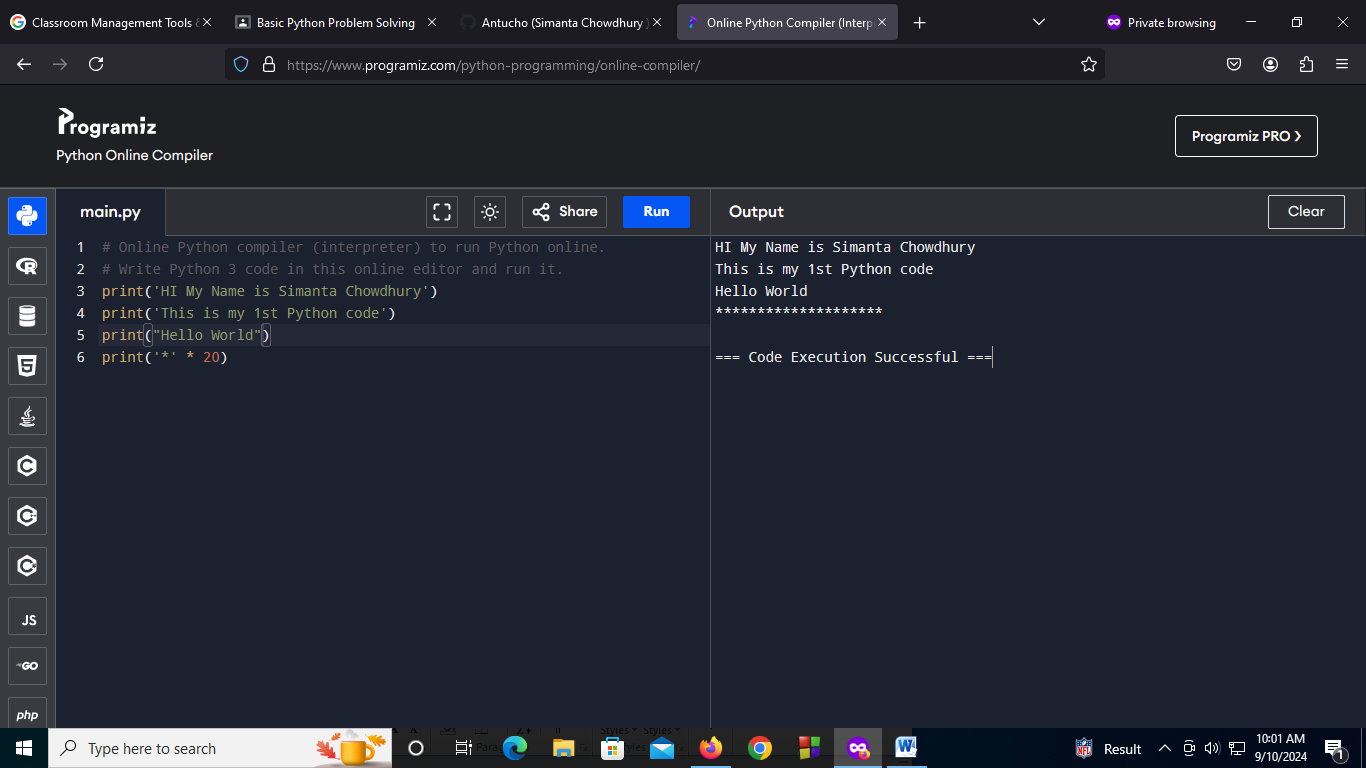
In the Lesson-01 code work like:

In Python Print command print the message in which is mentioned.

Like:

(HI My Name is Simanta Chowdhury, This is my 1st Python code, Hello World) this message will display in out put screen when the code will run. To print the message we have to use ‘print()’ command.

In python print command can also do mathematical trumps. So the last print command in code will print star(\*) in 20 time’s.

Output : 

In the Lesson-02 code work like:

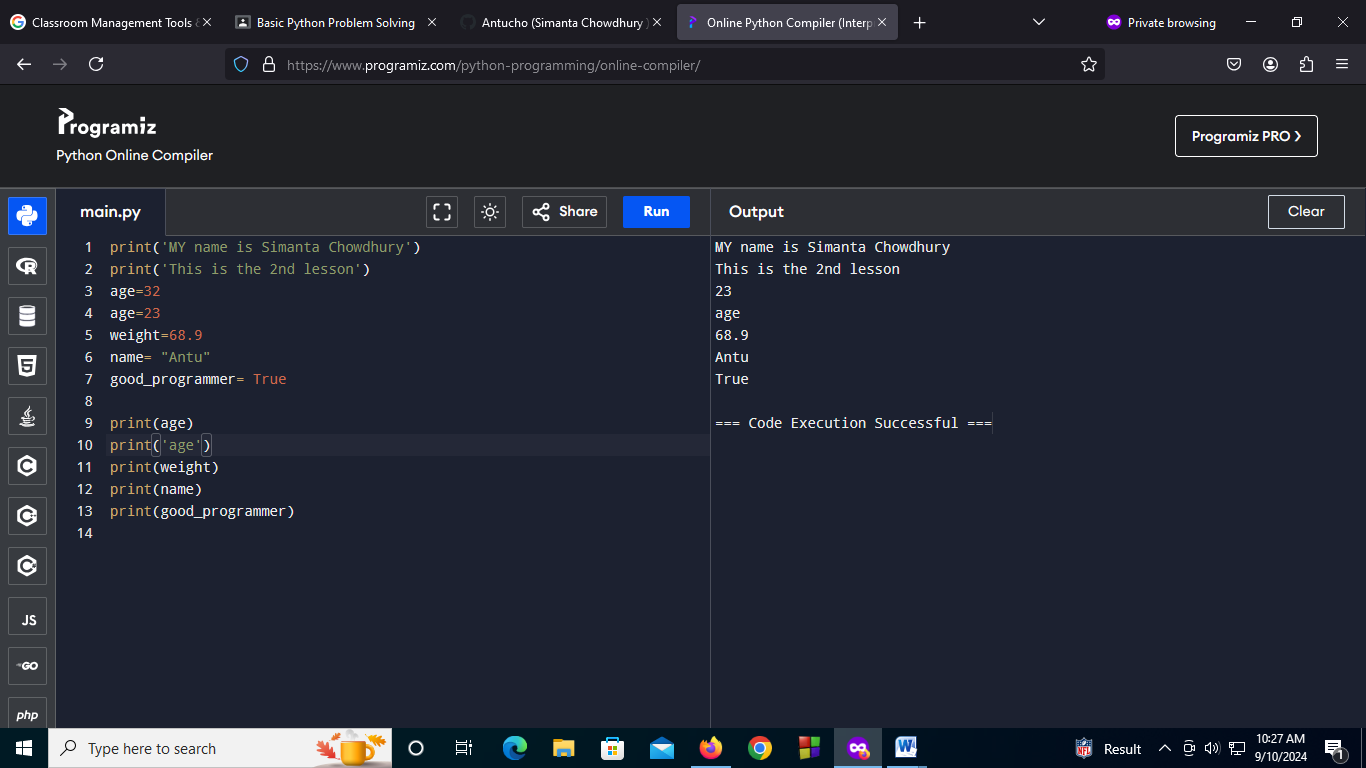
In Python Print command print the message in which is mentioned.

1st line age=32 and than age= 23. When the code will run output will show the the 2nd age means 23, because in python code will run the order of top to bottom. So compiler go to 1st age=32 than 2nd age=23 and then the value of age is replaced with 23.

In python we don’t have to mention the data type, that why we don’t declared the data type,

In print section print(age) refers a variable and print(‘age’) refers string . (“ ”) refers strings.

Output:



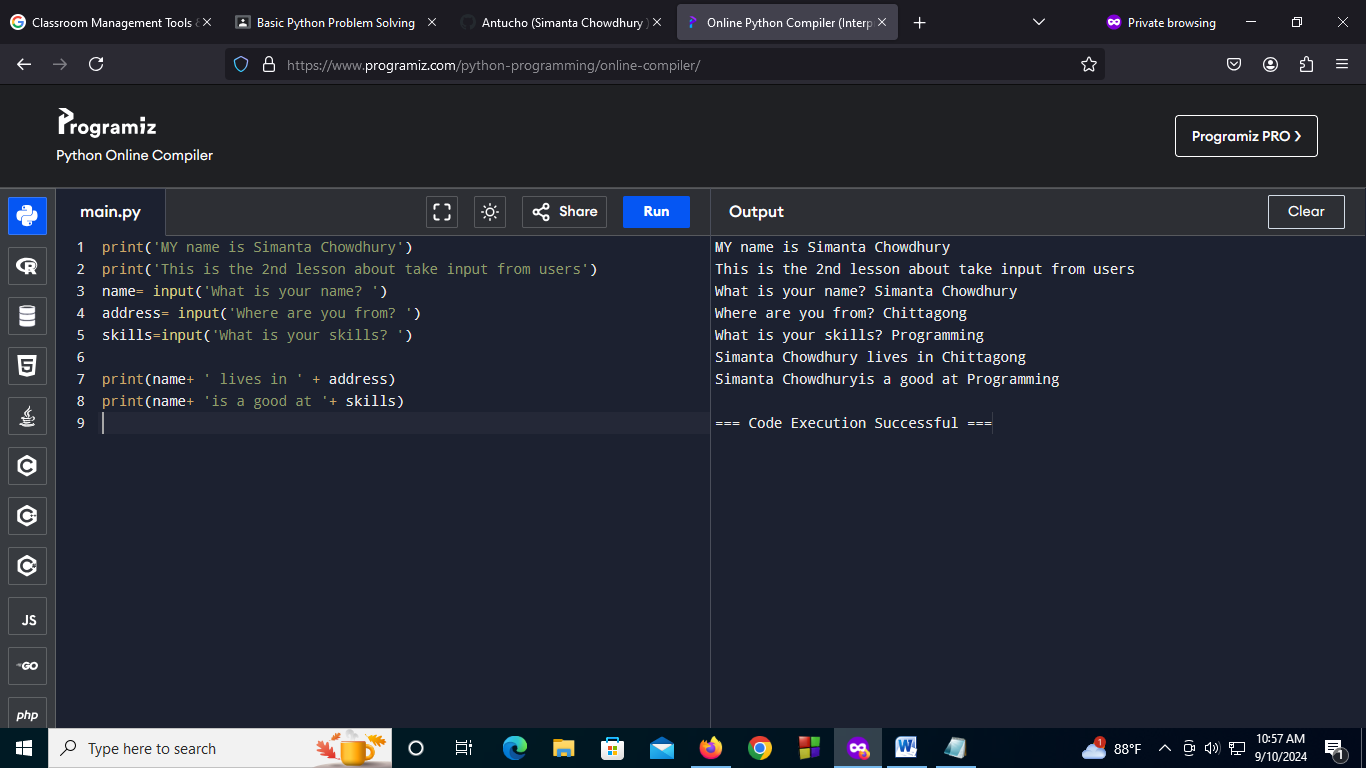
In the Lesson-03 code work like:

In Python Print command print the message in which is mentioned.

To take input from uses Python have a building function called “Input()”.

In this code we take input from users by using input() building function and store the data in variable (name, address, skills). Than in print section we marge variable and line text by using “+” operation.

Output:



In the Lesson-04 code work like:

In Python Print command print the message in which is mentioned.

In python we use # to add single lien comment. And

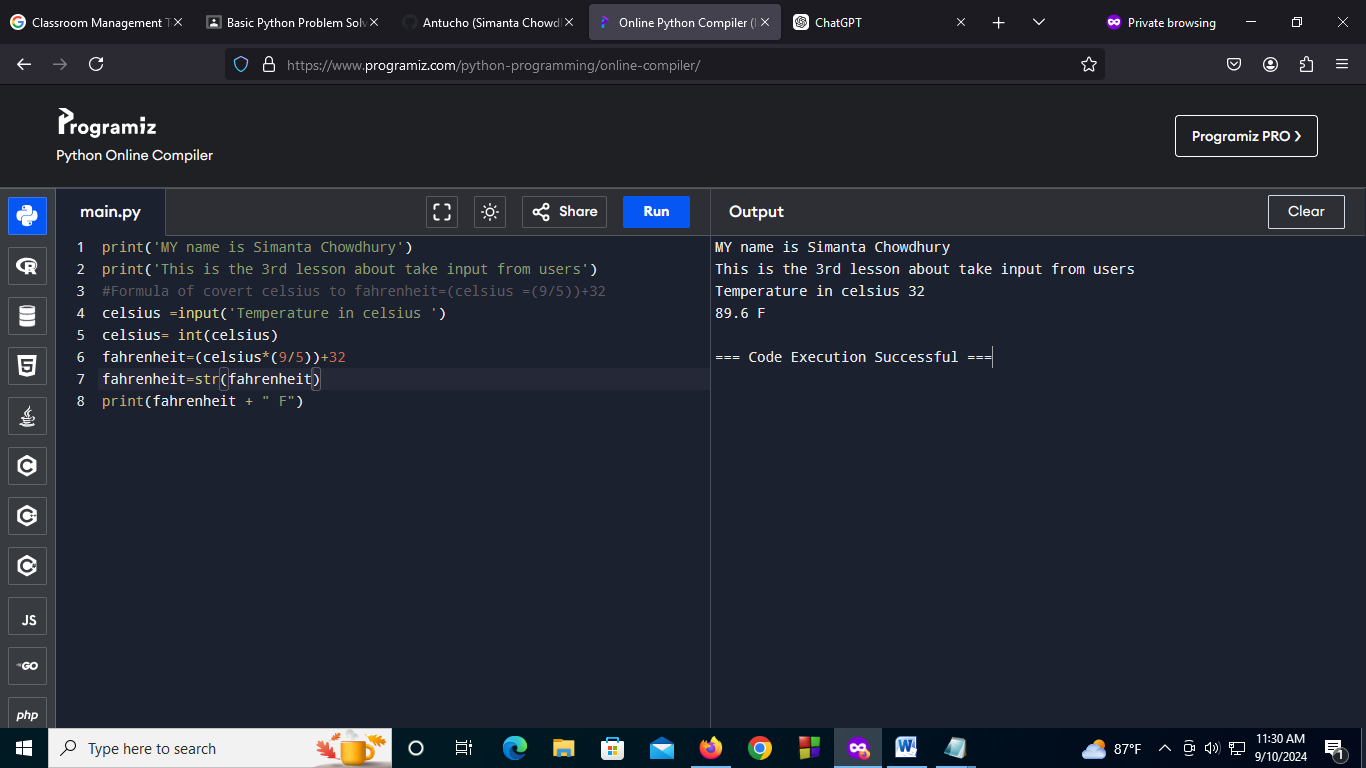
“‘

’’’ for multiline comment.

This code for convert temperature in Celsius to Fahrenheit. For this we take a temperature as a input from users by using input() function and store the data in variable ‘celsius’. Than declarer the Celsius as integer by using ‘int(celsius)’ to marge integer to float. Than use the formula to convert Celsius to Fahrenheit Fahrenheit=(Celsius\*(9/5))+32. And declarer the ‘fahrenheit=str(fahrenheit)’ to marge string to float. Otherwise the code will shows error.

Than print the Fahrenheit.

Output:

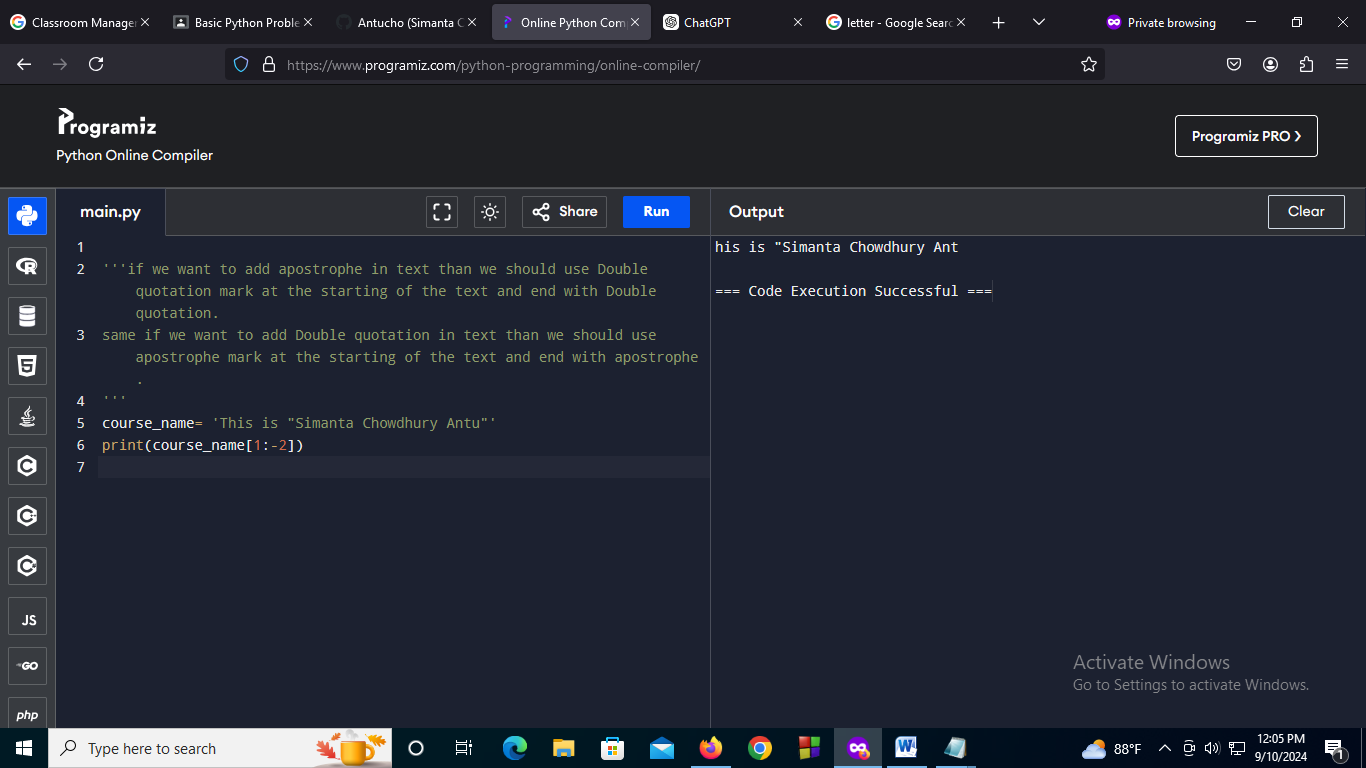


In the Lesson-05 code work like:

In this code we will learned about how to use apostrophe(‘) and Double quotation(“) using a print statement. For this if we want to add apostrophe in text than we should use Double quotation mark at the starting of the text and end with Double quotation same if we want to add Double quotation in text than we should use apostrophe mark at the starting of the text and end with apostrophe.

It works like an array. We used the 0 based index to access T and u of the string. Then we print apart of the string using[starting index : ending index]

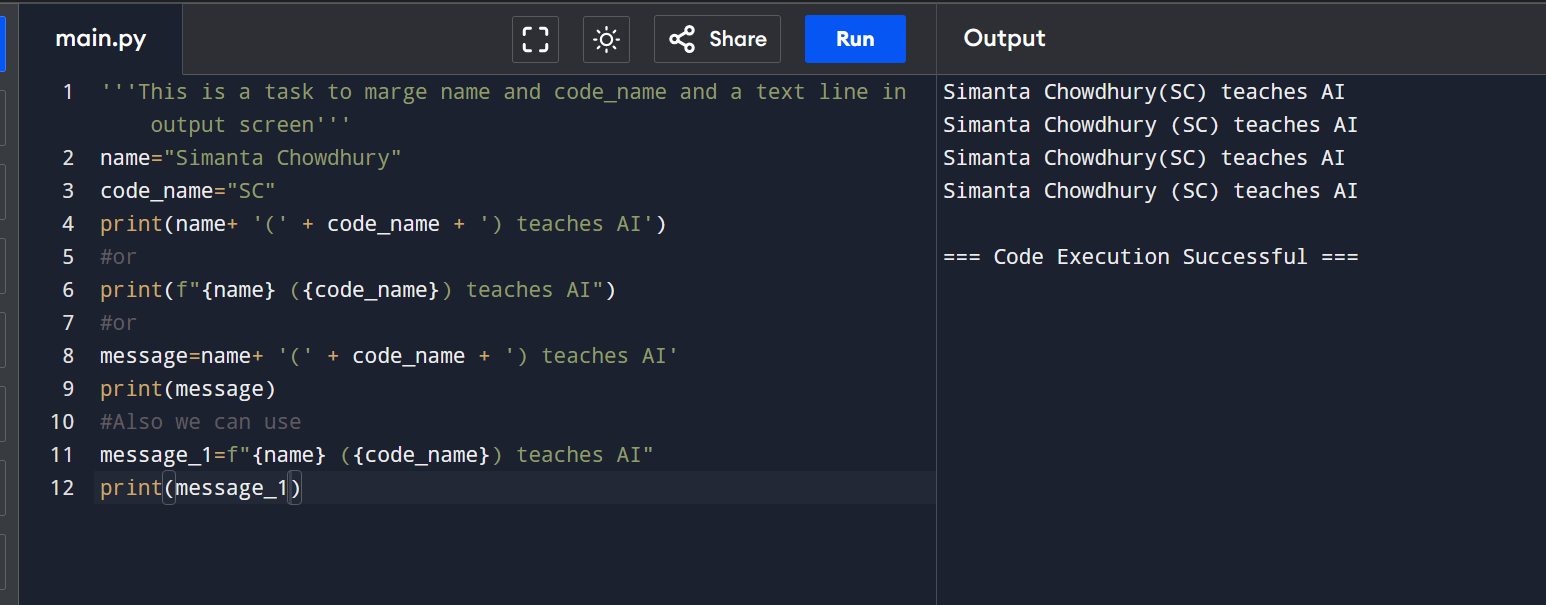
Output:



In the Lesson-06 code work like:

This lesson about on a task. The task is to marge strings and a text line. Like a teacher teaches AI. We have to marge the teacher’s name and code name with a text line that he/she teaches AI.

Output:

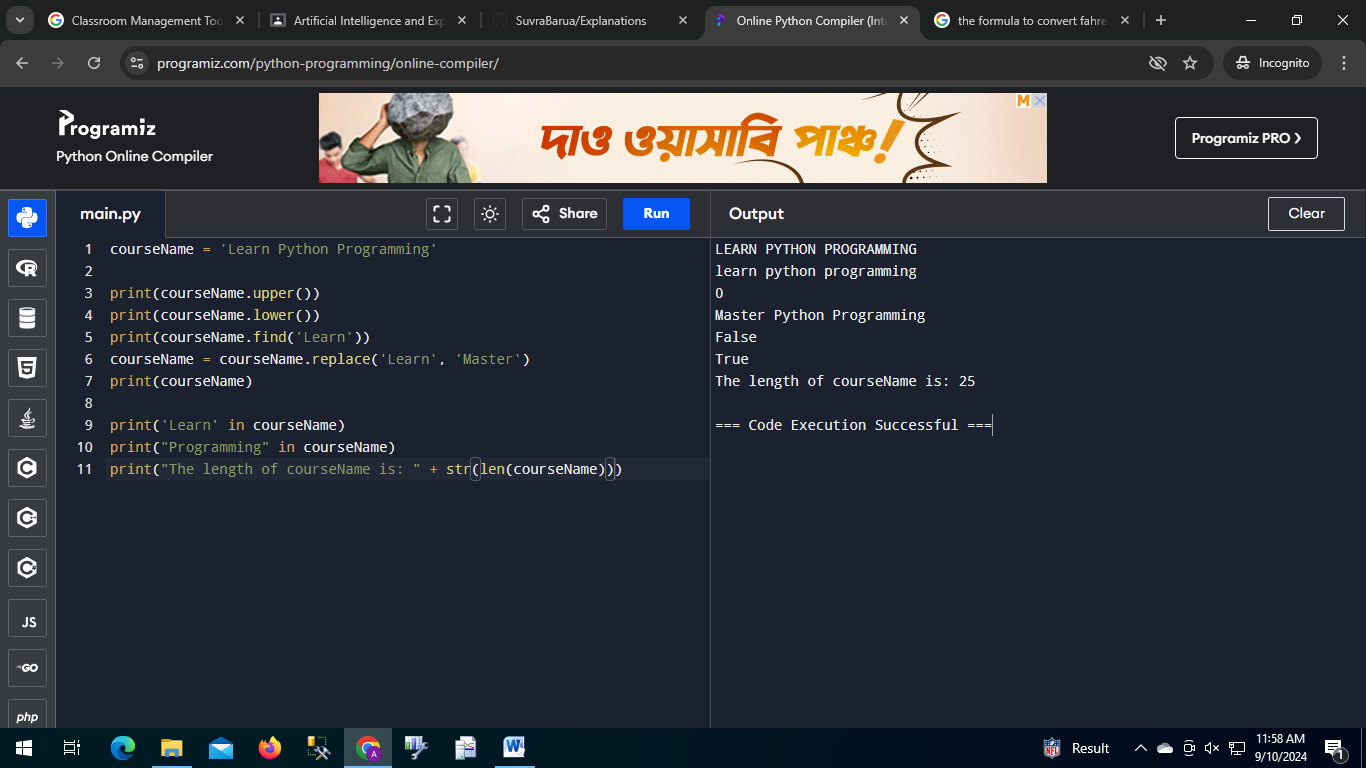


In the Lesson-07 code work like:

In this lesson, we learned about the difference between function and method in python.

A function can be called for any objects but a method can only be called from a specific objects and we can access them using the (.)dot operator.

In this lesson, we used some method for strings to manipulate them. upper() to make the string uppercase, lower() to make the string lowercase, find(“substring”) to find any substring inside of a string, replace(“sbustringToReplace”, “substitute” ) to replace some of the string.

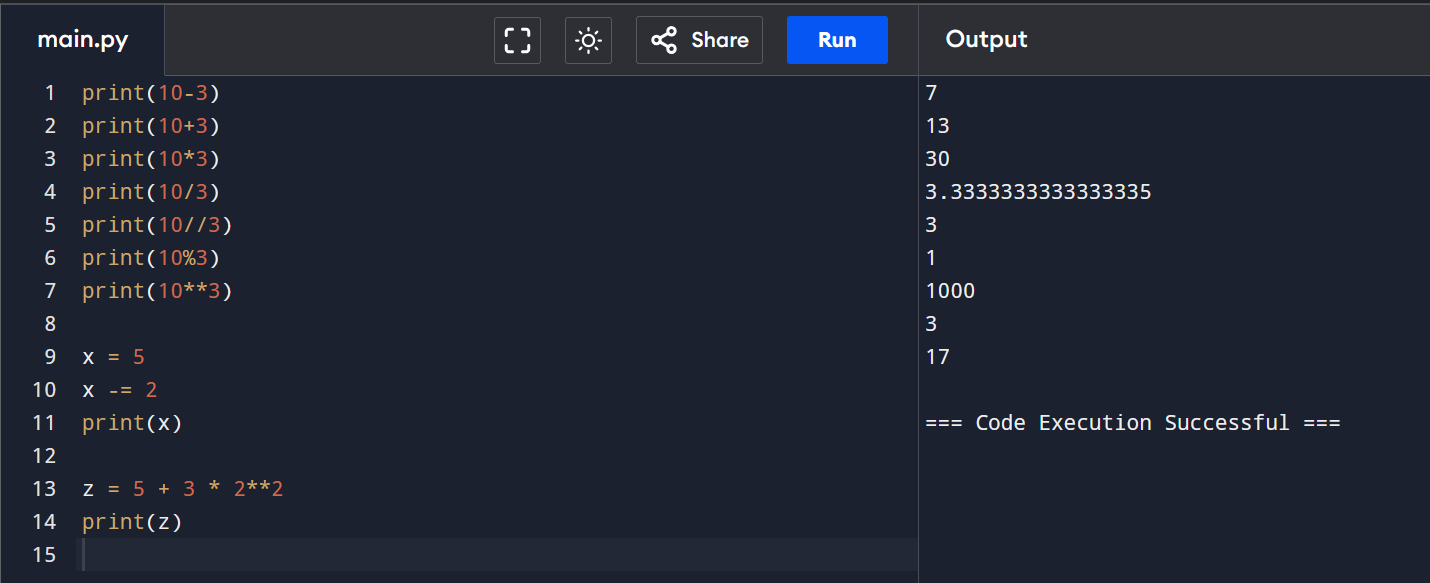


The output of lesson7.py

In the Lesson-08 code work like:

In this lesson, we learned about math operations in python.

We learned about addition, subtraction, division, multiplication, modulus and power of a number and how to get them is python.

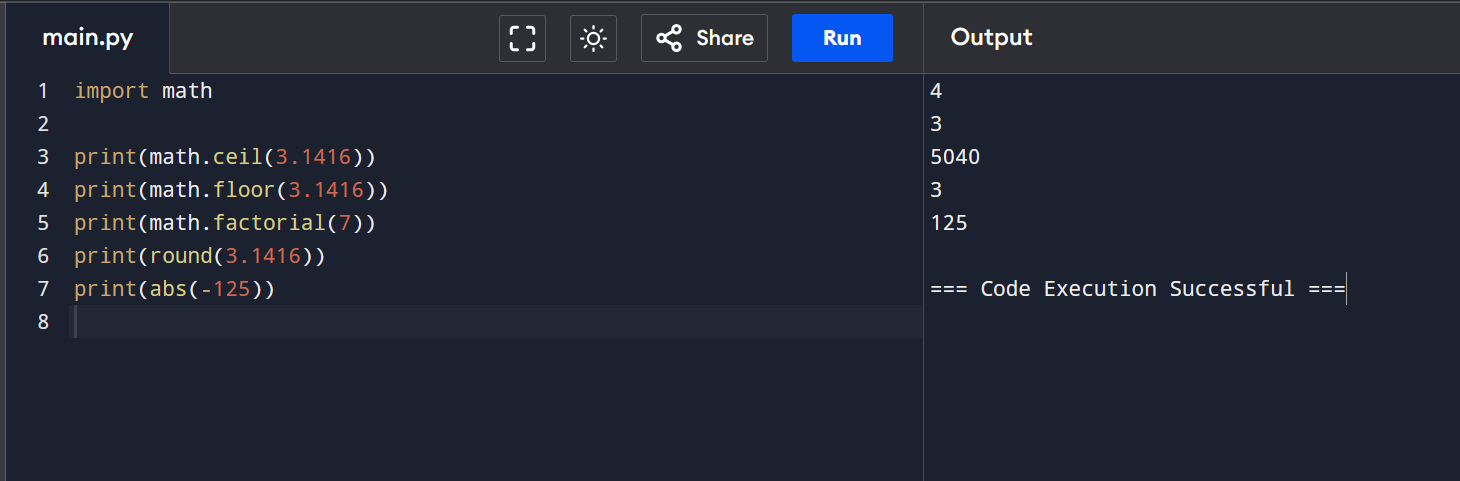
The output of lesson8.py

In the Lesson-09 code work like:

In this lesson, we learned about how to import other module inside our code and use them to solve questions.

We Imported the math module and used ceil() to get the ceiling and floor() to get the floor of a number. We used the factorial method of the module math to get the factorial of 5.

We also used the functions named round() and abs() to get the integer from a float and the absolute value of a number.

The output of lesson9.py

In the Lesson-10 code work like:

In this lesson, we learned about if else statement.

We created two variables named is\_rainy and is\_sunny and used them in if-else statement.

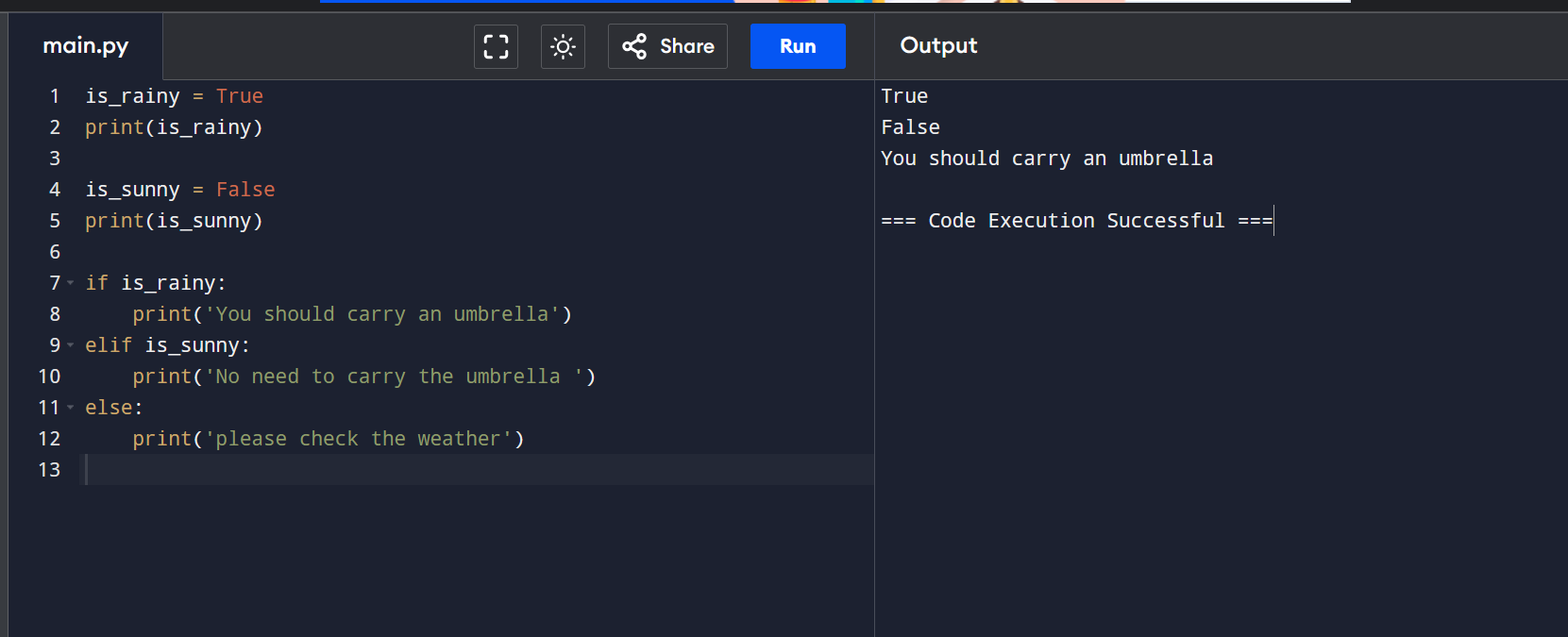
It works following way:

if followed by a condition

elif followed by a condition

else

If the condition after if is true then the code inside the if will be executed and elif and else will be ignored. But if the condition after if is not true and the condition after elif is true then the block of code inside the elif will execute. But if all condition is false then the block inside else will execute.



The output of lesson10.py

In the Lesson-11 code work like:

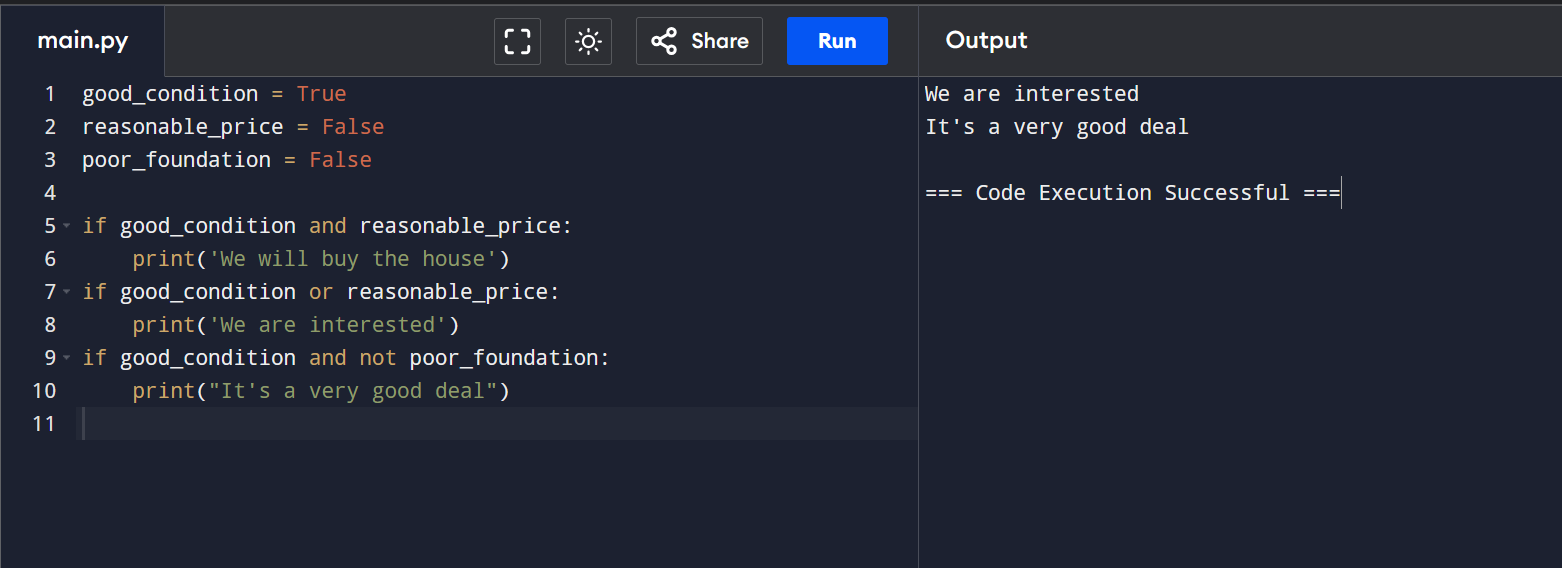
In this lesson, we learned about logical operators.

We can use them to generate a binary value.

and = logical and

or = logical or

not = logical not



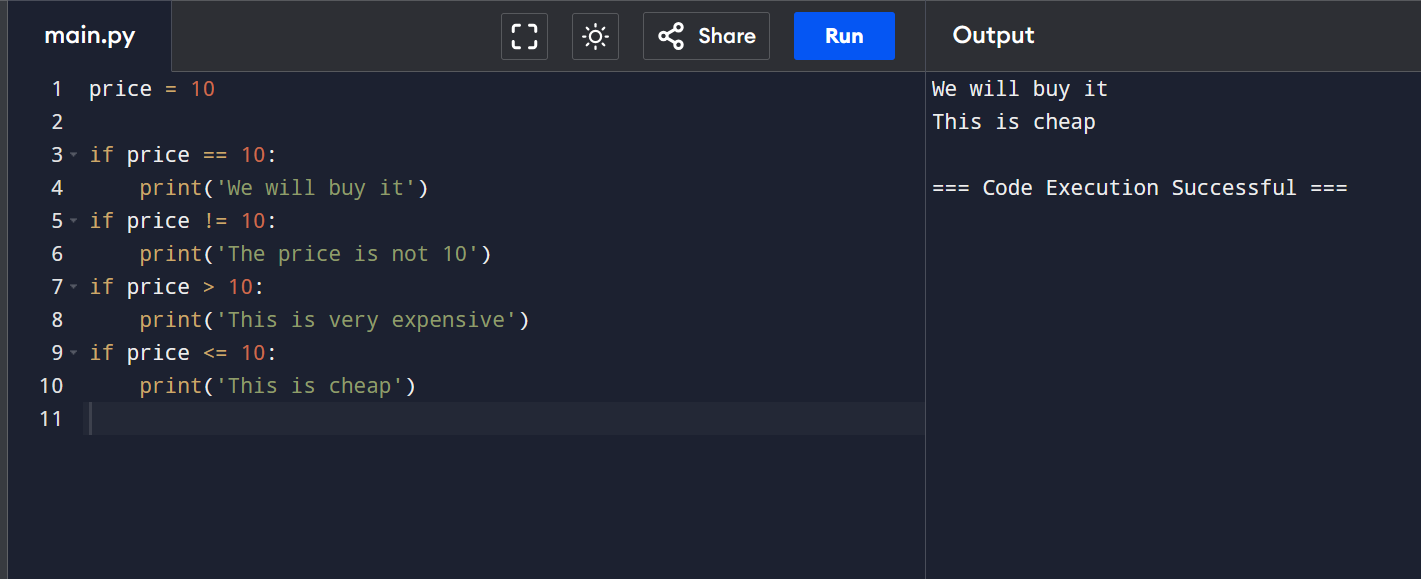
The output of lesson11.py

In the Lesson-12 code work like:

In this lesson, we learned about comparison operators.

We declared a variable named price and initialize it to 10 .

Then we used the comparison operators to compare the price to make a decision.



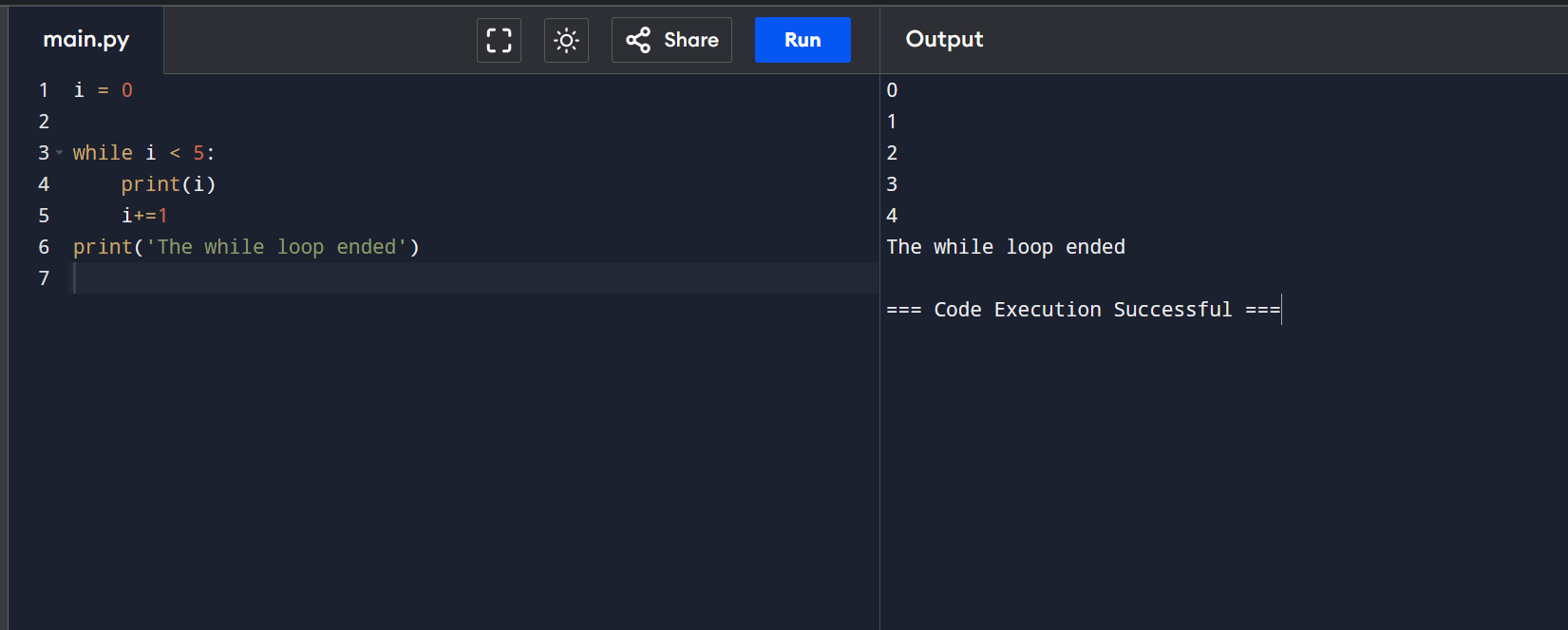
The output of lesson12.py

In the Lesson-13 code work like:

In this lesson, We learned about while loop. We declared a variable named i and initialized it to 0.

While i<5 will check to see if the value of i is greater then 5. If not, it will continue to execute the block inside the loop.

Inside the loop we incremented i by 1 so the loop might end. Else we would have created a infinite loop.



The output of lesson13.py

In the Lesson-14 code work like:

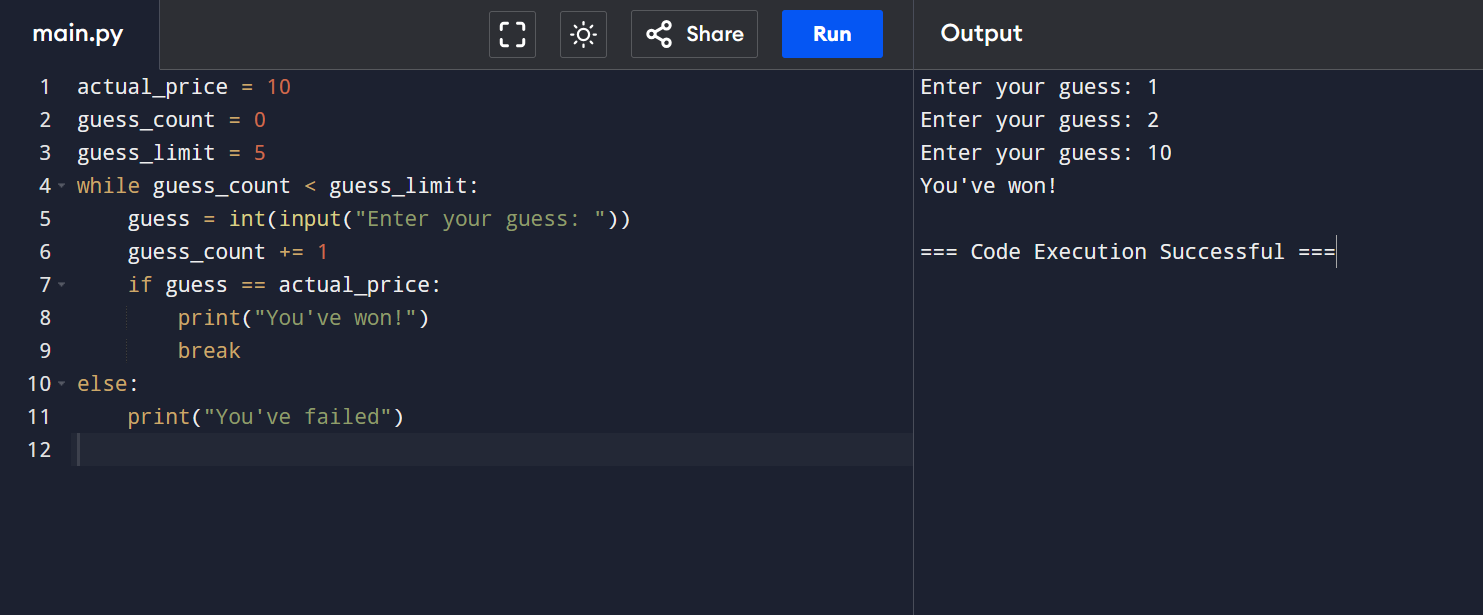
In this lesson, we created a guessing game using python.

We have 3 variables to keep track of the game.

We want to give user 5 try to guess the value of the actual\_price. That’s why, we used while loop. It loop until a certain condition becomes false.

In our case, the condition is guess\_count<guess\_limit.

Then if the user is able to guess the number before the given tries run out we printed you’ve won. And if the user fails to do so then we printed you failed.



The output of lesson14.py

In the Lesson-15 code work like:

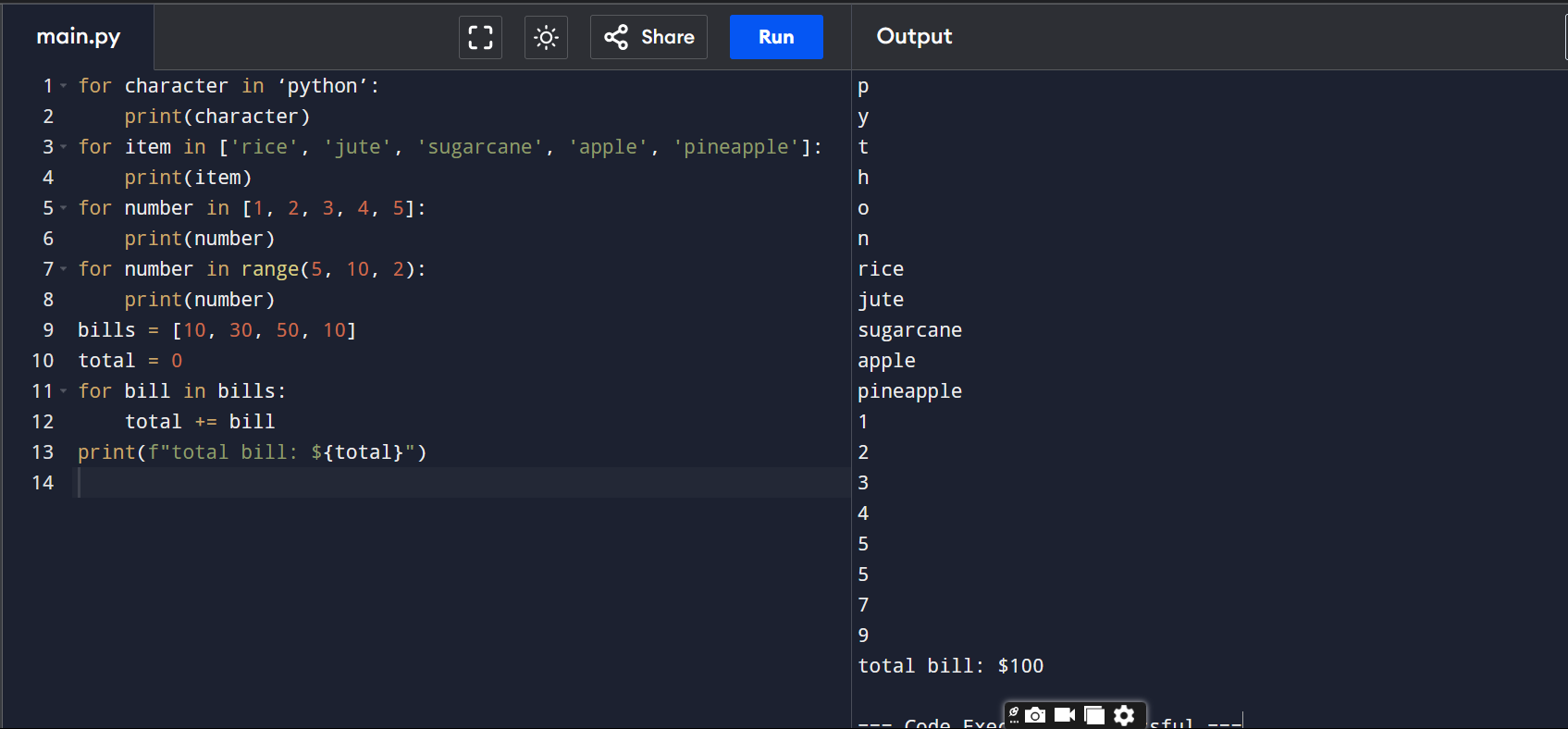
In this lesson, we learned about for loops in python and their uses.

First we printed the characters of the string “python”. We used a variable named character that will go through all characters inside the string exactly once. So, we just printed the character variable a bunch of time.

Then we created a list of strings and learned how can we can go through all the elements of that using for loop.

Then we created a list of numbers and printed all the numbers.

Then we created another list of numbers and calculated the sum of them using for loop.

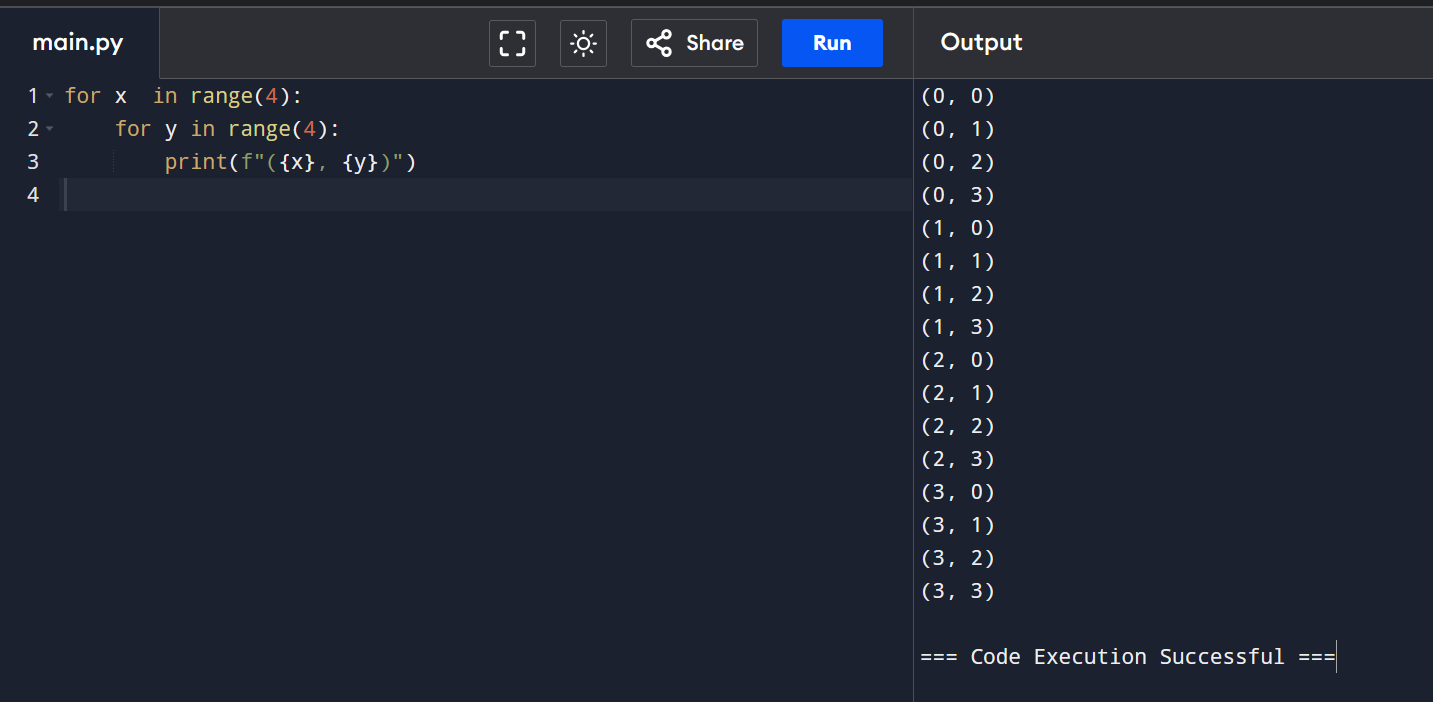


The output of lesson15.py

In the Lesson-16 code work like:

In this very short and simple lesson, we learned about how we can create a nested for loop in python.

We printed every point of a graph that has the value of x and y from 0 to 4.



The output of lesson16.py

In the Lesson-17 code work like:

In this lesson, we learned about lists in python. How can we declare and store any value inside of list.

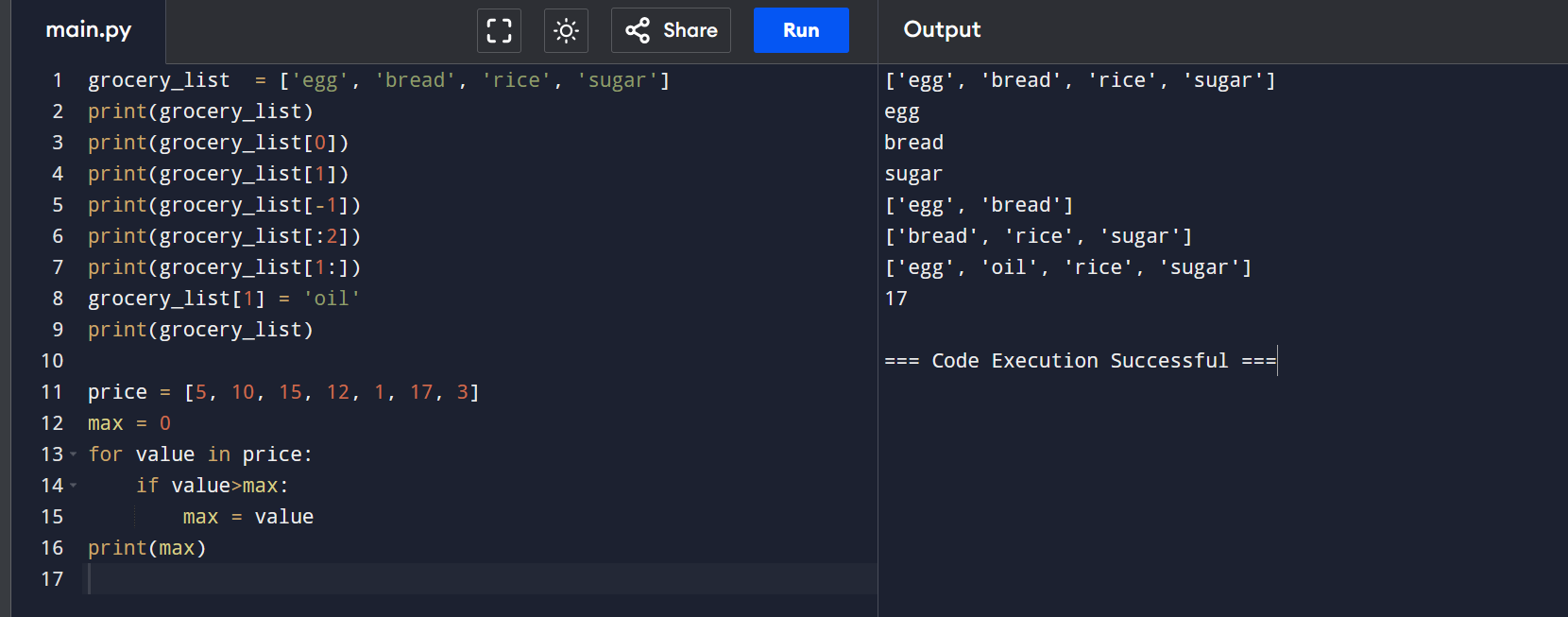
We created a list named grocery\_list and stored some strings.

Then we printed the list.

And then we learned about how we can access the elements using indexes and print them.

We also learned about how can we change any elements inside the list.

Then we created a list of numbers and used for loop to find and print the largest value of the list.



The output of lesson17.py

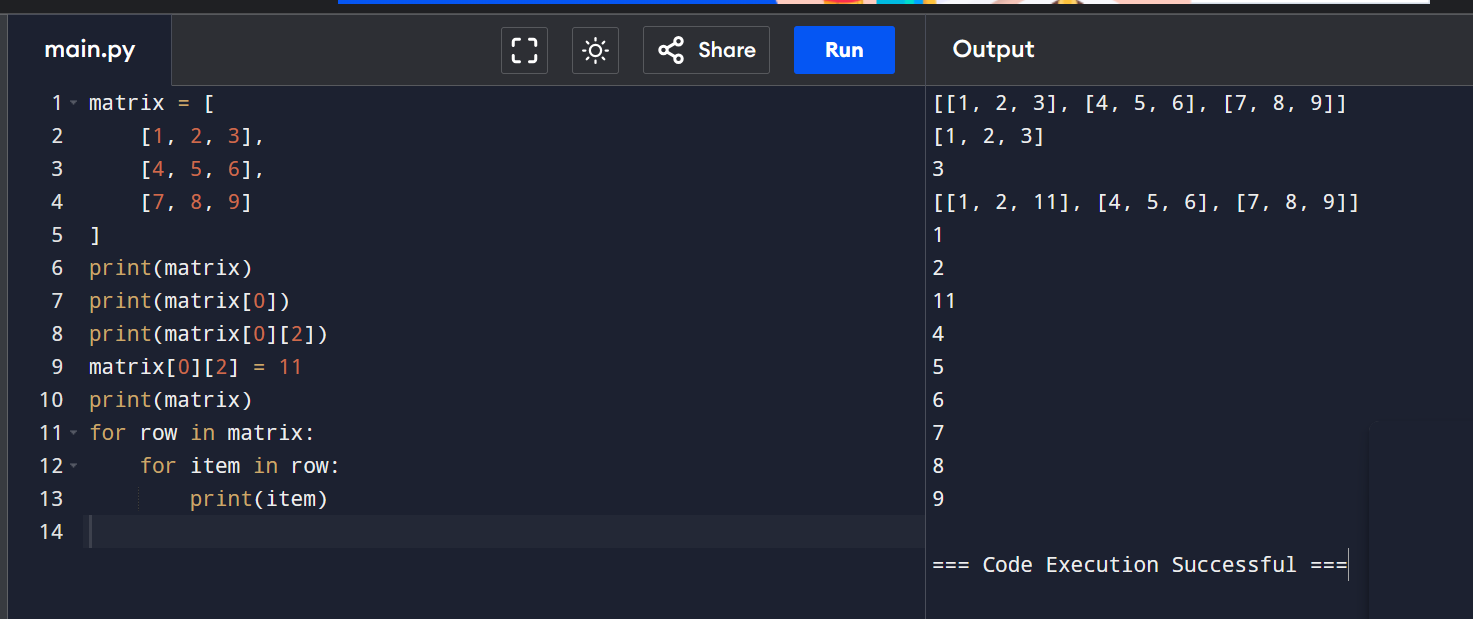
In the Lesson-18 code work like:

In this lesson, we learned about 2D lists. The 2D lists are list inside of a list.

We can use it to declare/store matrixes inside our code.

Here we stored a matrix inside a 2D list, then printed it using the print(matrix) statement.

Then we learned how to print a row and how to access/print a number/point of the matrix.

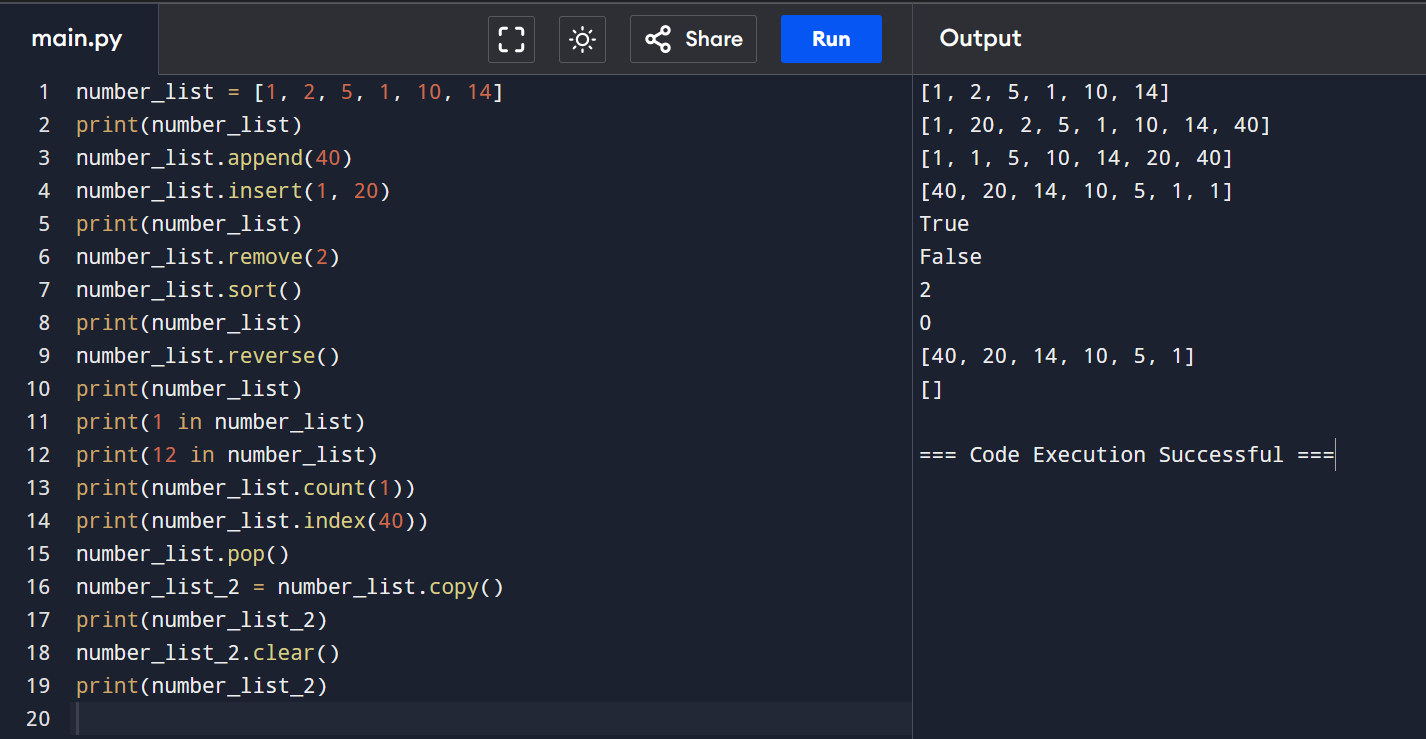


The output of lesson18.py

In the Lesson-19 code work like:

In this experiment, we learned about some methods of the list.

We learned about how to add element to a list using append(), how to insert at some index using insert(), how to remove an element using remove(), how to sort the list using sort(), how to reverse the list using reverse(). We learned how to verify a number exists in list using (element\_name in list\_name). We also learned about the uses of count(), index(), clear().



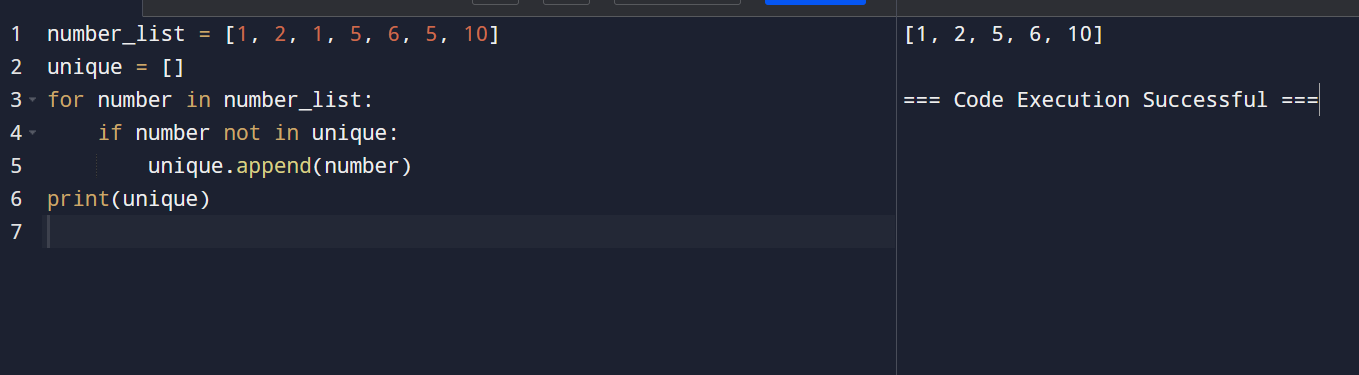
The output of lesson19.py

In the Lesson-20 code work like:

In this experiment, we learned how to create a unique list from a list just using for loop.

We first declared the number\_list consisting duplicate numbers.

Then we used for loop to iterate through all the element and then added first occurrence of a number into the unique list.

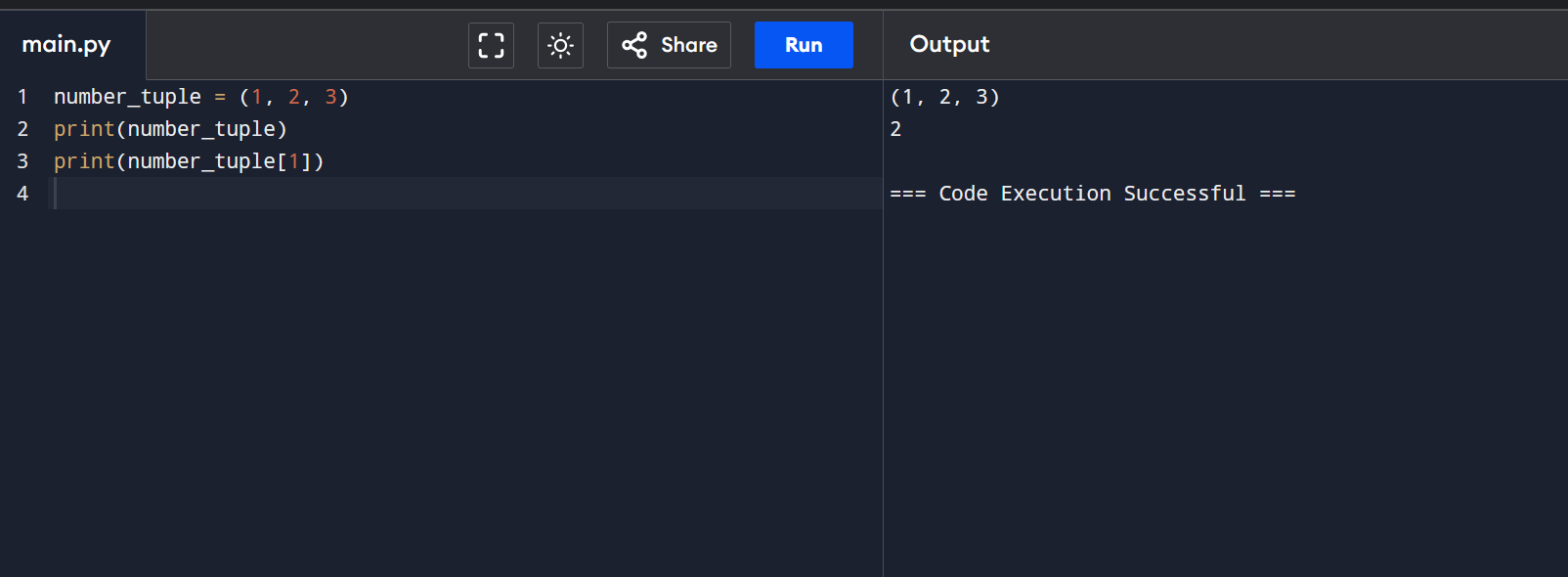


The output of lesson20.py

In the Lesson-21 code work like:

In this lesson, we learned about tuples and the difference between tuple and list.

The main difference is that we can change the elements of a list but we can not change the elements of a tuple. But the tuple is faster.



The output of lesson21.py

In the Lesson-22 code work like:

In this lesson, we learned about the fastest/easiest way to add the elements of a tuple or list.

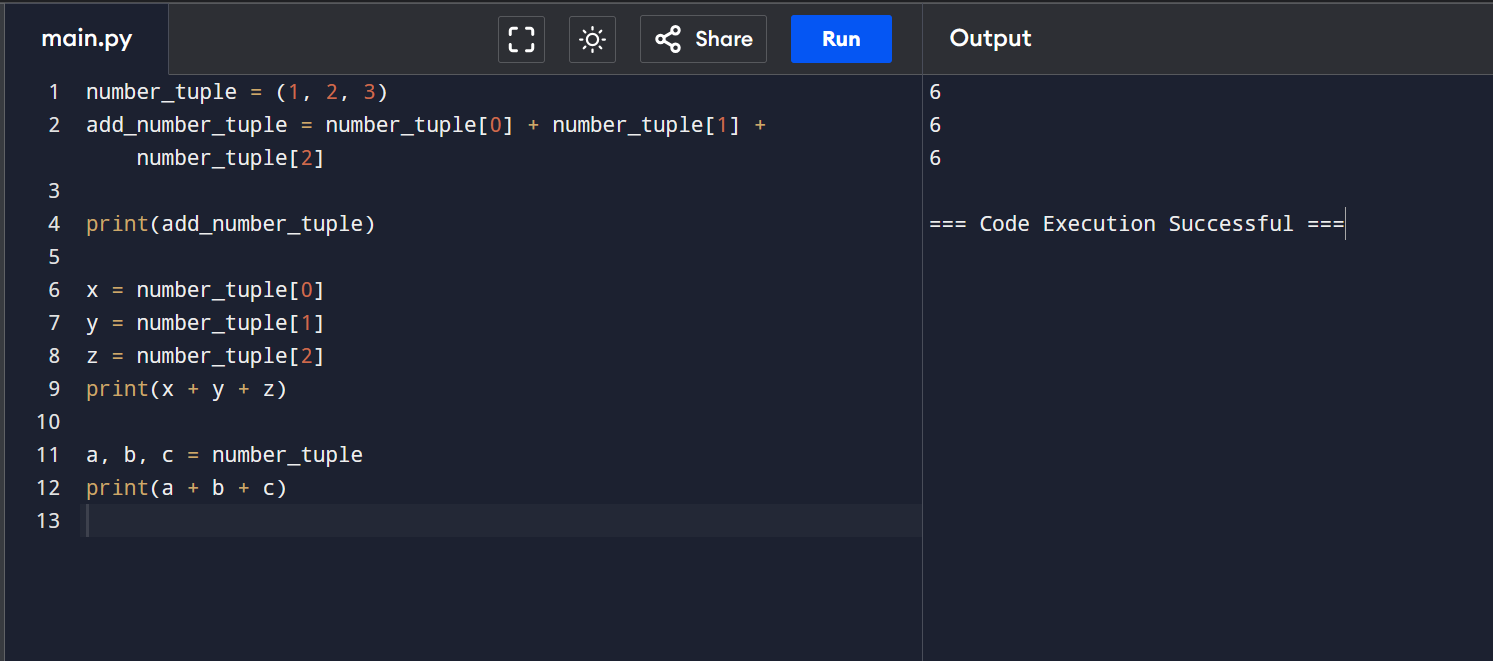
Instead of, x = list[0]

y = list[1]

We can do

x, y = list.

And it will the store the values of list [0] in first variable and list[1] in the second variable.

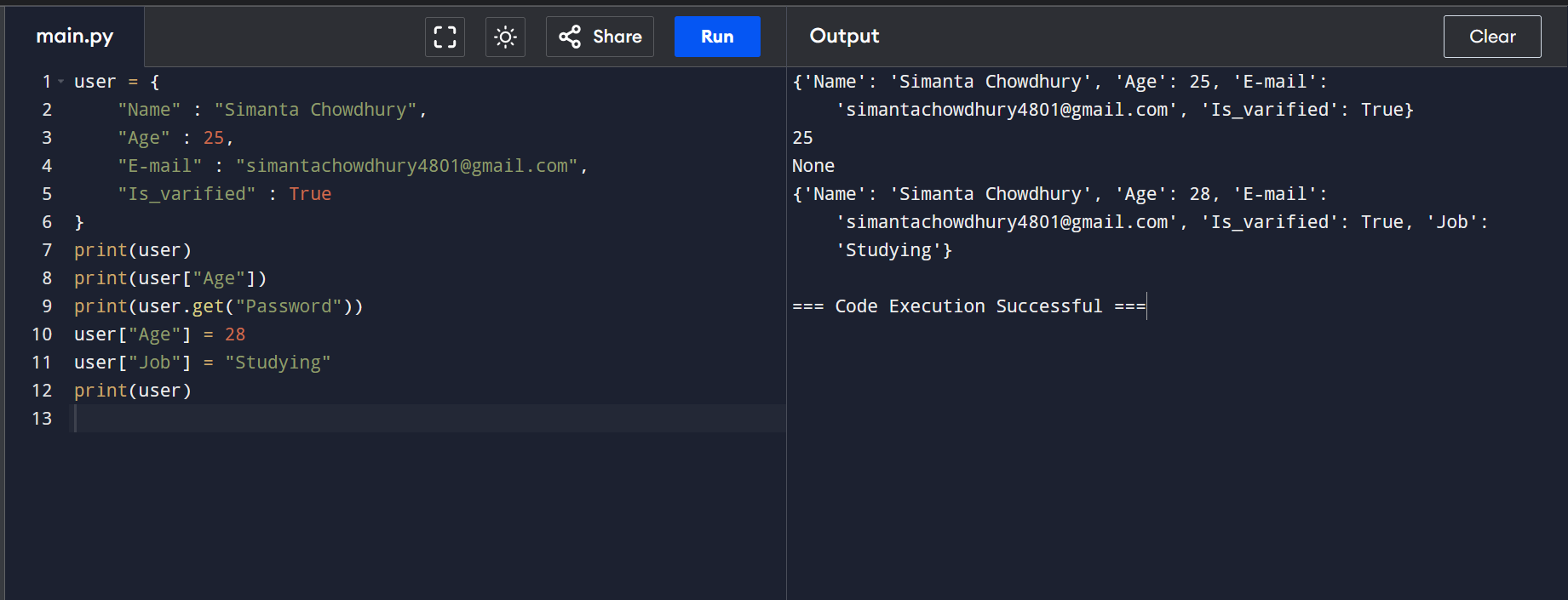


The output of lesson22.py

In the Lesson-23 code work like:

In this lesson, we learned about dictionary.

We can store key value pair using the dictionary. We learned about how to use dictionary efficiently and how can we access/change the elements of a dictionary.



The output of lesson23.py

In the Lesson-24 code work like:

In this lesson, we learned about how we can create a emoji converter in python using the dictionary.

First, we stored the string we want to change in a variable named message.

Then, we separated the sentences into individual words using split(‘ ‘) and stored that in separate\_words list.

Then we created a dictionary that consists the changes of emojis.

Then we created a output variable and filled it with the sentence using for loop and changed where we found an emoji.



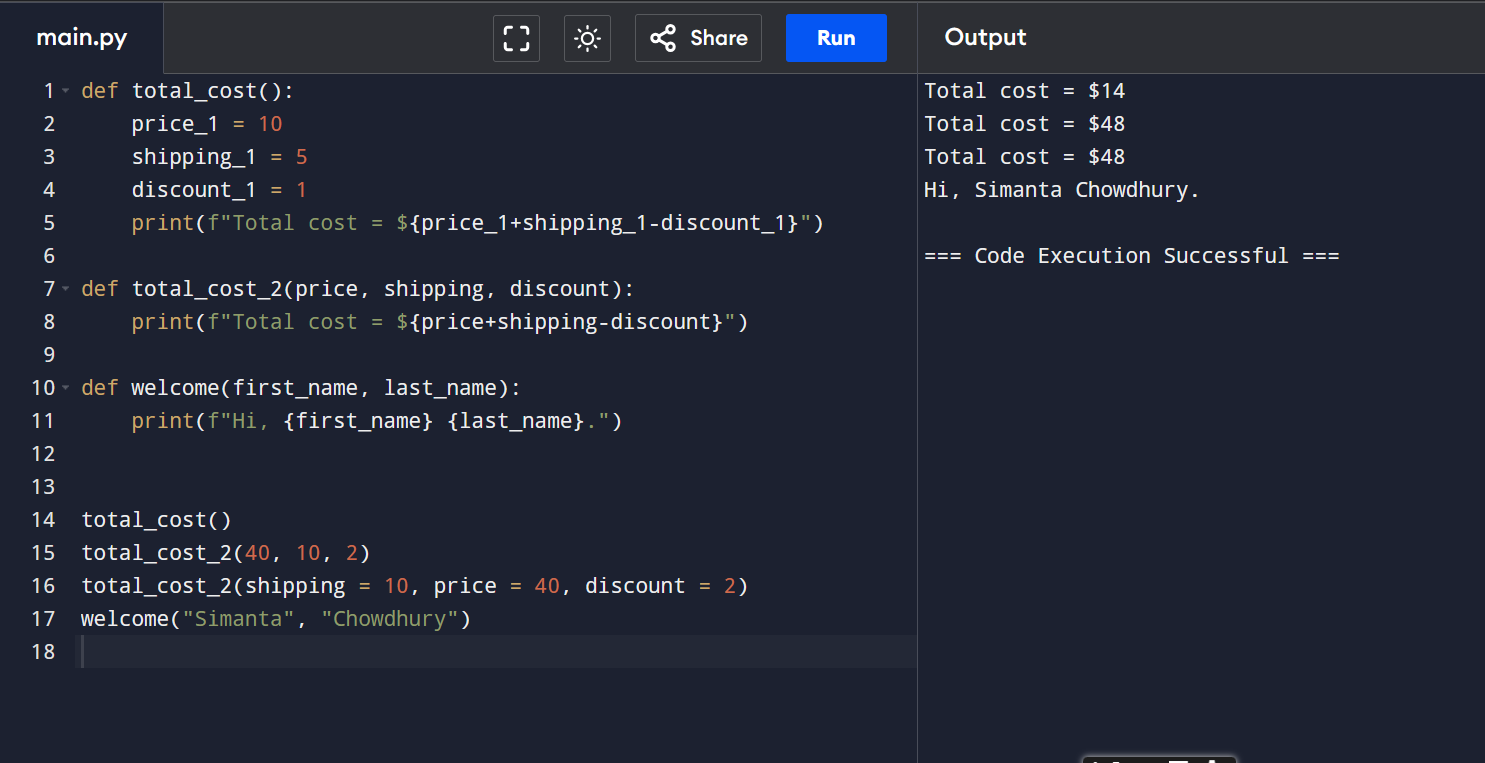
The output of lesson24.py

In the Lesson-25 code work like:

In this experiment, we learned about functions and how they work.

We declared two total\_cost() and total\_cost\_2(), where in the first one we need to change the function everytime we want to change the values. But in the second one we can just pass the new values as argument.

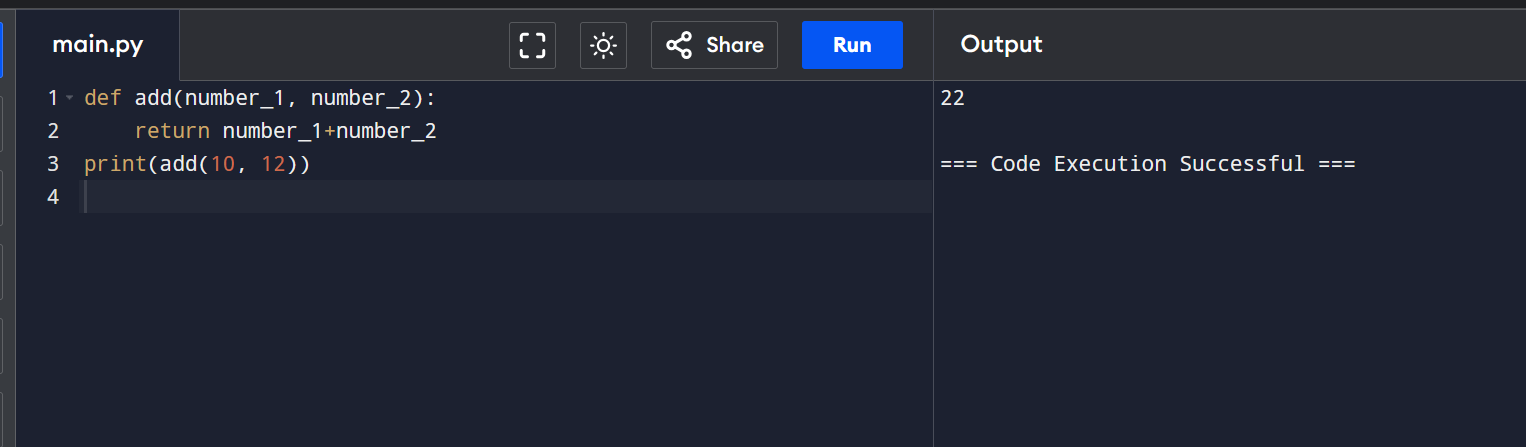
We also created a welcome function that takes the name of a person and prints hi, the\_name.



The output of lesson25.py

In the Lesson-26 code work like:

In this experiment, we created a very simple function named add that takes two number as parameter and returns the sum of the variables. We learned about how a function can return a value after running.



The output of lesson26.py

In the Lesson-27 code work like:

In this experiment, we learned about how can we make a block reusable using a function.

We used the emoji converter from lesson-24 and made a function for it. So that, we can use it whenever we want just by calling the function without needing to write the whole block of code.

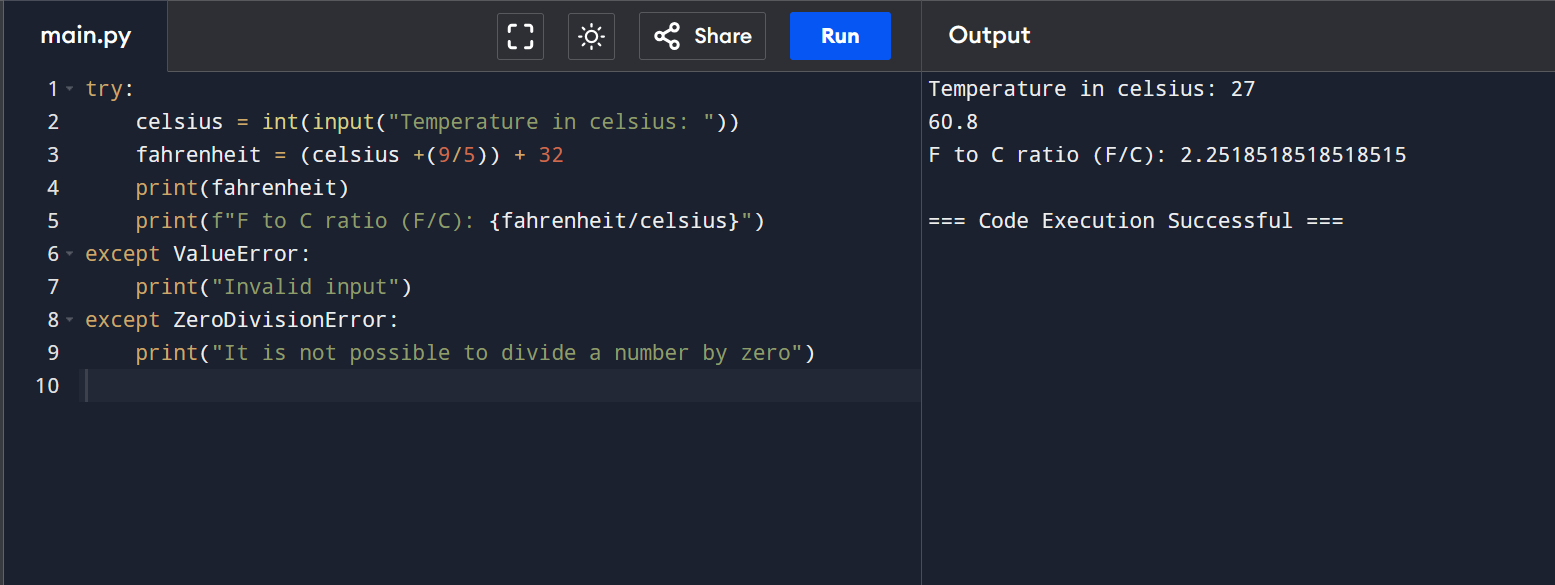


The output of lesson27.py

In the Lesson-28 code work like:

In this experiment, we learned how to handle exceptions. While creating the temperature converter in our previous lesson, we had came across some exceptions. In this lesson, we learned how we can handle them using try and exception block.

We put the code that might give an exception in the try block and in exception, we give the possible exceptions to try and catch them.



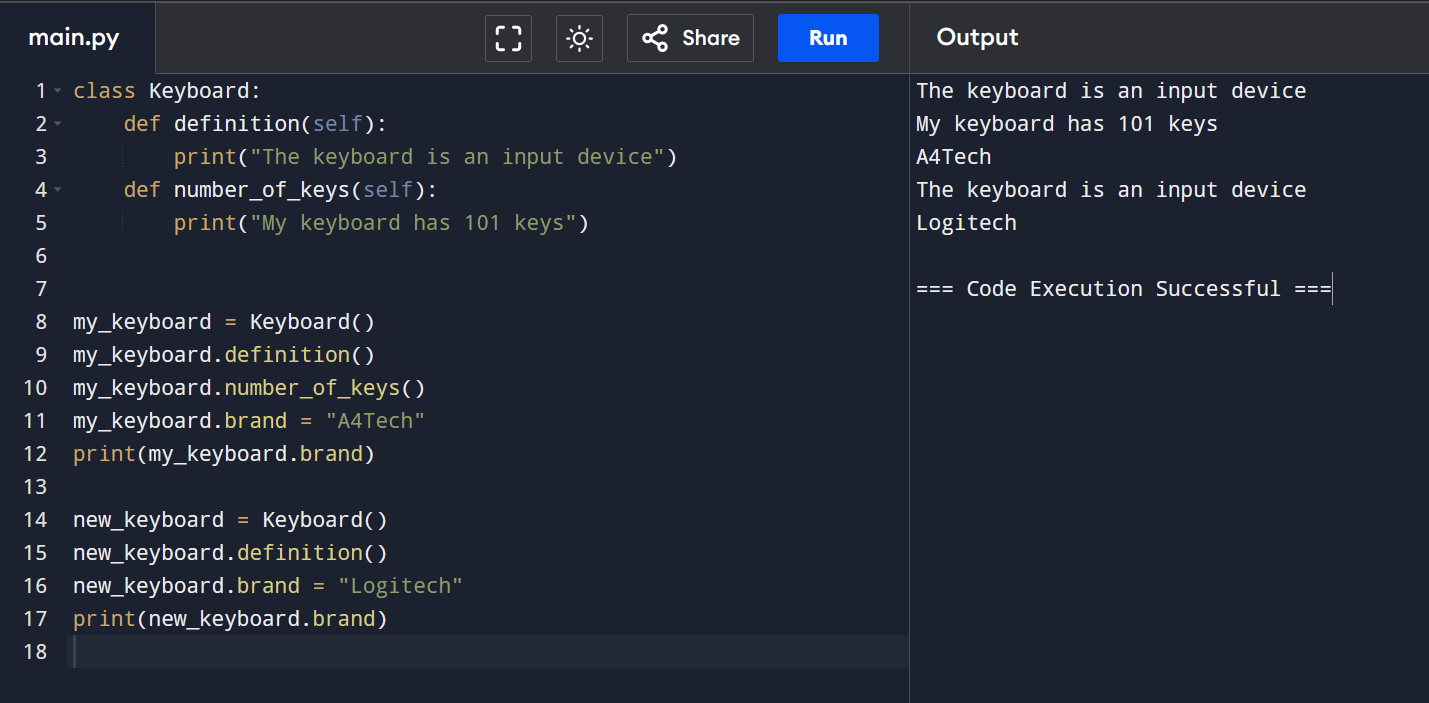
The output of lesson28.py

In the Lesson-29 code work like:

In this lesson, we learned about class. How can we create a class and the uses of a class.

A class can contain some methods and some attributes.

We created a class called Keyboard and the created two objects of the class Keyboard and called the methods of the Keyboard using them.



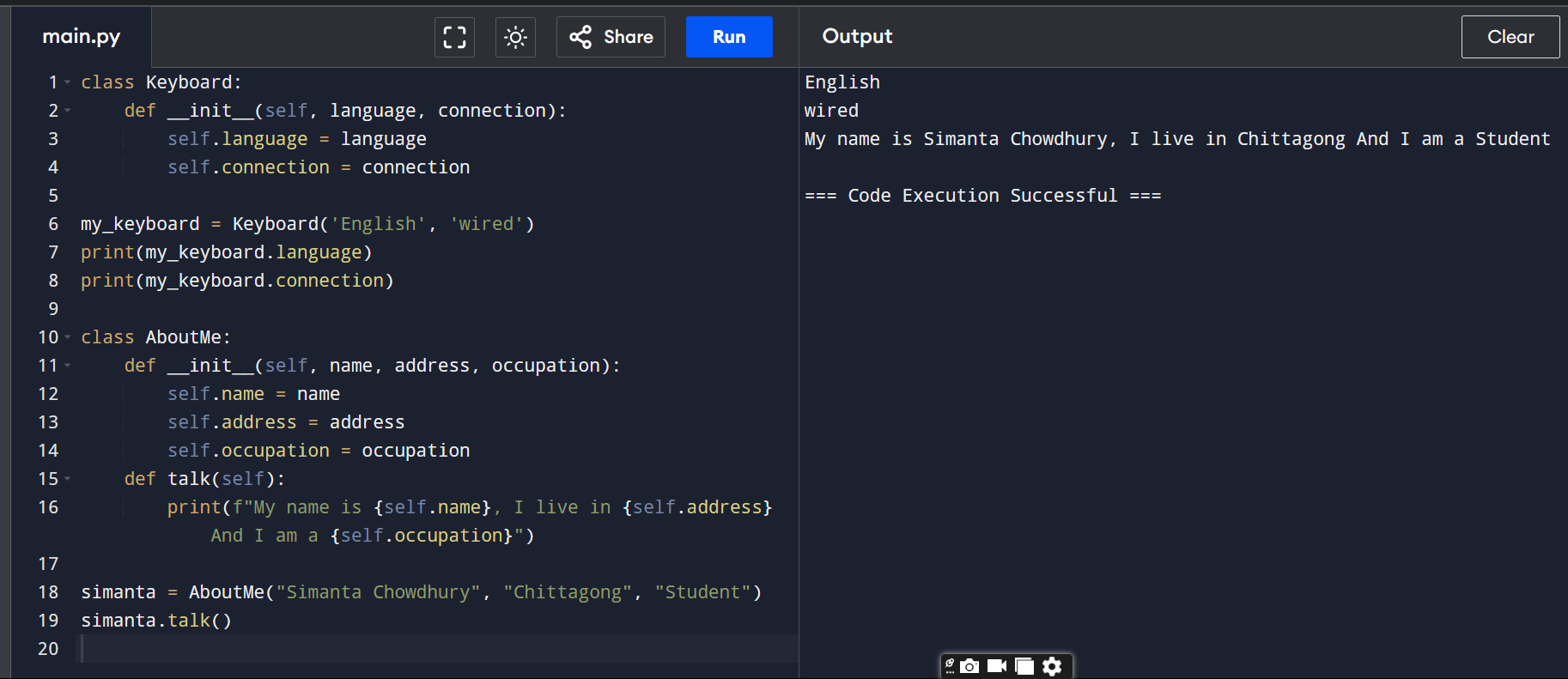
The output of lesson29.py

In the Lesson-30 code work like:

In this lesson, we learned about the constructors in python. A constructor has the same name as the class but has parentheses like a method,

Keyboard() -> is a constructor

When we create an object of any class, we call the constructor of that class.



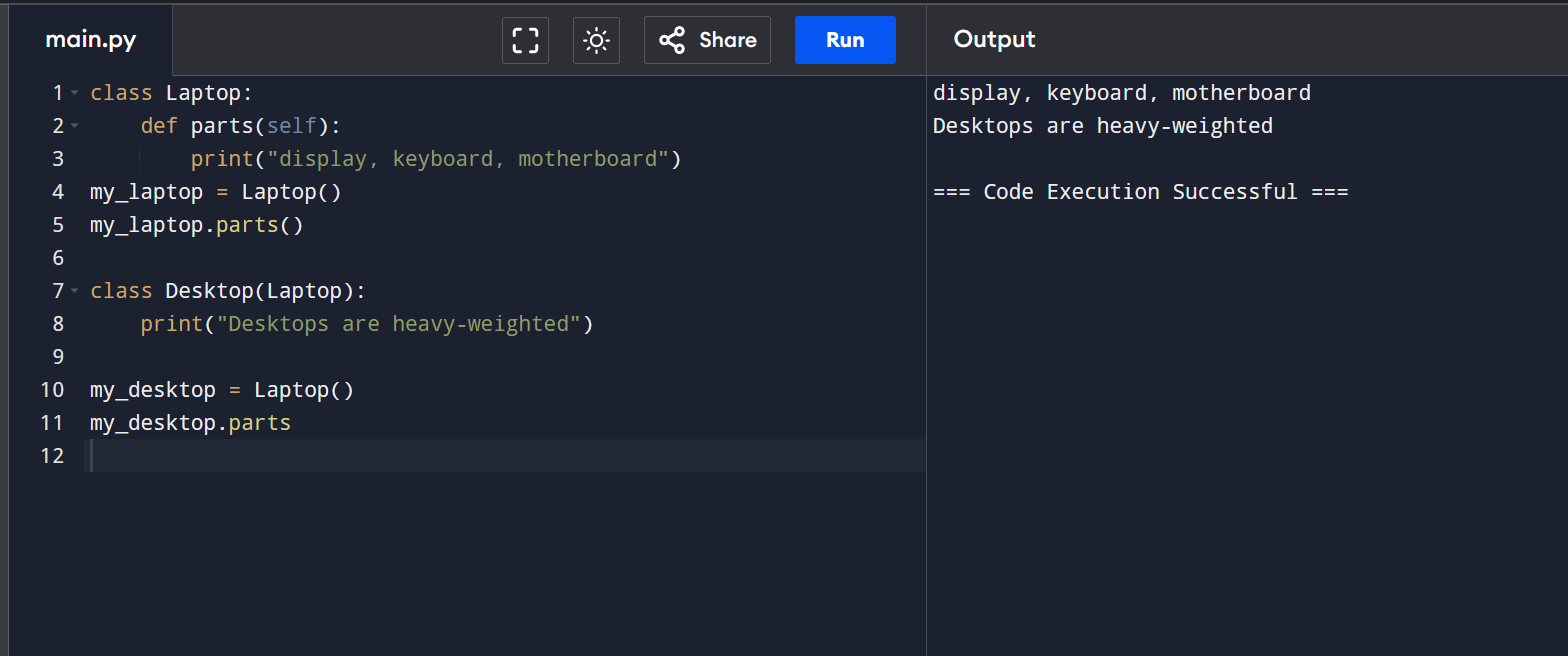
The output of lesson30.py

In the Lesson-31 code work like:

In this lesson, we learned about inheritance.

Any child class can access the methods and variables of any parent class.

Like in this example, the Desktop class is a child class of the class Laptop. So even though it doesn’t have the parts method in it, it can access the parts method of the class Laptop. We can also create it’s own methods and variables.



The output of lesson31.py