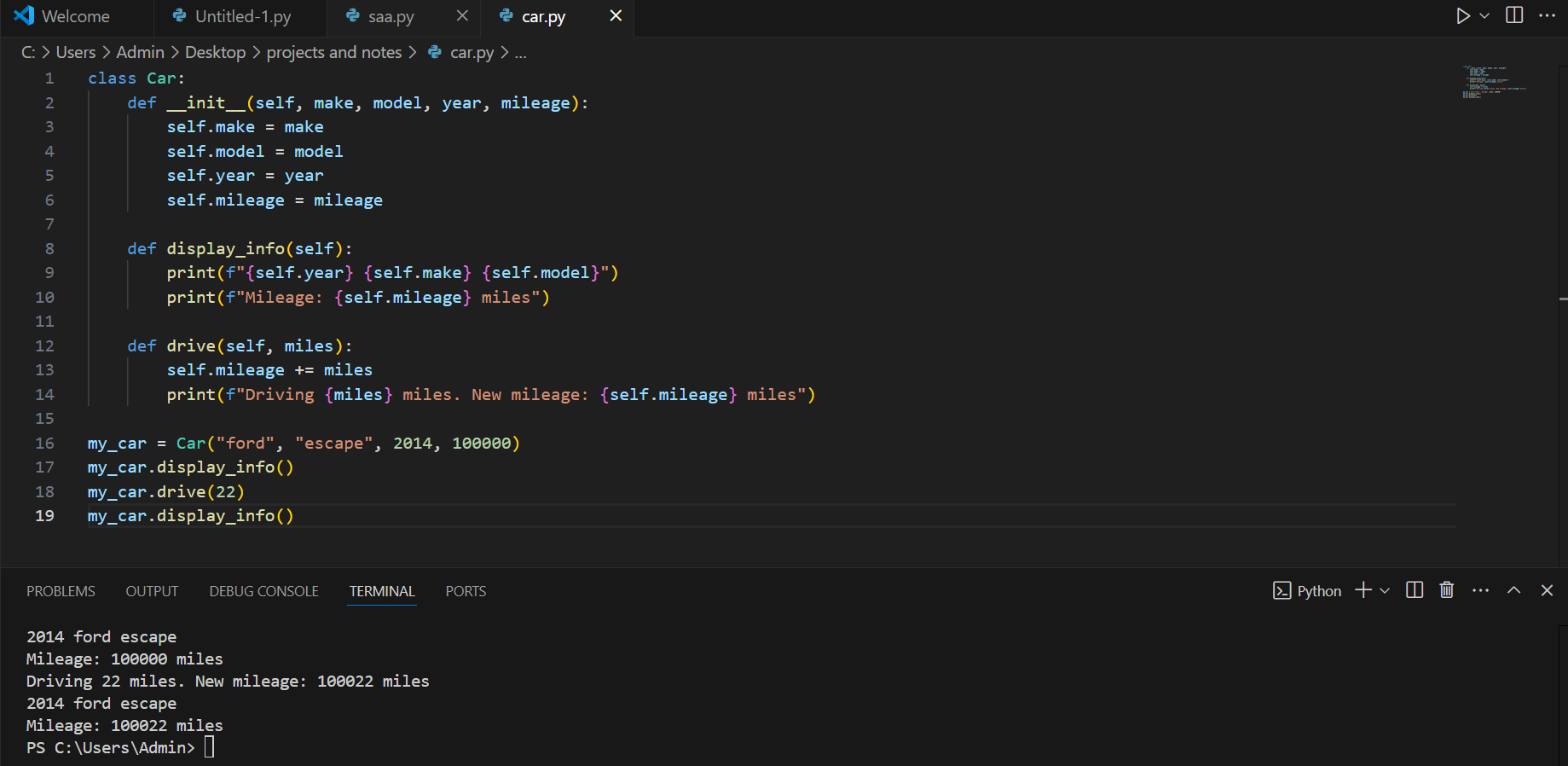
* Start: count = {}
* Process 'h': {'h': 1}
* Process 'e': {'h': 1, 'e': 1}
* Process 'l': {'h': 1, 'e': 1, 'l': 1}
* Process second 'l': {'h': 1, 'e': 1, 'l': 2}
* Process 'o': {'h': 1, 'e': 1, 'l': 2, 'o': 1}
* And so on until every character is counted.

4.**Create a Car Class**: Define a class called Car. The Car class should have the following attributes:

* + make (string): The make of the car (e.g., "Ford").
  + model (string): The model of the car (e.g., "escape").
  + year (integer): The manufacturing year of the car (e.g., 2014).
  + mileage (float): The current mileage of the car (e.g., 100000 miles).

The class should also have the following methods:

* + display\_info: Prints out the car's details (make, model, year, mileage).
  + drive: Takes an argument miles (the number of miles driven), and increases the car's mileage by that amount.

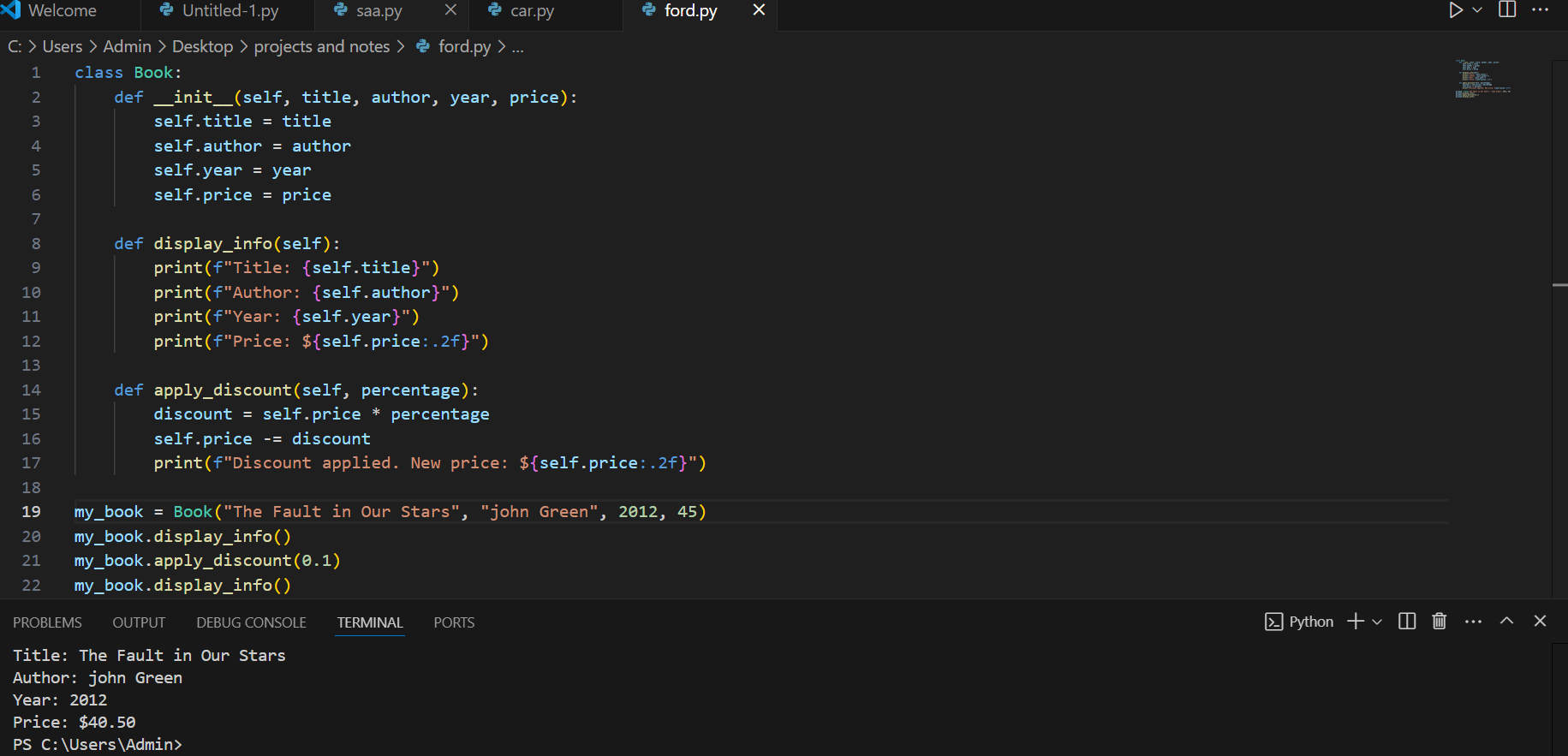


This code defines a Car class with attributes (make, model, year, mileage) and methods to display the car's details and simulate driving by updating its mileage. An instance of the class is created, its details are displayed, it is driven 22 miles (updating the mileage), and the updated details are displayed again.

1. **Create a Book Class**: Define a class called Book. The Book class should have the following attributes:
   * title (string): The title of the book (e.g., "The fault in our star").
   * author (string): The author of the book (e.g., "john green").
   * year (integer): The year the book was published.
   * price (float): The price of the book.

The class should also have the following methods:

* + display\_info: Prints the book's title, author, year, and price.
  + apply\_discount: Takes a percentage (e.g., 0.1 for a 10% discount) and applies it to the price of the book.



The code defines a Book class with attributes such as title, author, year, and price, and two methods. The display\_info method displays the book's details, including the title, author, year, and price. The apply\_discount method takes a percentage and applies the discount to the book's price, updating it accordingly. In this example usage, it creates a Book object with the title "The Fault in Our Stars," from author John Green, published in 2012 and priced at $45. It then calls a method, display\_info, which first shows the initial details; afterward, it applies a 10% discount, dropping the price down to $40.50 and then displays the details once again. The code simulates a program managing a book's information and applying discounts.