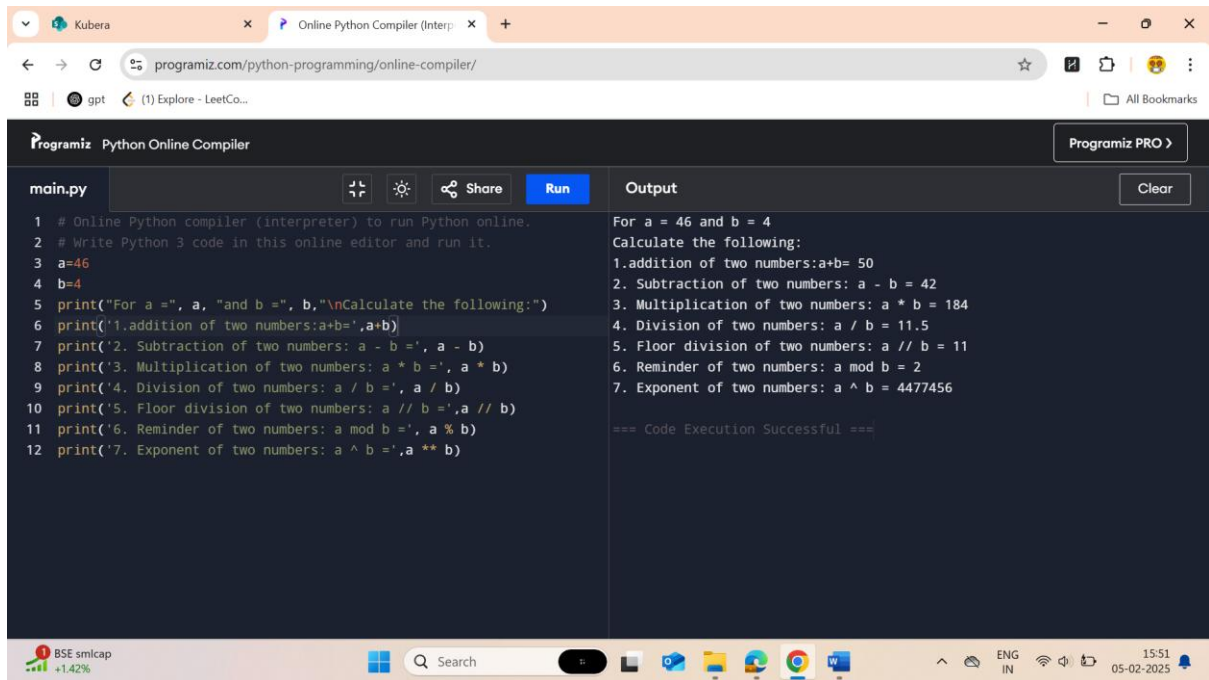


# Python Basics

## Python operators:

Example1:Arithmetic operations:



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a Python script that defines variables `a=46` and `b=4`, and then prints seven arithmetic operations. The output panel on the right displays the results of these operations. The browser's taskbar at the bottom shows the system clock as 15:51 on 05-02-2025.

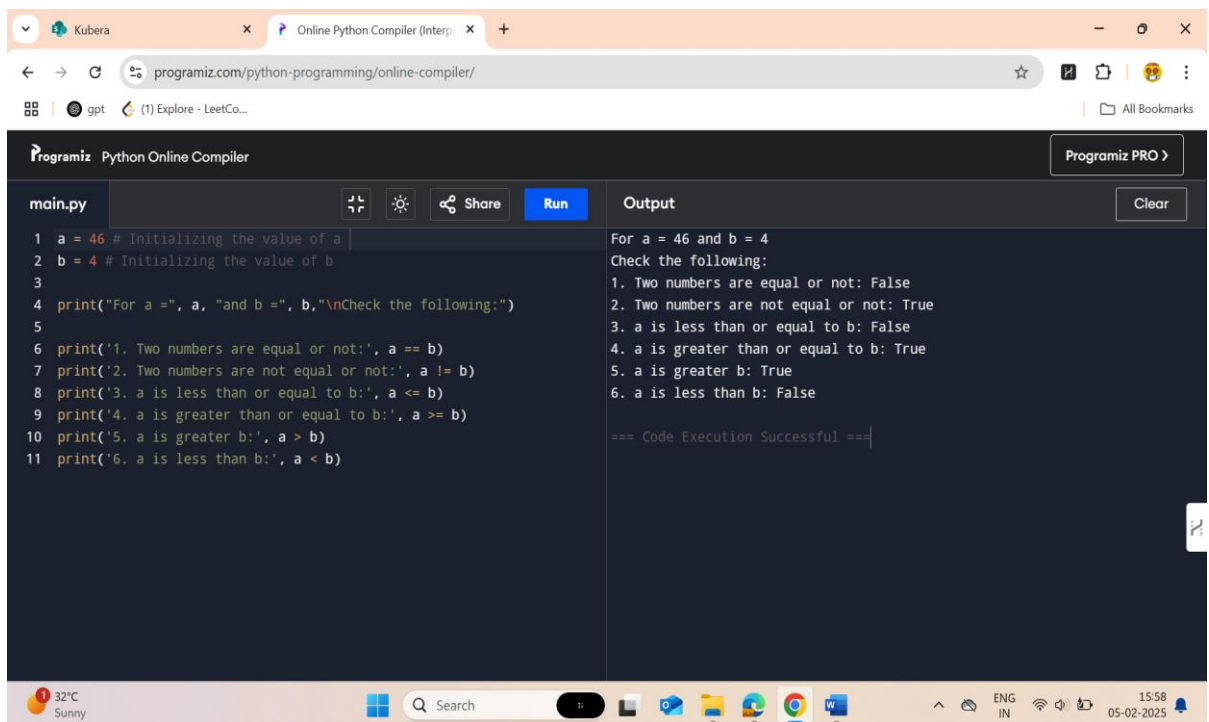
```
main.py
1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 a=46
4 b=4
5 print("For a =", a, "and b =", b,"\nCalculate the following:")
6 print('1.addition of two numbers:a+b=',a+b)
7 print('2. Subtraction of two numbers: a - b =', a - b)
8 print('3. Multiplication of two numbers: a * b =', a * b)
9 print('4. Division of two numbers: a / b =', a / b)
10 print('5. Floor division of two numbers: a // b =',a // b)
11 print('6. Remainder of two numbers: a mod b =', a % b)
12 print('7. Exponent of two numbers: a ^ b =',a ** b)
```

Output

```
For a = 46 and b = 4
Calculate the following:
1.addition of two numbers:a+b= 50
2. Subtraction of two numbers: a - b = 42
3. Multiplication of two numbers: a * b = 184
4. Division of two numbers: a / b = 11.5
5. Floor division of two numbers: a // b = 11
6. Remainder of two numbers: a mod b = 2
7. Exponent of two numbers: a ^ b = 4477456

=== Code Execution Successful ===
```

Example 2: Comparison operators in Python.



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a Python script that defines variables `a=46` and `b=4`, and then prints six comparison operations. The output panel on the right displays the results of these operations. The browser's taskbar at the bottom shows the system clock as 15:58 on 05-02-2025.

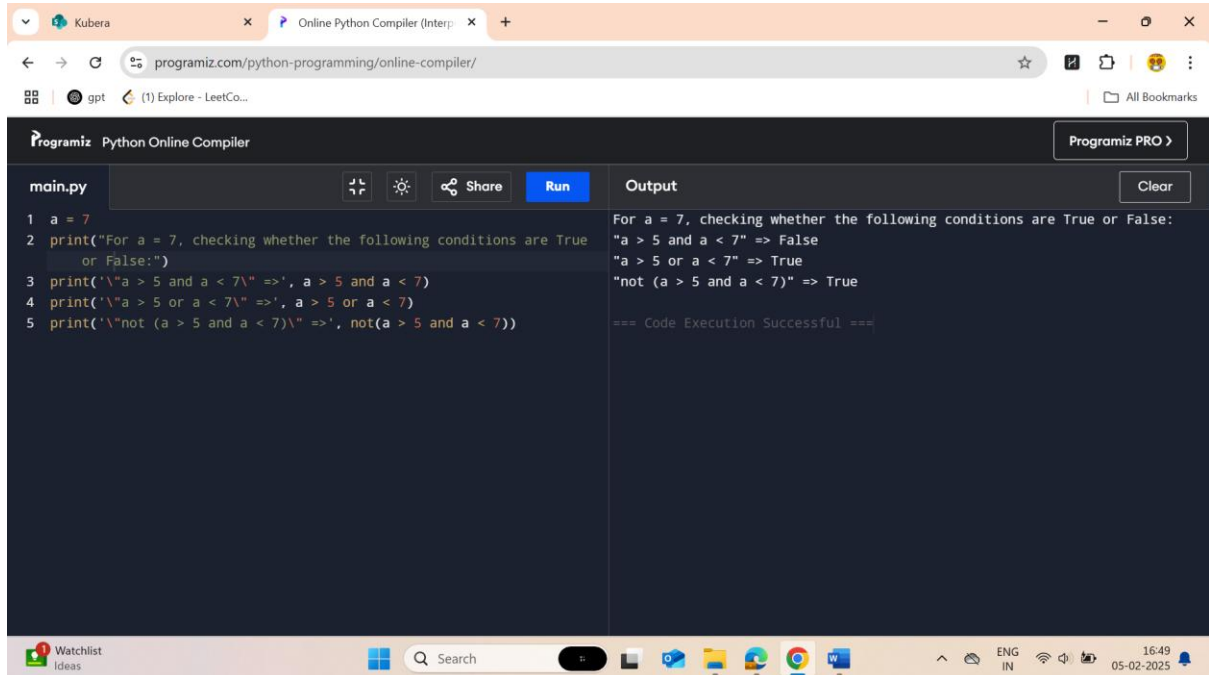
```
main.py
1 a = 46 # Initializing the value of a
2 b = 4 # Initializing the value of b
3
4 print("For a =", a, "and b =", b,"\nCheck the following:")
5
6 print('1. Two numbers are equal or not:', a == b)
7 print('2. Two numbers are not equal or not:', a != b)
8 print('3. a is less than or equal to b:', a <= b)
9 print('4. a is greater than or equal to b:', a >= b)
10 print('5. a is greater b:', a > b)
11 print('6. a is less than b:', a < b)
```

Output

```
For a = 46 and b = 4
Check the following:
1. Two numbers are equal or not: False
2. Two numbers are not equal or not: True
3. a is less than or equal to b: False
4. a is greater than or equal to b: True
5. a is greater b: True
6. a is less than b: False

=== Code Execution Successful ===
```

### Example 3: examples of Logical operators in Python



The screenshot shows the Programiz Python Online Compiler interface. The code in the editor is as follows:

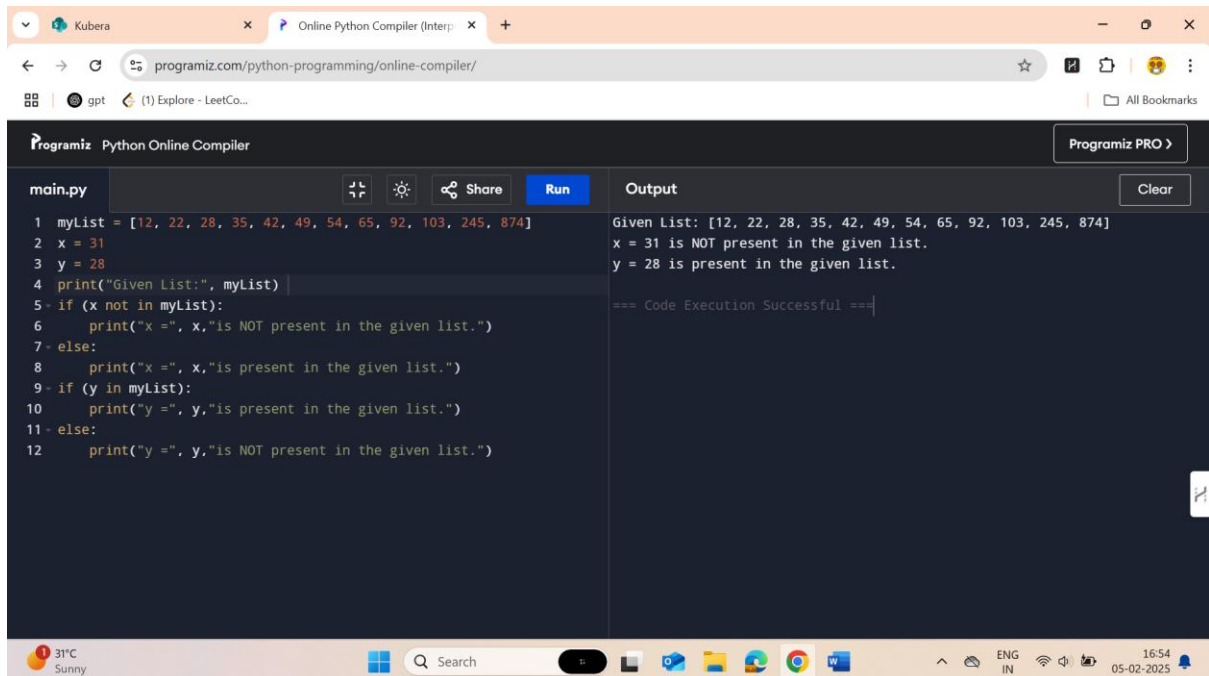
```
1 a = 7
2 print("For a = 7, checking whether the following conditions are True or False:")
3 print('\na > 5 and a < 7\' =>', a > 5 and a < 7)
4 print('\na > 5 or a < 7\' =>', a > 5 or a < 7)
5 print('\not (a > 5 and a < 7)' =>', not(a > 5 and a < 7))
```

The output on the right shows the results of these logical operations:

```
For a = 7, checking whether the following conditions are True or False:
"a > 5 and a < 7" => False
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => True

=== Code Execution Successful ===
```

### Example 4: examples of Membership operators in Python



The screenshot shows the Programiz Python Online Compiler interface. The code in the editor is as follows:

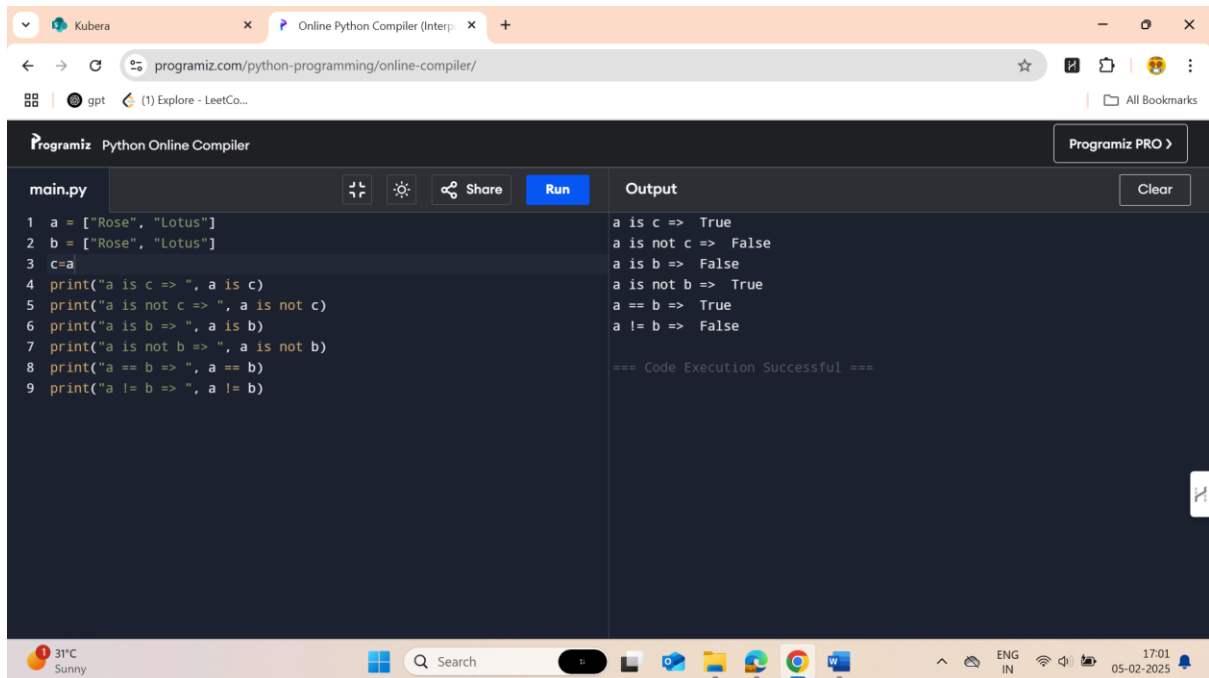
```
1 myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
2 x = 31
3 y = 28
4 print("Given List:", myList)
5 if (x not in myList):
6     print("x =", x, "is NOT present in the given list.")
7 else:
8     print("x =", x, "is present in the given list.")
9 if (y in myList):
10    print("y =", y, "is present in the given list.")
11 else:
12    print("y =", y, "is NOT present in the given list.")
```

The output on the right shows the results of these membership checks:

```
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 31 is NOT present in the given list.
y = 28 is present in the given list.

=== Code Execution Successful ===
```

## Example 5: examples of Identity operators in Python



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains the following Python code:

```
main.py
1 a = ["Rose", "Lotus"]
2 b = ["Rose", "Lotus"]
3 c=a
4 print("a is c => ", a is c)
5 print("a is not c => ", a is not c)
6 print("a is b => ", a is b)
7 print("a is not b => ", a is not b)
8 print("a == b => ", a == b)
9 print("a != b => ", a != b)
```

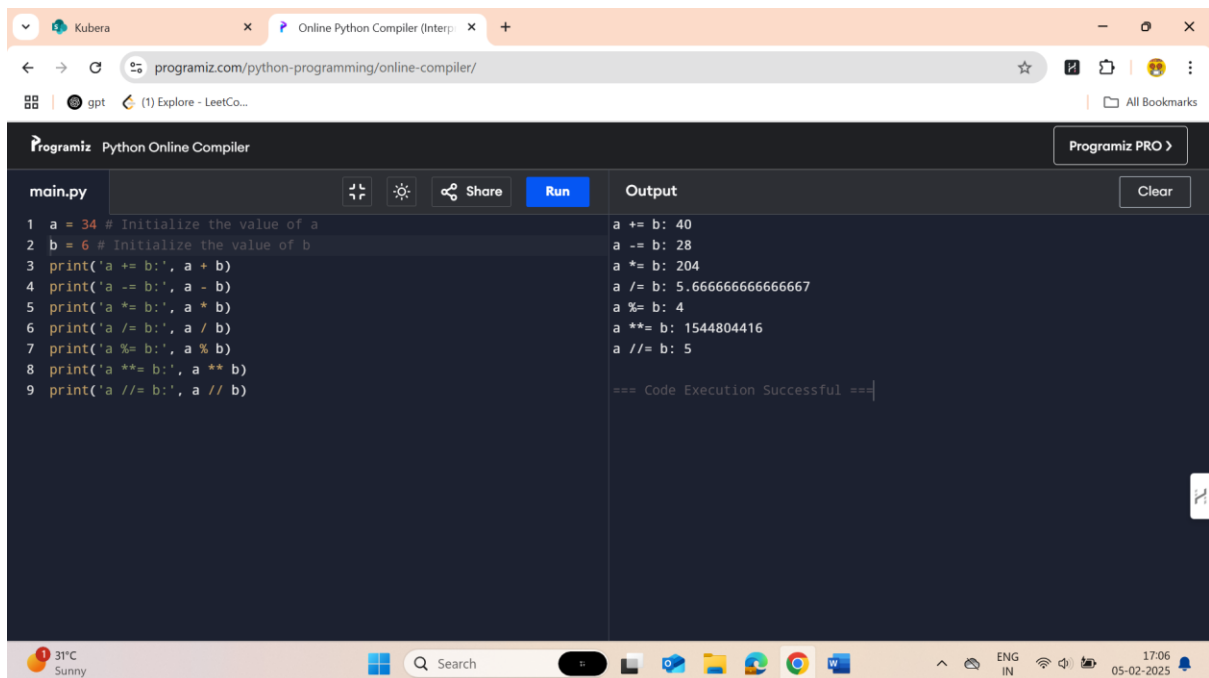
The output panel on the right displays the results of the execution:

```
a is c => True
a is not c => False
a is b => False
a is not b => True
a == b => True
a != b => False

=== Code Execution Successful ===
```

The browser's taskbar at the bottom shows the date as 05-02-2025 and the time as 17:01.

## Example 6: examples of Assignment operators in Python



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains the following Python code:

```
main.py
1 a = 34 # Initialize the value of a
2 b = 6 # Initialize the value of b
3 print('a += b:', a + b)
4 print('a -= b:', a - b)
5 print('a *= b:', a * b)
6 print('a /= b:', a / b)
7 print('a %= b:', a % b)
8 print('a **= b:', a ** b)
9 print('a //= b:', a // b)
```

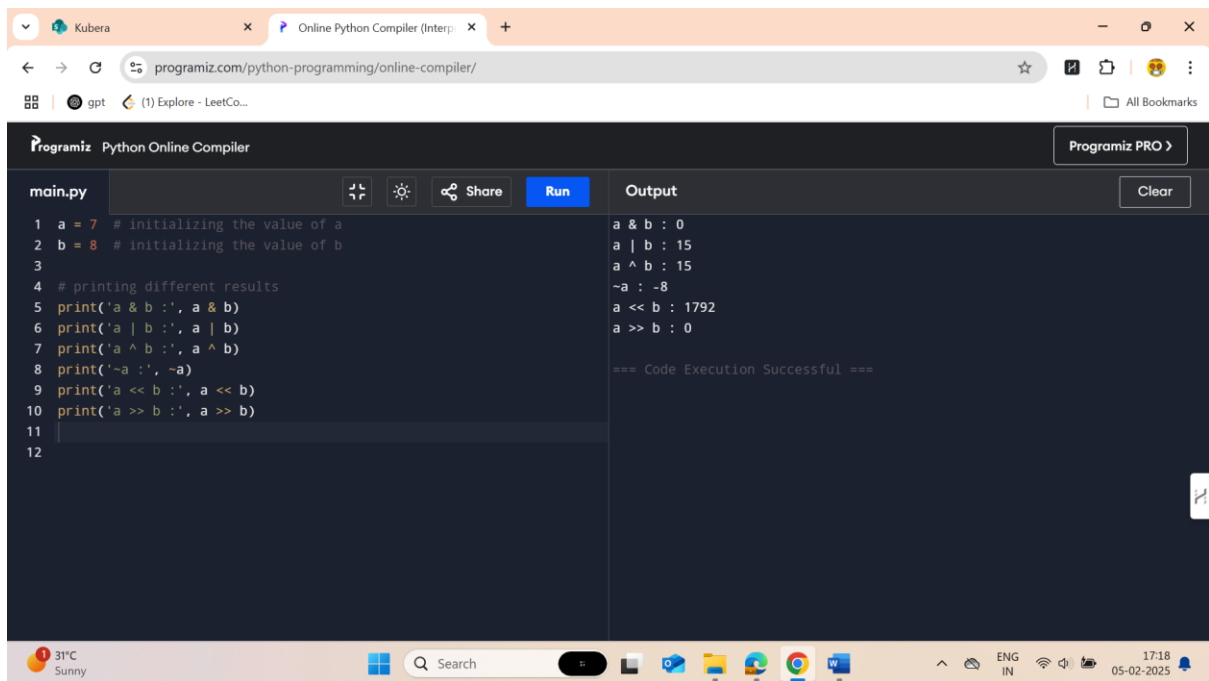
The output panel on the right displays the results of the execution:

```
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //= b: 5

=== Code Execution Successful ===
```

The browser's taskbar at the bottom shows the date as 05-02-2025 and the time as 17:06.

## Example 7: examples of Bitwise operators in Python



The screenshot shows a web browser window with the URL `programiz.com/python-programming/online-compiler/`. The page displays the Programiz Python Online Compiler interface. On the left, a code editor shows a Python script named `main.py` with the following code:

```
1 a = 7 # initializing the value of a
2 b = 8 # initializing the value of b
3
4 # printing different results
5 print('a & b :', a & b)
6 print('a | b :', a | b)
7 print('a ^ b :', a ^ b)
8 print('~a :', ~a)
9 print('a << b :', a << b)
10 print('a >> b :', a >> b)
11
12
```

On the right, the Output panel shows the results of the execution:

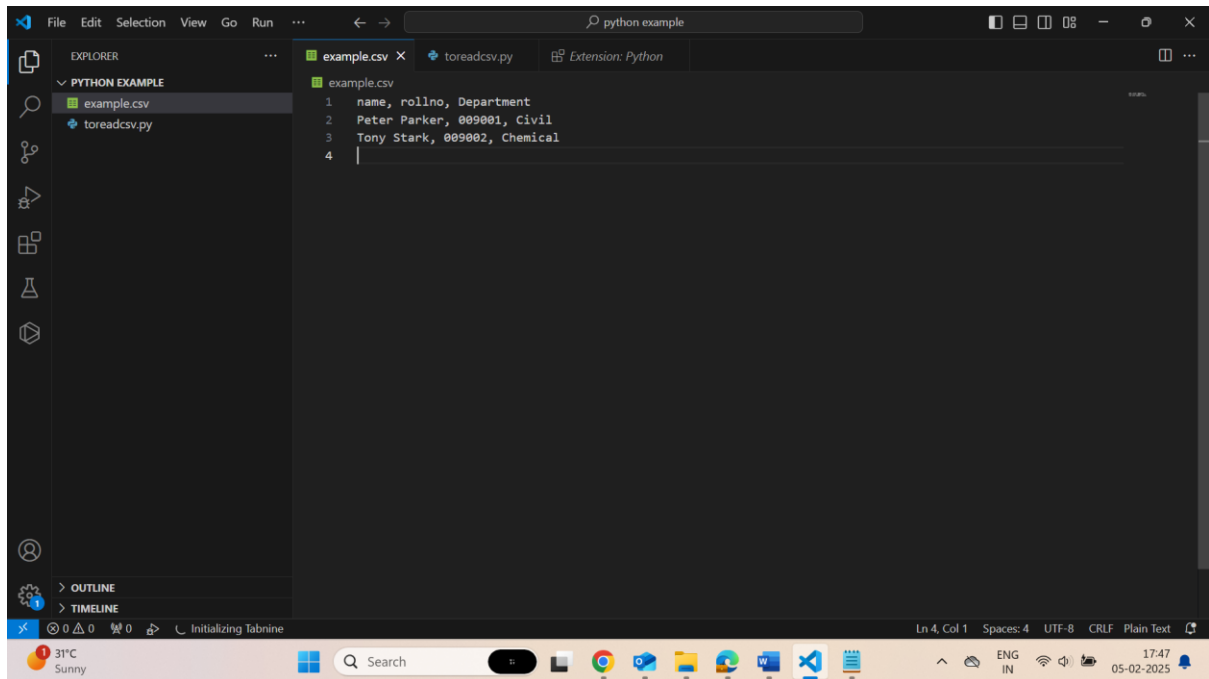
```
a & b : 0
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0

=== Code Execution Successful ===
```

The bottom of the image shows a Windows taskbar with the date and time as 05-02-2025, 17:18.

## How to read CSV file in Python?

Example –



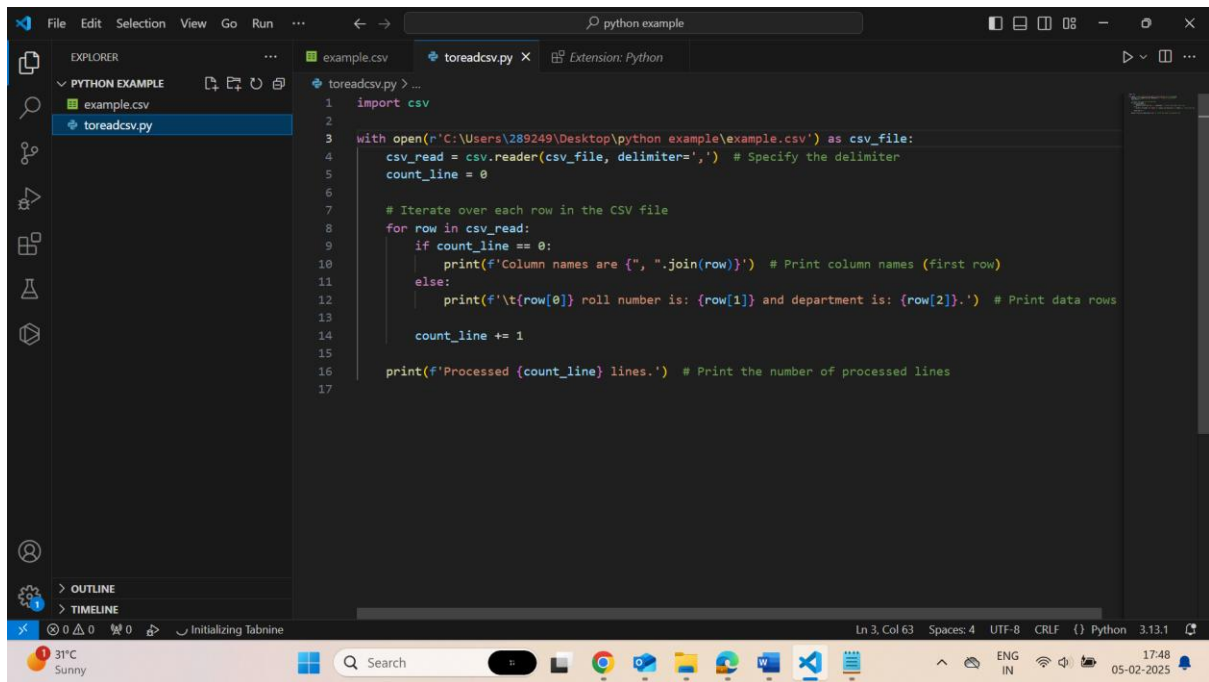
The screenshot shows a Visual Studio Code editor window with the file explorer on the left. The Explorer panel shows a folder named `PYTHON EXAMPLE` containing two files: `example.csv` and `toreadcsv.py`. The `example.csv` file is open in the editor, showing the following content:

```
1 name, rollno, Department
2 Peter Parker, 009001, Civil
3 Tony Stark, 009002, Chemical
4
```

The `toreadcsv.py` file is also open in the editor, showing the following content:

```
1 import csv
2
3 with open('example.csv', 'r') as file:
4     reader = csv.reader(file)
5     for row in reader:
6         print(row)
```

The bottom of the image shows a Windows taskbar with the date and time as 05-02-2025, 17:47.

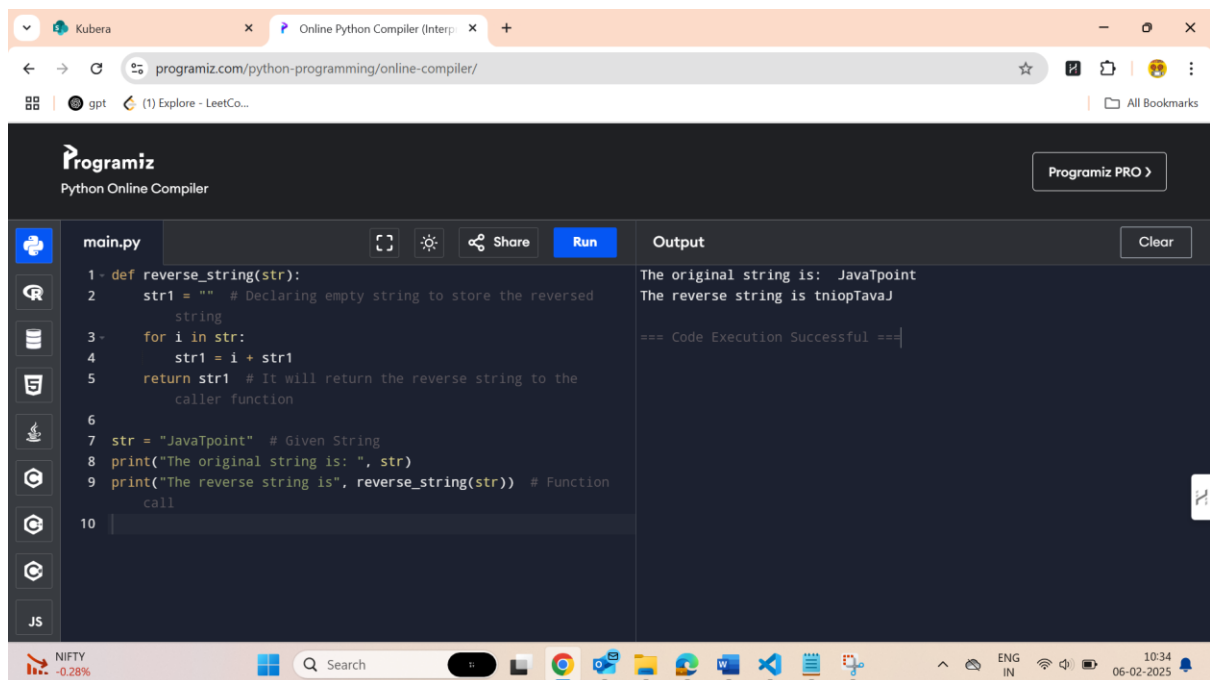


Output:

```
PS C:\Users\289249\Desktop\python example> c:; cd 'c:\Users\289249\Desktop\p
ograms\Python\Python313\python.exe' 'c:\Users\289249\.vscode\extensions\ms-py
py\adapter\..\..\debugpy\launcher' '65401' '--' 'c:\Users\289249\Desktop\pyth
Column names are name, rollno, Department
    Peter Parker roll number is: 009001 and department is: Civil.
    Tony Stark roll number is: 009002 and department is: Chemical.
Processed 3 lines.
PS C:\Users\289249\Desktop\python example>
```

# Reverse A String :

## Example 1:Using For



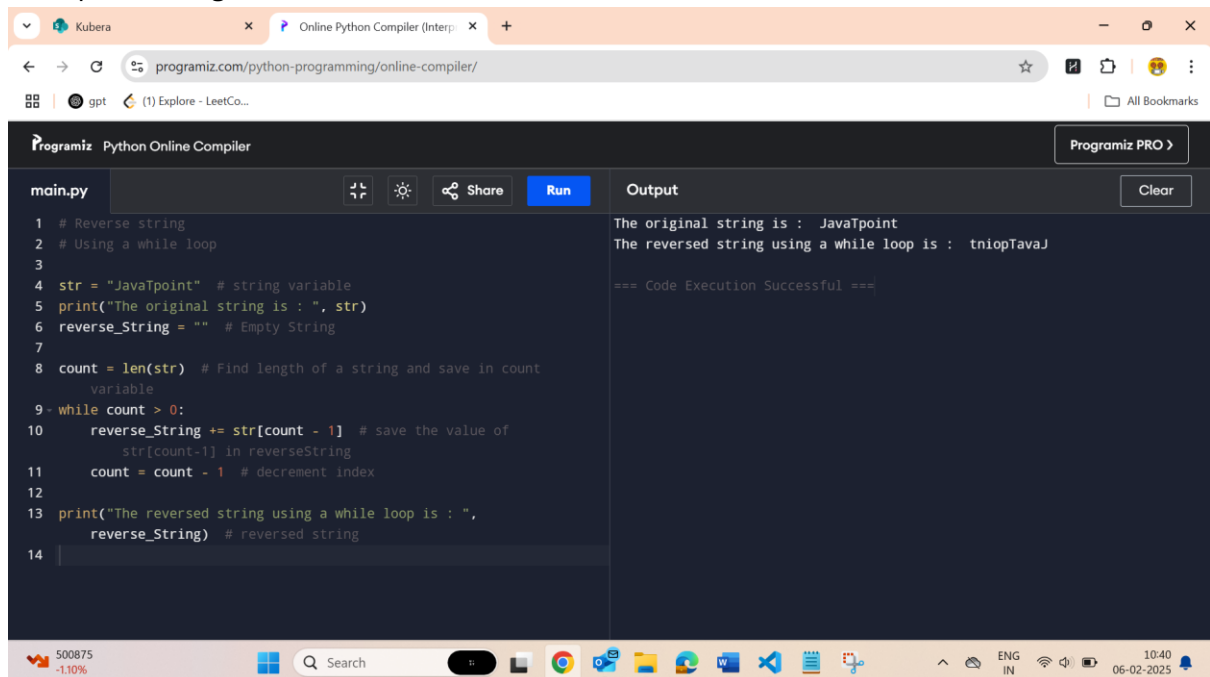
The screenshot shows the Programiz Python Online Compiler interface. The code in `main.py` defines a function `reverse_string(str)` that iterates over each character in the string and builds a reversed string. The main code calls this function with the string "JavaTpoint".

```
1- def reverse_string(str):
2     str1 = "" # Declaring empty string to store the reversed
        string
3-     for i in str:
4         str1 = i + str1
5     return str1 # It will return the reverse string to the
        caller function
6
7 str = "JavaTpoint" # Given String
8 print("The original string is: ", str)
9 print("The reverse string is", reverse_string(str)) # Function
        call
10
```

The output shows the original string "JavaTpoint" and the reversed string "tniopTavaJ".

=== Code Execution Successful ===

## Example 2: Using While



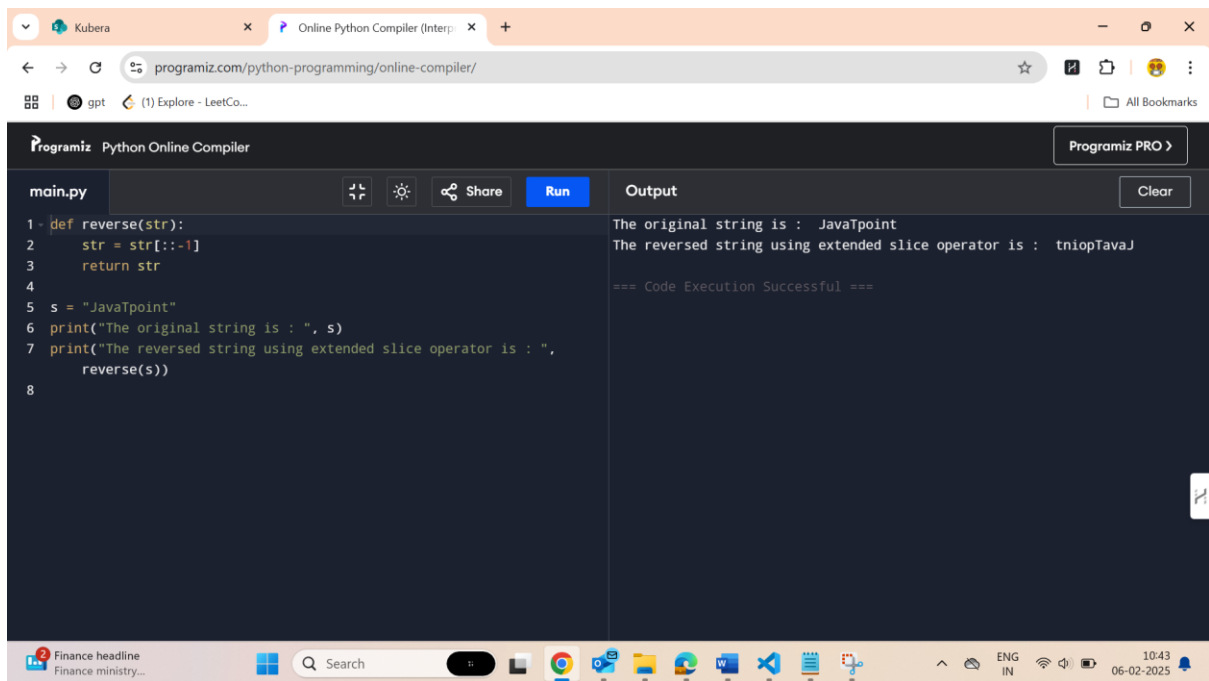
The screenshot shows the Programiz Python Online Compiler interface. The code in `main.py` uses a while loop to reverse the string "JavaTpoint" by iterating from the end of the string to the beginning.

```
1 # Reverse string
2 # Using a while loop
3
4 str = "JavaTpoint" # string variable
5 print("The original string is : ", str)
6 reverse_String = "" # Empty String
7
8 count = len(str) # Find length of a string and save in count
        variable
9- while count > 0:
10     reverse_String += str[count - 1] # save the value of
        str[count-1] in reverseString
11     count = count - 1 # decrement index
12
13 print("The reversed string using a while loop is : ",
        reverse_String) # reversed string
14
```

The output shows the original string "JavaTpoint" and the reversed string "tniopTavaJ".

=== Code Execution Successful ===

### Example 3: Using the slice operator



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a Python script that defines a `reverse` function using slicing (`str[::-1]`). The script sets `s = "JavaTpoint"` and prints both the original string and the reversed string. The output panel on the right shows the execution results: "The original string is : JavaTpoint" and "The reversed string using extended slice operator is : tniopTavaJ". A status message at the bottom of the output panel reads "=== Code Execution Successful ===". The browser's taskbar at the bottom shows the Windows Start button, a search bar, and various application icons. The system clock indicates 10:43 on 06-02-2025.

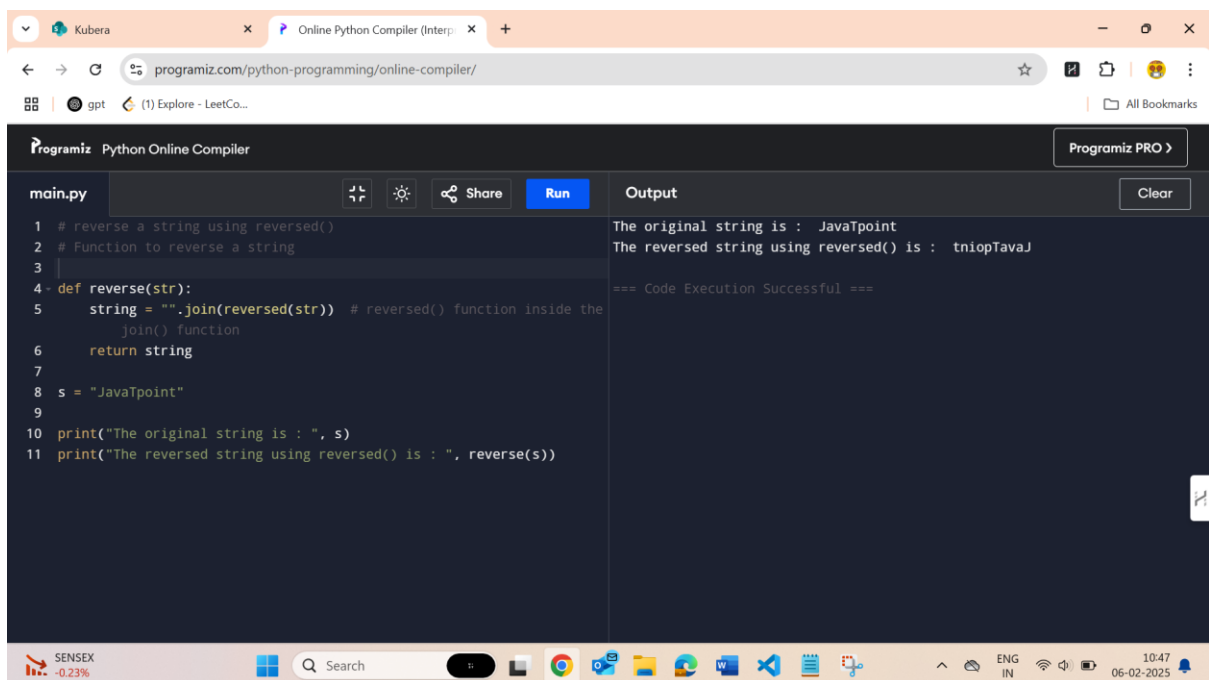
```
main.py 1 def reverse(str):  
2     str = str[::-1]  
3     return str  
4  
5 s = "JavaTpoint"  
6 print("The original string is : ", s)  
7 print("The reversed string using extended slice operator is : ",  
8     reverse(s))
```

Output

The original string is : JavaTpoint  
The reversed string using extended slice operator is : tniopTavaJ

=== Code Execution Successful ===

### Example 4: Using the reversed() function



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a Python script that defines a `reverse` function using the `reversed()` function and `join()`. The script sets `s = "JavaTpoint"` and prints both the original string and the reversed string. The output panel on the right shows the execution results: "The original string is : JavaTpoint" and "The reversed string using reversed() is : tniopTavaJ". A status message at the bottom of the output panel reads "=== Code Execution Successful ===". The browser's taskbar at the bottom shows the Windows Start button, a search bar, and various application icons. The system clock indicates 10:47 on 06-02-2025.

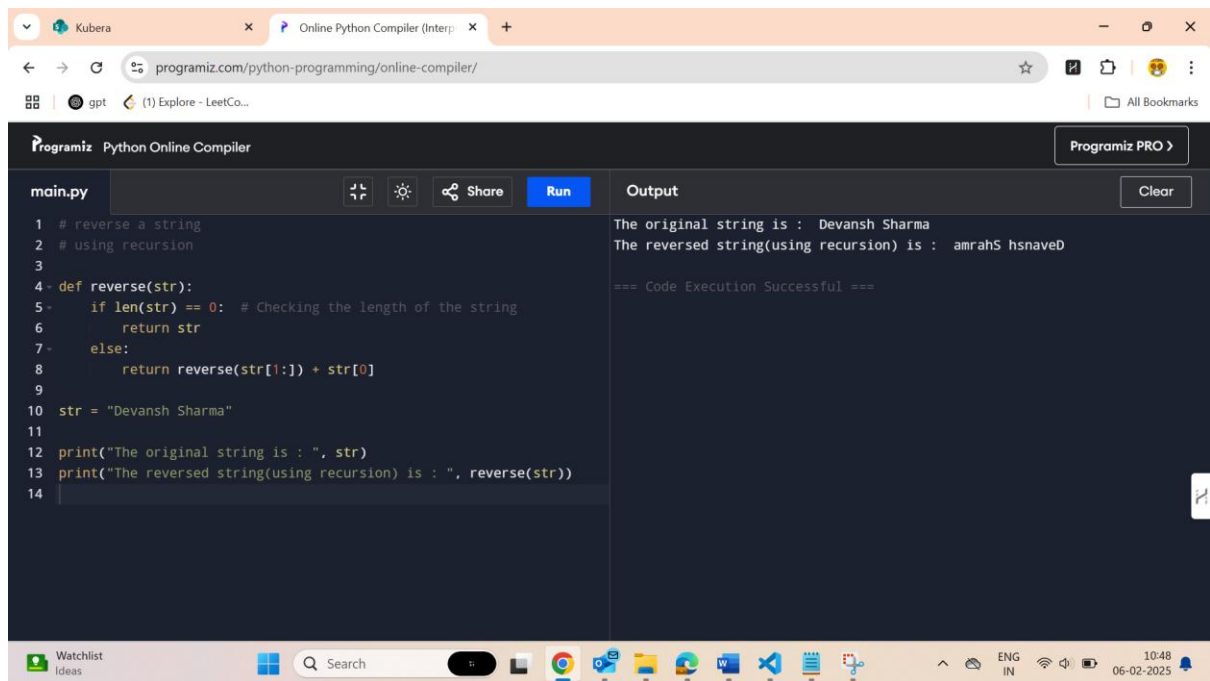
```
main.py 1 # reverse a string using reversed()  
2 # Function to reverse a string  
3  
4 def reverse(str):  
5     string = "".join(reversed(str)) # reversed() function inside the  
6     join() function  
7     return string  
8  
9 s = "JavaTpoint"  
10 print("The original string is : ", s)  
11 print("The reversed string using reversed() is : ", reverse(s))
```

Output

The original string is : JavaTpoint  
The reversed string using reversed() is : tniopTavaJ

=== Code Execution Successful ===

## Example 5: Using recursion()



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a Python program that reverses a string using recursion. The output panel on the right shows the execution results.

```
main.py
1 # reverse a string
2 # using recursion
3
4 def reverse(str):
5     if len(str) == 0: # Checking the length of the string
6         return str
7     else:
8         return reverse(str[1:]) + str[0]
9
10 str = "Devansh Sharma"
11
12 print("The original string is : ", str)
13 print("The reversed string(using recursion) is : ", reverse(str))
14
```

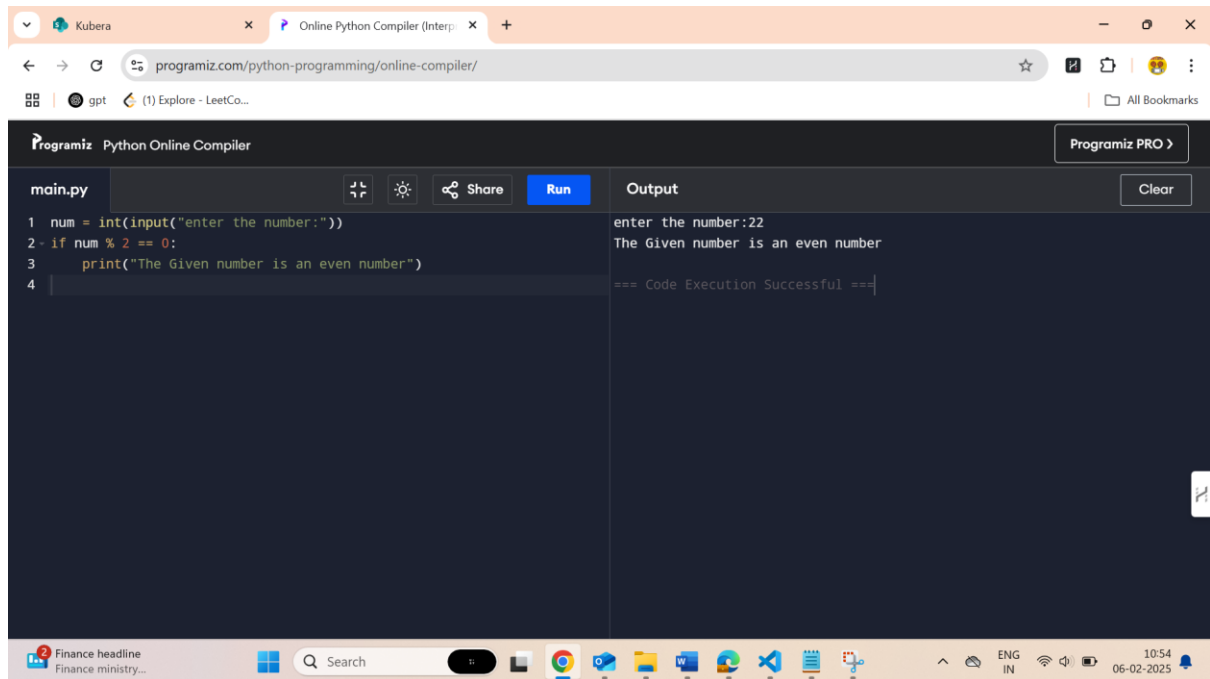
Output

```
The original string is : Devansh Sharma
The reversed string(using recursion) is : amrahS hsnaveD

=== Code Execution Successful ===
```

## Python If else:

Example 1: Simple Python program to understand the if statement



The screenshot shows the Programiz Python Online Compiler interface. The code editor on the left contains a simple Python program using an if statement to check if a number is even. The output panel on the right shows the execution results.

```
main.py
1 num = int(input("enter the number:"))
2 if num % 2 == 0:
3     print("The Given number is an even number")
4
```

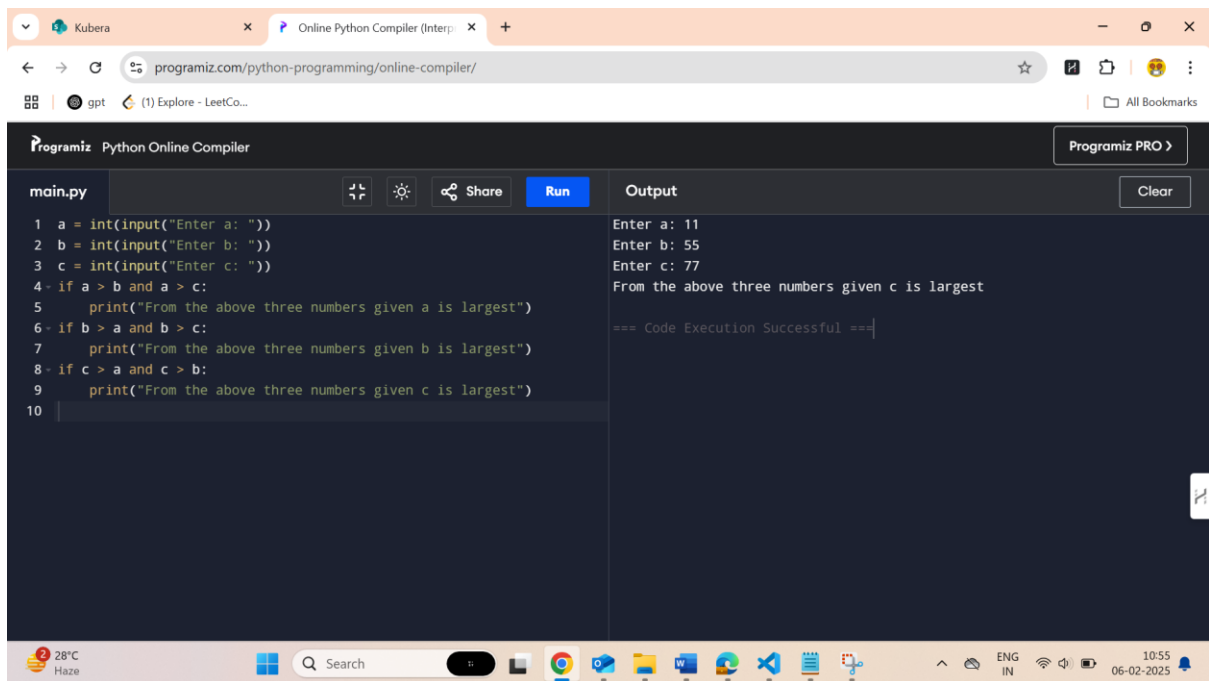
Output

```
enter the number:22
The Given number is an even number

=== Code Execution Successful ===
```



## Example 2: Program to print the largest of the three numbers



The screenshot shows a web browser window with the URL `programiz.com/python-programming/online-compiler/`. The page title is "Programiz Python Online Compiler". The code editor on the left contains the following Python code:

```
main.py
1 a = int(input("Enter a: "))
2 b = int(input("Enter b: "))
3 c = int(input("Enter c: "))
4 if a > b and a > c:
5     print("From the above three numbers given a is largest")
6 if b > a and b > c:
7     print("From the above three numbers given b is largest")
8 if c > a and c > b:
9     print("From the above three numbers given c is largest")
10
```

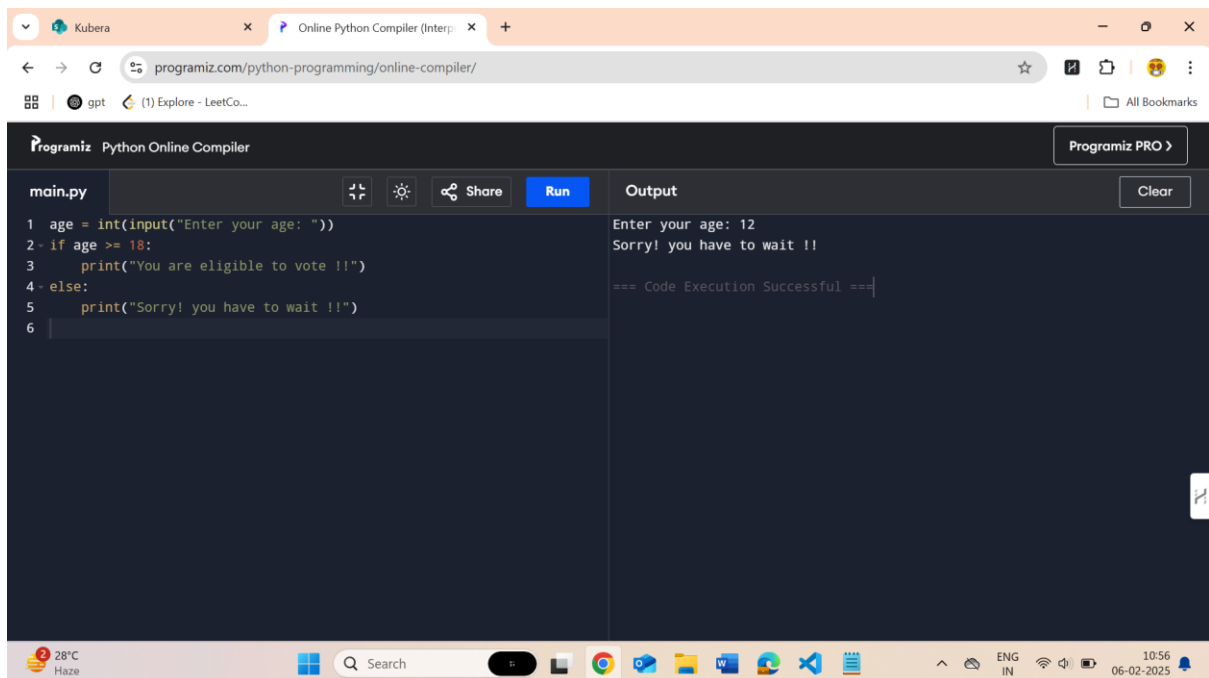
The "Run" button is highlighted in blue. The output window on the right shows the following text:

```
Enter a: 11
Enter b: 55
Enter c: 77
From the above three numbers given c is largest

=== Code Execution Successful ===
```

The bottom of the browser window shows a Windows taskbar with a search bar, task view button, and several application icons. The system tray shows the date and time as 10:55 on 06-02-2025.

## Example 3: Program to check whether a person is eligible to vote or not (if-else)



The screenshot shows the same online Python compiler interface. The code editor on the left contains the following Python code:

```
main.py
1 age = int(input("Enter your age: "))
2 if age >= 18:
3     print("You are eligible to vote !!")
4 else:
5     print("Sorry! you have to wait !!")
6
```

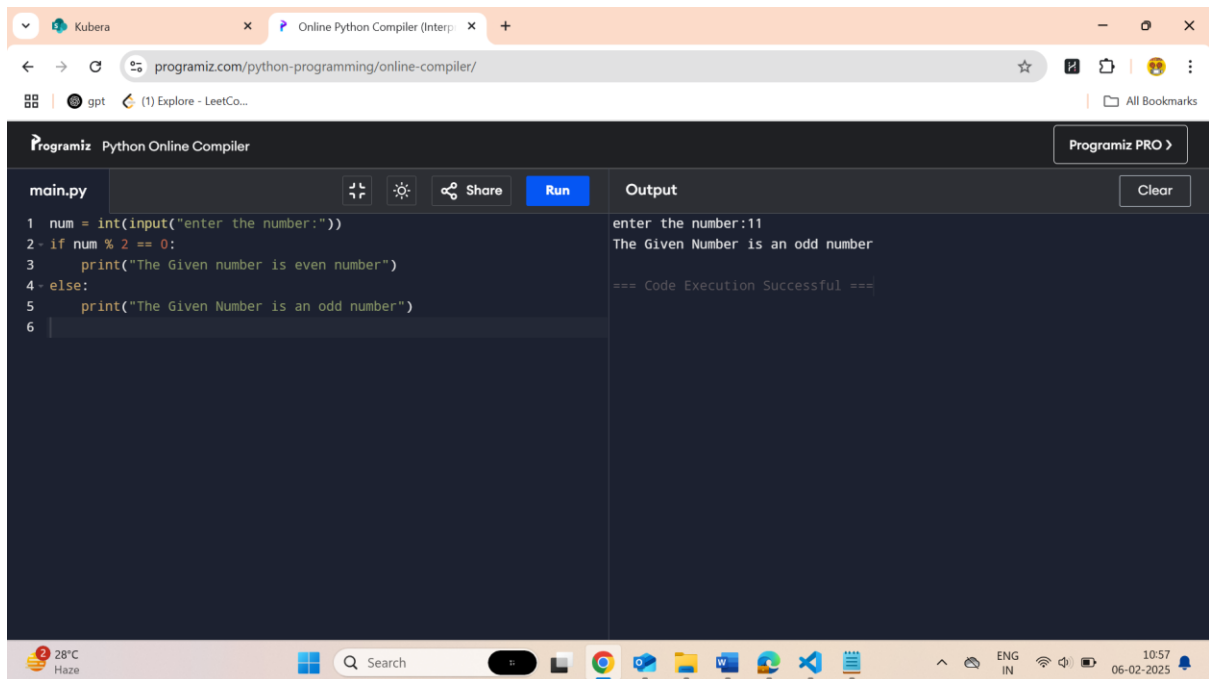
The "Run" button is highlighted in blue. The output window on the right shows the following text:

```
Enter your age: 12
Sorry! you have to wait !!

=== Code Execution Successful ===
```

The bottom of the browser window shows a Windows taskbar with a search bar, task view button, and several application icons. The system tray shows the date and time as 10:56 on 06-02-2025.

#### Example 4: Program to check whether a number is even or not (if-else)



The screenshot shows a web browser window with the URL `programiz.com/python-programming/online-compiler/`. The browser tabs include 'Kubera' and 'Online Python Compiler (Interp...'. The page title is 'Programiz Python Online Compiler'. The main area is divided into two panels: 'main.py' on the left and 'Output' on the right. The 'main.py' panel contains the following Python code:

```
1 num = int(input("enter the number:"))
2 if num % 2 == 0:
3     print("The Given number is even number")
4 else:
5     print("The Given Number is an odd number")
6
```

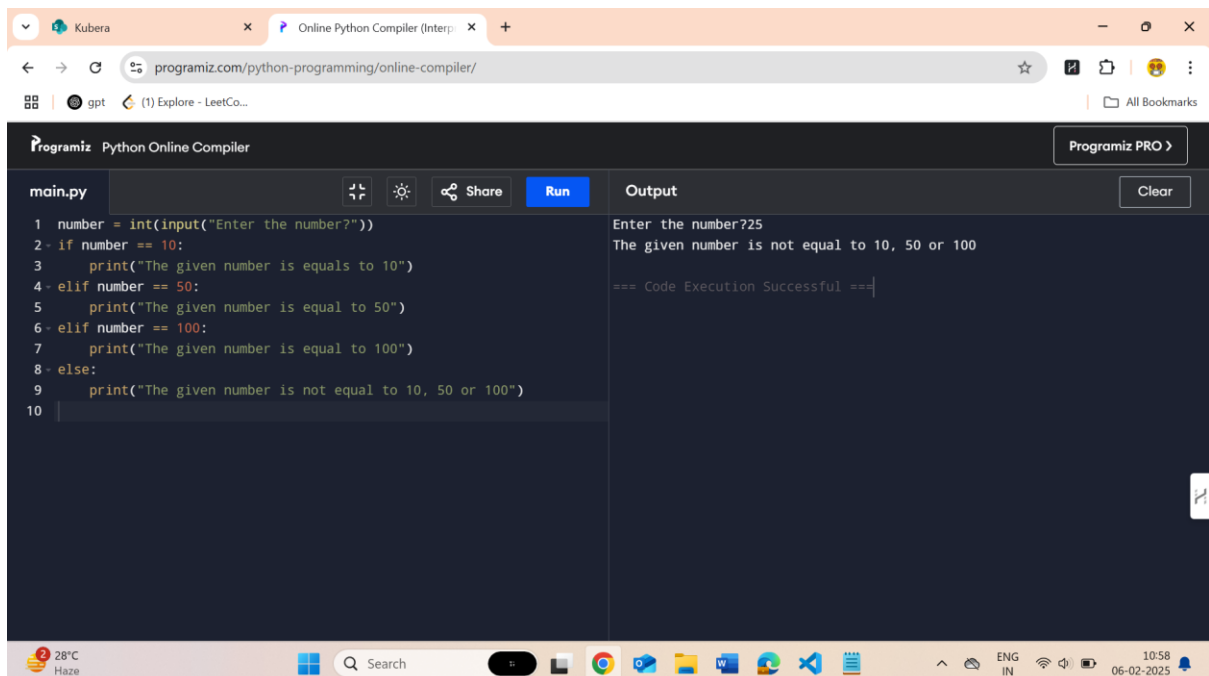
The 'Output' panel shows the result of running the code with the input '11':

```
enter the number:11
The Given Number is an odd number

=== Code Execution Successful ===
```

The bottom of the browser window shows a Windows taskbar with a search bar, system icons, and the date/time '10:57 06-02-2025'.

#### Example 5: Simple Python program to understand elif statement



The screenshot shows the same online Python compiler interface. The 'main.py' panel contains the following Python code:

```
1 number = int(input("Enter the number?"))
2 if number == 10:
3     print("The given number is equals to 10")
4 elif number == 50:
5     print("The given number is equal to 50")
6 elif number == 100:
7     print("The given number is equal to 100")
8 else:
9     print("The given number is not equal to 10, 50 or 100")
10
```

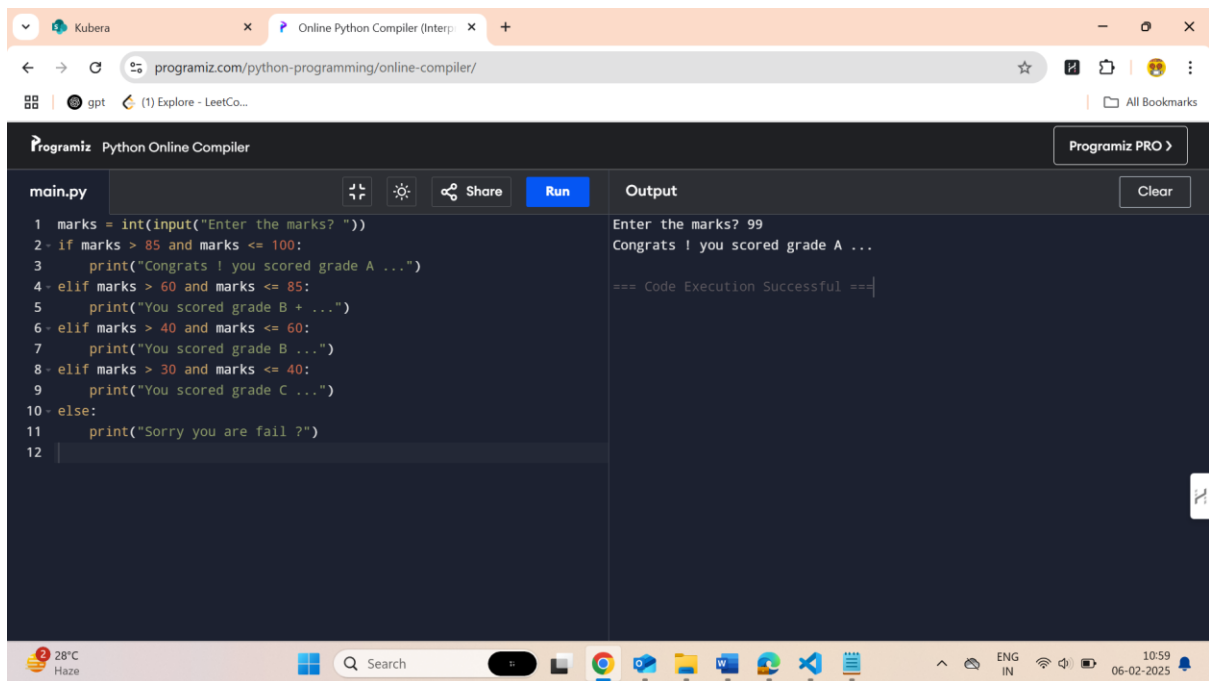
The 'Output' panel shows the result of running the code with the input '25':

```
Enter the number?25
The given number is not equal to 10, 50 or 100

=== Code Execution Successful ===
```

The bottom of the browser window shows a Windows taskbar with a search bar, system icons, and the date/time '10:58 06-02-2025'.

## Example 6: Python program using elif for grading



The screenshot shows the Programiz Python Online Compiler interface. The code in `main.py` is as follows:

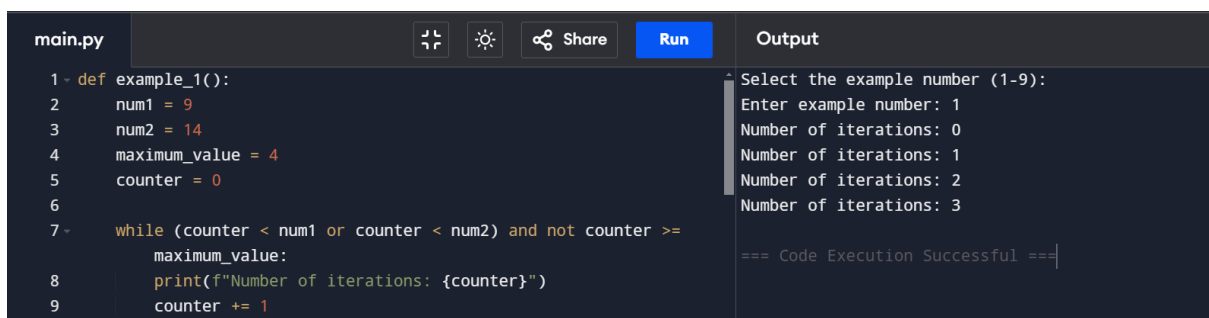
```
1 marks = int(input("Enter the marks? "))
2 if marks > 85 and marks <= 100:
3     print("Congrats ! you scored grade A ...")
4 elif marks > 60 and marks <= 85:
5     print("You scored grade B + ...")
6 elif marks > 40 and marks <= 60:
7     print("You scored grade B ...")
8 elif marks > 30 and marks <= 40:
9     print("You scored grade C ...")
10 else:
11     print("Sorry you are fail ?")
12
```

The output shows the user entered 99, resulting in the message "Congrats ! you scored grade A ...". Below the output, it says "=== Code Execution Successful ===".

## LOOPS:

### Examples:

1.

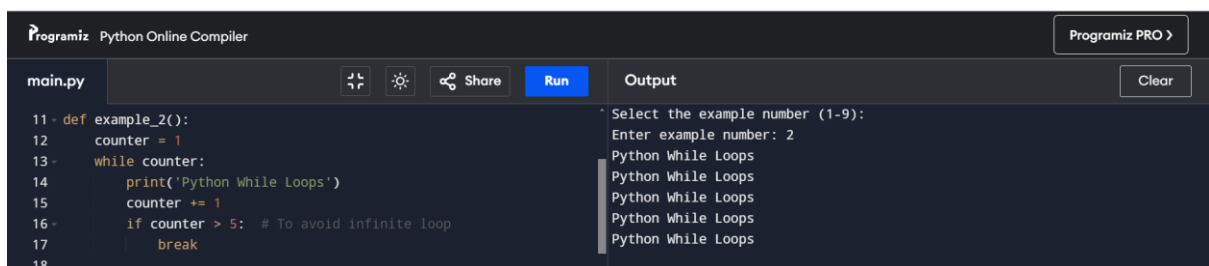


The screenshot shows the Programiz Python Online Compiler interface. The code in `main.py` is as follows:

```
1 def example_1():
2     num1 = 9
3     num2 = 14
4     maximum_value = 4
5     counter = 0
6
7     while (counter < num1 or counter < num2) and not counter >=
8         maximum_value:
9         print(f"Number of iterations: {counter}")
10        counter += 1
```

The output shows the user selected example number 1, entered 1, and the program printed "Number of iterations: 0", "Number of iterations: 1", "Number of iterations: 2", and "Number of iterations: 3". Below the output, it says "=== Code Execution Successful ===".

2.



The screenshot shows the Programiz Python Online Compiler interface. The code in `main.py` is as follows:

```
11 def example_2():
12     counter = 1
13     while counter:
14         print('Python While Loops')
15         counter += 1
16         if counter > 5: # To avoid infinite loop
17             break
18
```

The output shows the user selected example number 1, entered 2, and the program printed "Python While Loops" five times. Below the output, it says "=== Code Execution Successful ===".

3.

```
Programiz Python Online Compiler Programiz PRO >
```

```
main.py 🔍 🌞 🔗 Share Run Output Clear
```

```
18
19 def example_3():
20     for string in "While Loops":
21         if string in "oie":
22             continue
23         print(f'Current Letter: {string}')
24
25
26
27
28
29
```

```
Select the example number (1-9):
Enter example number: 3
Current Letter: W
Current Letter: h
Current Letter: l
Current Letter:
Current Letter: L
Current Letter: p
Current Letter: s

=== Code Execution Successful ===
```

4.

```
Programiz Python Online Compiler Programiz PRO >
```

```
main.py 🔍 🌞 🔗 Share Run Output Clear
```

```
36 def example_4():
37     my_list = [1, 2, 3, 4]
38     count = 1
39     for item in my_list:
40         if item == 4:
41             print("Item matched")
42             break
43         count += 1
44     print("Found at location", count)
45
```

```
Select the example number (1-9):
Enter example number: 4
Item matched
Found at location 4

=== Code Execution Successful ===
```

5.

```
Programiz Python Online Compiler Programiz PRO >
```

```
main.py 🔍 🌞 🔗 Share Run Output Clear
```

```
46 def example_5():
47     i = 0
48     while 1:
49         print(i, " ", end="")
50         i += 1
51         if i == 10:
52             break
53     print("came out of while loop")
54
```

```
Select the example number (1-9):
Enter example number: 5
0 1 2 3 4 5 6 7 8 9 came out of while loop

=== Code Execution Successful ===
```

6.

```
Programiz Python Online Compiler Programiz PRO >
```

```
main.py 🔍 🌞 🔗 Share Run Output Clear
```

```
55 def example_6():
56     n = 2
57     while True:
58         i = 1
59         while i <= 10:
60             print(f"{n} X {i} = {n * i}")
61             i += 1
62         choice = int(input("Do you want to continue printing the
        table? Press 0 for no: "))
63         if choice == 0:
64             print("Exiting the program...")
65             break
66         n += 1
67     print("Program finished successfully.")
68
```

```
Select the example number (1-9):
Enter example number: 6
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20
Do you want to continue printing the table? Press 0 for no:
```

7.

```
Programiz Python Online Compiler
main.py
71
72 - def example_7():
73 -     for iterator in range(10, 21):
74 -         if iterator == 15:
75 -             continue
76 -         print(iterator)
77
78
79
80
81
82
```

Output

```
Select the example number (1-9):
Enter example number: 7
10
11
12
13
14
16
17
18
19
20
```

8.

```
Programiz Python Online Compiler
main.py
86
87 - def example_8():
88     string = "JavaTpoint"
89     iterator = 0
90
91 -     while iterator < len(string):
92 -         if string[iterator] == 'a':
93             iterator += 1
94             continue
95         print(string[iterator])
96         iterator += 1
97
98
```

Output

```
Select the example number (1-9):
Enter example number: 8
J
v
T
p
o
i
n
t

=== Code Execution Successful ===
```

9.

```
Programiz Python Online Compiler
main.py
101 - def example_9():
102     numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
103     sq_num = [num ** 2 for num in numbers if num % 2 == 0]
104     print(sq_num)
105
106
107
108
```

Output

```
Select the example number (1-9):
Enter example number: 9
[4, 16, 36, 64, 100]

=== Code Execution Successful ===
```

10.

```
Programiz Python Online Compiler
main.py
1 - def while_loop_multiple_conditions():
2     # Example 10: While Loop with Multiple Conditions
3     num1 = 9
4     num2 = 14
5     maximum_value = 4
6     counter = 0
7     # While loop with multiple conditions
8     result = []
9 -     while (counter < num1 or counter < num2) and not counter >=
10         maximum_value:
11         result.append(f"Number of iterations: {counter}")
12         counter += 1
13     return result
```

Output

```
Select the example number (10-18):
Enter example number: 10
Number of iterations: 0
Number of iterations: 1
Number of iterations: 2
Number of iterations: 3

=== Code Execution Successful ===
```

11.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
14- def single_statement_while():
15-     # Example 11: Single Statement While Loop
16-     counter = 1
17-     # Single statement while loop
18-     result = []
19-     while counter:
20-         result.append('Python While Loops')
21-         counter += 1
22-         if counter > 5: # Preventing infinite loop
23-             break
24-     return result
```

```
Select the example number (10-18):
Enter example number: 11
Python While Loops
Python While Loops
Python While Loops
Python While Loops
Python While Loops

=== Code Execution Successful ===
```

12.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
26- def continue_statement_example():
27-     # Example 12: Continue Statement Example (for loop)
28-     result = []
29-     for string in "While Loops":
30-         if string in "oie":
31-             continue
32-         result.append(f'Current Letter: {string}')
33-     return result
```

```
Select the example number (10-18):
Enter example number: 12
Current Letter: W
Current Letter: h
Current Letter: l
Current Letter:
Current Letter: L
Current Letter: p
Current Letter: s

=== Code Execution Successful ===
```

13.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
40- def break_statement_example():
41-     # Example 13: Break Statement Example (for loop)
42-     my_list = [1, 2, 3, 4]
43-     count = 1
44-     result = []
45-     for item in my_list:
46-         if item == 4:
47-             result.append("Item matched")
48-             break
49-         count += 1
50-     result.append("Found at location " + str(count))
51-     return result
```

```
Select the example number (10-18):
Enter example number: 13
Item matched
Found at location 4

=== Code Execution Successful ===
```

14.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
53- def break_statement_while():
54-     # Example 14: Break Statement Example (while loop)
55-     i = 0
56-     result = []
57-     while 1:
58-         result.append(f"{i} ")
59-         i += 1
60-         if i == 10:
61-             break
62-     result.append("came out of while loop")
63-     return result
```

```
Select the example number (10-18):
Enter example number: 14
0
1
2
3
4
5
6
7
8
9
came out of while loop
```

15.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
--
69 def break_statement_nested_loops():
70     # Example 15: Break Statement Example (nested loops)
71     n = 2
72     result = []
73     while True:
74         i = 1
75         while i <= 10:
76             result.append(f"{n} X {i} = {n * i}")
77             i += 1
78             choice = int(input("Do you want to continue printing the
table? Press 0 for no: "))
79             if choice == 0:
80                 result.append("Exiting the program...")
81                 break
82             n += 1
83     result.append("Program finished successfully.")
84     return result
85
```

```
Select the example number (10-18):
Enter example number: 15
Do you want to continue printing the table? Press 0 for no: 0
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20
Exiting the program...
Program finished successfully.

=== Code Execution Successful ===
```

16.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
85
86 def continue_statement_for_loop():
87     # Example 16: Continue Statement Example (for loop with 10-20)
88     result = []
89     for iterator in range(10, 21):
90         if iterator == 15:
91             continue
92         result.append(str(iterator))
93     return result
94
95
96
97
```

```
Select the example number (10-18):
Enter example number: 16
10
11
12
13
14
16
17
18
19
20
```

17.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
101
102 def continue_statement_while_loop():
103     # Example 17: Continue Statement Example (while loop with
string)
104     string = "JavaTpoint"
105     iterator = 0
106     result = []
107     while iterator < len(string):
108         if string[iterator] == 'a':
109             iterator += 1
110             continue
111         result.append(string[iterator])
112         iterator += 1
113     return result
114
```

```
Select the example number (10-18):
Enter example number: 17
J
v
T
p
o
i
n
t

=== Code Execution Successful ===
```

18.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
119 def continue_statement_list_comprehension():
120     # Example 18: Continue Statement Example (list comprehension)
121     numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
122     sq_num = [num ** 2 for num in numbers if num % 2 == 0]
123     return sq_num
124
125
126
127
```

```
Select the example number (10-18):
Enter example number: 18
4
16
36
64
100

=== Code Execution Successful ===
```

# Python Strings and Tuple

## examples:

1.

```
Programiz Python Online Compiler
main.py
1 - def example1():
2   # Using single quotes
3   str1 = 'Hello Python'
4   print(str1)
5   # Using double quotes
6   str2 = "Hello Python"
7   print(str2)
8   # Using triple quotes
9   str3 = '''Triple quotes are generally used for
10  represent the multiline or
11  docstring'''
12  print(str3)
```

Output

```
Enter the example number (1-14):
1
Hello Python
Hello Python
'''Triple quotes are generally used for
represent the multiline or
docstring'''
=== Code Execution Successful ===
```

2.

```
Programiz Python Online Compiler
main.py
14 - def example2():
15   str = "HELLO"
16   print(str[0])
17   print(str[1])
18   print(str[2])
19   print(str[3])
20   print(str[4])
21   # It returns the IndexError because 6th index doesn't
    exist
22   try:
23     print(str[6])
24   except IndexError:
25     print("Index Error: string index out of range")
```

Output

```
Enter the example number (1-14):
2
ERROR!
H
E
L
L
O
Index Error: string index out of range
=== Code Execution Successful ===
```

3.

```
Programiz Python Online Compiler
main.py
27 - def example3():
28   str = "JAVATPOINT"
29   # Start 0th index to end
30   print(str[0:])
31   # Starts 1th index to 4th index
32   print(str[1:5])
33   # Starts 2nd index to 3rd index
34   print(str[2:4])
35   # Starts 0th to 2nd index
36   print(str[:3])
37   # Starts 4th to 6th index
38   print(str[4:7])
```

Output

```
Enter the example number (1-14):
3
JAVATPOINT
AVAT
VA
JAV
TPO
=== Code Execution Successful ===
```



4.

```
Programiz Python Online Compiler Programiz PRO >

main.py ⌵ ☀ 🔗 Share Run Output Clear

40 def example4():
41     str = 'JAVATPOINT'
42     print(str[-1])
43     print(str[-3])
44     print(str[-2:])
45     print(str[-4:-1])
46     print(str[-7:-2])
47     # Reversing the given string
48     print(str[::-1])
49     try:
50         print(str[-12])
51     except IndexError:
52         print("IndexError: string index out of range")
53
```

```
Enter the example number (1-14):
4
ERROR!
T
I
NT
OIN
ATPOI
TNIPTAVAJ
IndexError: string index out of range

=== Code Execution Successful ===
```

5.

```
Programiz Python Online Compiler Programiz PRO >

main.py ⌵ ☀ 🔗 Share Run Output Clear

53
54 def example5():
55     str = "HELLO"
56     try:
57         str[0] = "h"
58     except TypeError:
59         print("TypeError: 'str' object does not support item
        assignment")
60     str = "hello"
61     print(str)
62
```

```
Enter the example number (1-14):
5
ERROR!
TypeError: 'str' object does not support item assignment
hello

=== Code Execution Successful ===
```

6.

```
Programiz Python Online Compiler Programiz PRO >

main.py ⌵ ☀ 🔗 Share Run Output Clear

63 def example6():
64     str1 = "JAVATPOINT"
65     try:
66         del str1[1]
67     except TypeError:
68         print("TypeError: 'str' object doesn't support item
        deletion")
69     try:
70         del str1
71         print(str1)
72     except NameError:
73         print("NameError: name 'str1' is not defined")
74
```

```
Enter the example number (1-14):
6
ERROR!
TypeError: 'str' object doesn't support item deletion
NameError: name 'str1' is not defined

=== Code Execution Successful ===
```

7.

```
Programiz Python Online Compiler Programiz PRO >

main.py ⌵ ☀ 🔗 Share Run Output Clear

75 def example7():
76     str = "Hello"
77     str1 = " world"
78     print(str*3) # prints HelloHelloHello
79     print(str+str1) # prints Hello world
80     print(str[4]) # prints o
81     print(str[2:4]) # prints ll
82     print('w' in str) # prints False as w is not present in
        str
83     print('wo' not in str1) # prints False as wo is present in
        str1
84     print(r'C://python37') # prints C://python37 as it is
        written
85     print("The string str : %s"%(str)) # prints The string str
        : Hello
```

```
Enter the example number (1-14):
7
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello

=== Code Execution Successful ===
```

8.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
87 - def example8():
88     # using triple quotes
89     print(''''They said, "What's there?''')
90     # escaping single quotes
91     print('They said, "What\'s going on?")
92     # escaping double quotes
93     print("They said, \"What's going on?\"")
94
95 - def example9():
96     Integer = 10
97     Float = 1.290
98     String = "Devansh"
99     print("Hi I am Integer ... My value is %d\nHi I am float
    ... My value is %f\nHi I am string ... My value is
    %s"%(Integer,Float,String))
100
```

```
Enter the example number (1-14):
8
'''They said, "What's there?"
They said, "What's going on?"
They said, "What's going on?"

=== Code Execution Successful ===
```

9.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
94
95 - def example9():
96     Integer = 10
97     Float = 1.290
98     String = "Devansh"
99     print("Hi I am Integer ... My value is %d\nHi I am float
    ... My value is %f\nHi I am string ... My value is
    %s"%(Integer,Float,String))
100
```

```
Enter the example number (1-14):
9
Hi I am Integer ... My value is 10
Hi I am float ... My value is 1.290000
Hi I am string ... My value is Devansh

=== Code Execution Successful ===
```

10.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
101 - def example10():
102     # Python code to show the difference between creating a
    list and a tuple
103     list_ = [4, 5, 7, 1, 7]
104     tuple_ = (4, 1, 8, 3, 9)
105     print("List is: ", list_)
106     print("Tuple is: ", tuple_)
```

```
Enter the example number (1-14):
10
List is:  [4, 5, 7, 1, 7]
Tuple is:  (4, 1, 8, 3, 9)

=== Code Execution Successful ===
```

11.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

Output

Clear

```
107
108 - def example11():
109     # Updating the element of list and tuple at a particular
    index
110     list_ = ["Python", "Lists", "Tuples", "Differences"]
111     tuple_ = ("Python", "Lists", "Tuples", "Differences")
112     list_[3] = "Mutable"
113     print(list_)
114 - try:
115     tuple_[3] = "Immutable"
116     print(tuple_)
117 - except TypeError:
118     print("Tuples cannot be modified because they are
    immutable")
```

```
Enter the example number (1-14):
11
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable

=== Code Execution Successful ===
```

12.

Programiz Python Online Compiler

Programiz PRO >

main.py

Share

Run

```
119
120 - def example12():
121     # Code to show the difference in the size of a list and a
        tuple
122     list_ = ["Python", "Lists", "Tuples", "Differences"]
123     tuple_ = ("Python", "Lists", "Tuples", "Differences")
124     # printing sizes
125     print("Size of tuple: ", tuple_.__sizeof__())
126     print("Size of list: ", list_.__sizeof__())
127
```

Output

Clear

```
Enter the example number (1-14):
12
Size of tuple: 56
Size of list: 72

=== Code Execution Successful ===
```