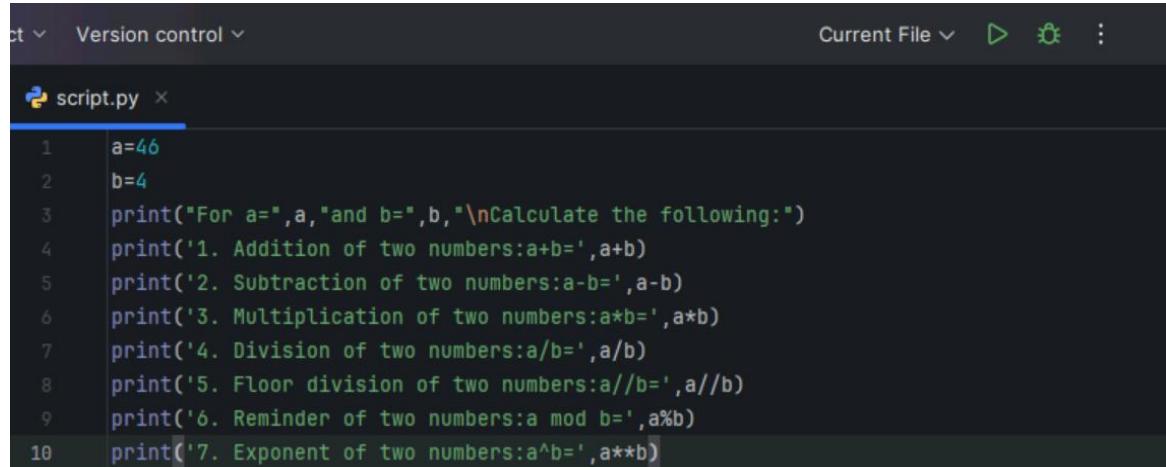


PYTHON BASICS

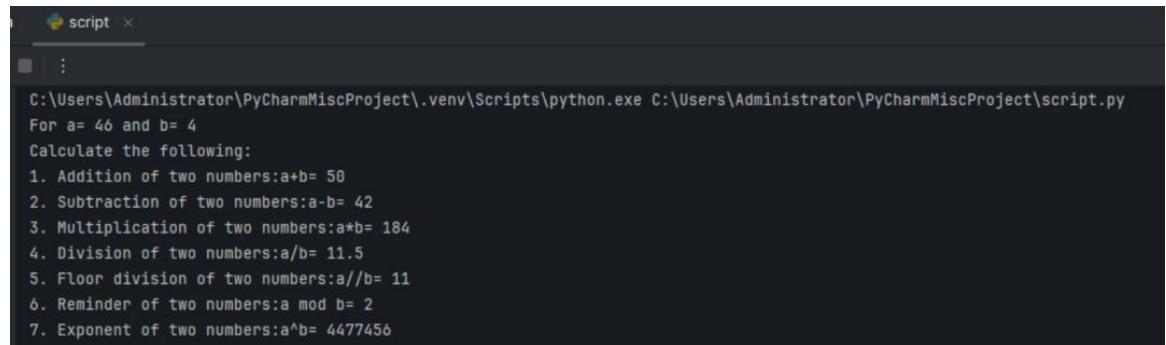
Python Operators

1. Python Program to perform ‘Arithmetic Operators’



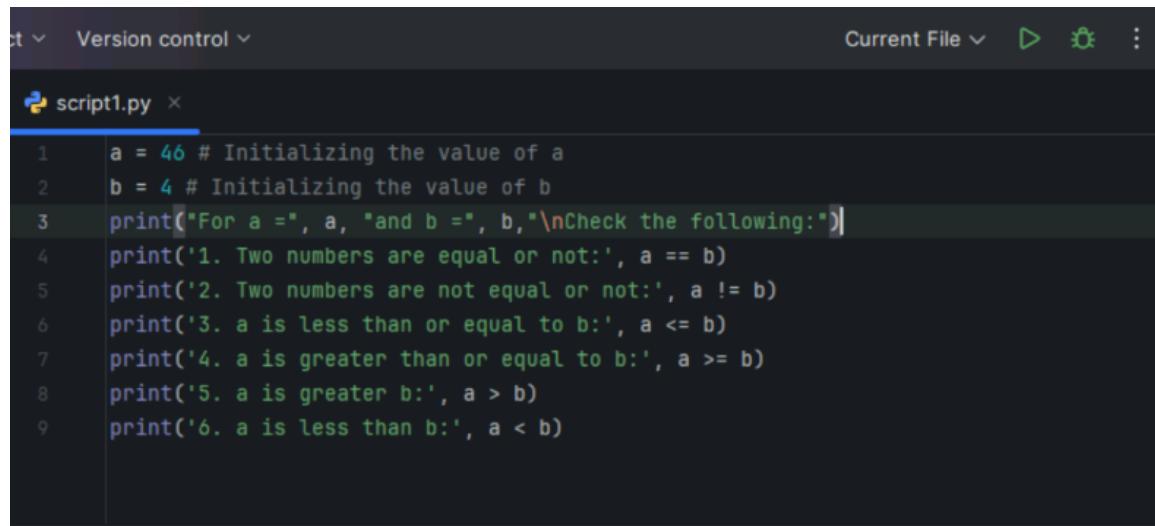
```
a=46
b=4
print("For a=",a,"and b=",b,"Calculate the following:")
print('1. Addition of two numbers:a+b=',a+b)
print('2. Subtraction of two numbers:a-b=',a-b)
print('3. Multiplication of two numbers:a*b=',a*b)
print('4. Division of two numbers:a/b=',a/b)
print('5. Floor division of two numbers:a//b=',a//b)
print('6. Remainder of two numbers:a mod b=',a%b)
print('7. Exponent of two numbers:a^b=',a**b)
```

Output:



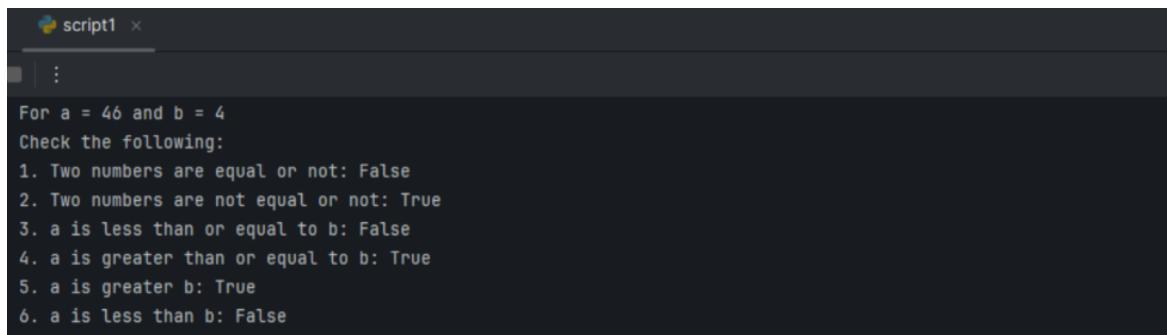
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script.py
For a= 46 and b= 4
Calculate the following:
1. Addition of two numbers:a+b= 50
2. Subtraction of two numbers:a-b= 42
3. Multiplication of two numbers:a*b= 184
4. Division of two 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
```

2. Python Program to perform ‘Comparison Operators’



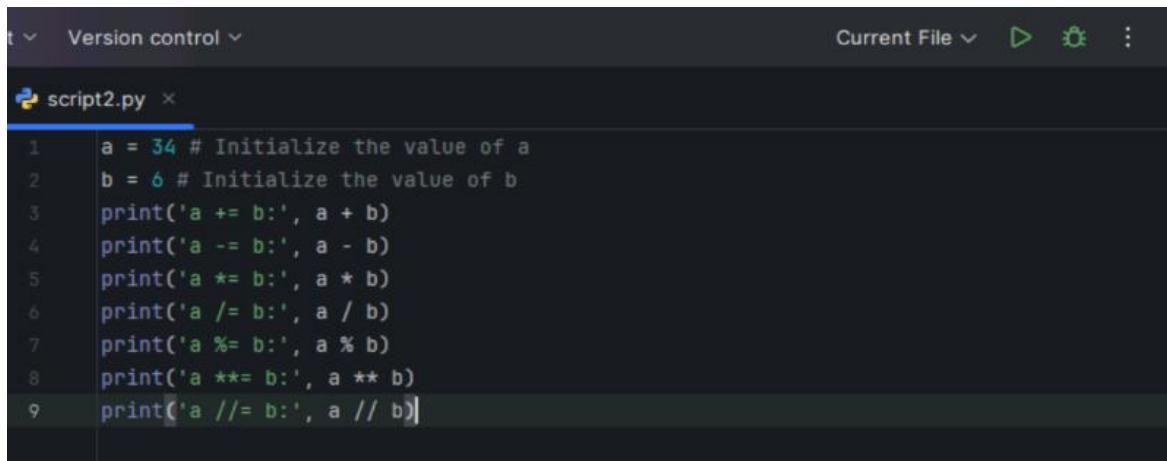
```
a = 46 # Initializing the value of a
b = 4 # Initializing the value of b
print("For a =", a, "and b =", b,"Check the following:")
print('1. Two numbers are equal or not:', a == b)
print('2. Two numbers are not equal or not:', a != b)
print('3. a is less than or equal to b:', a <= b)
print('4. a is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

Output:



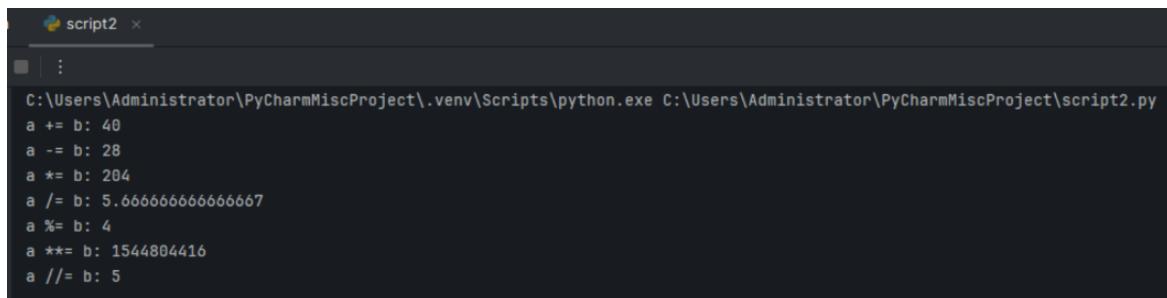
```
script1 ×
:
For a = 4 and b = 4
Check the following:
1. Two numbers are equal or not: False
2. Two numbers are not equal or not: True
3. a is less than or equal to b: False
4. a is greater than or equal to b: True
5. a is greater b: True
6. a is less than b: False
```

3. Python Program to perform ‘Assignment Operators’



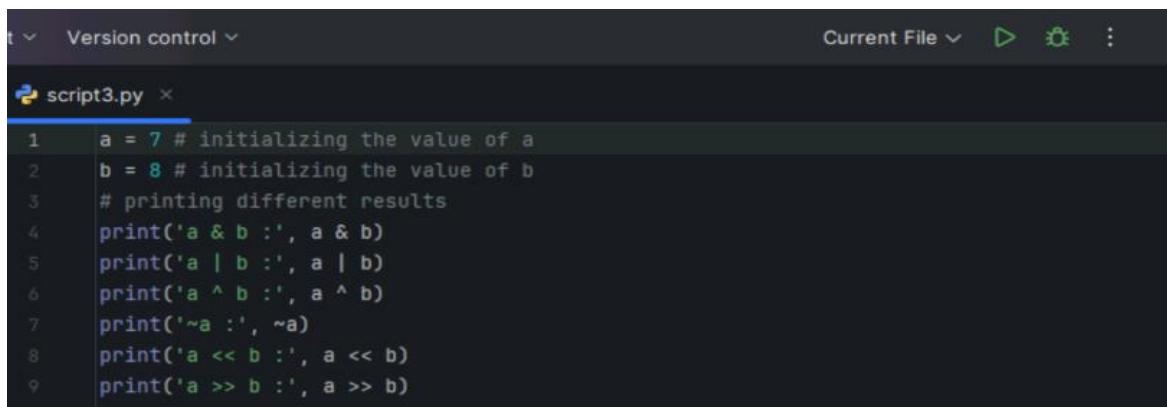
```
Version control Current File ▾ ▶ ⚡ ⋮
script2.py ×
1 a = 34 # Initialize the value of a
2 b = 6 # Initialize the value of b
3 print('a += b:', a + b)
4 print('a -= b:', a - b)
5 print('a *= b:', a * b)
6 print('a /= b:', a / b)
7 print('a %= b:', a % b)
8 print('a **= b:', a ** b)
9 print('a //=: b:', a // b)
```

Output:



```
script2 ×
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script2.py
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //=: 5
```

4. Python Program to perform ‘Bitwise Operators’



```
Version control Current File ▾ ▶ ⚡ ⋮
script3.py ×
1 a = 7 # initializing the value of a
2 b = 8 # initializing the value of b
3 # printing different results
4 print('a & b :', a & b)
5 print('a | b :', a | b)
6 print('a ^ b :', a ^ b)
7 print('~a :', ~a)
8 print('a << b :', a << b)
9 print('a >> b :', a >> b)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script3.py
a & b : 0
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0
```

5. Python Program to perform ‘Logical Operators’

```
a = 7 # initializing the value of a
# printing different results
print("For a = 7, checking whether the following conditions are True or False:")
print('\"a > 5 and a < 7\" =>', a > 5 and a < 7)
print('\"a > 5 or a < 7\" =>', a > 5 or a < 7)
print('\"not (a > 5 and a < 7)\" =>', not(a > 5 and a < 7))
```

Output:

```
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
```

6. Python Program to perform ‘Membership Operators’

```
myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 31
y = 28
print("Given List:", myList)

if (x not in myList):
    print("x =", x,"is NOT present in the given list.")
else:
    print("x =", x,"is present in the given list.")

if (y in myList):
    print("y =", y,"is present in the given list.")
else:
    print("y =", y,"is NOT present in the given list.")
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script5.py
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.
```

7. Python Program to perform ‘Identity Operators’

```
1 # initializing two variables a and b
2 a = ["Rose", "Lotus"]
3 b = ["Rose", "Lotus"]
4 # initializing a variable c and storing the value of a in c
5 c = a
6 # printing the different results
7 print("a is c => ", a is c)
8 print("a is not c => ", a is not c)
9 print("a is b => ", a is b)
10 print("a is not b => ", a is not b)
11 print("a == b => ", a == b)
12 print("a != b => ", a != b)
```

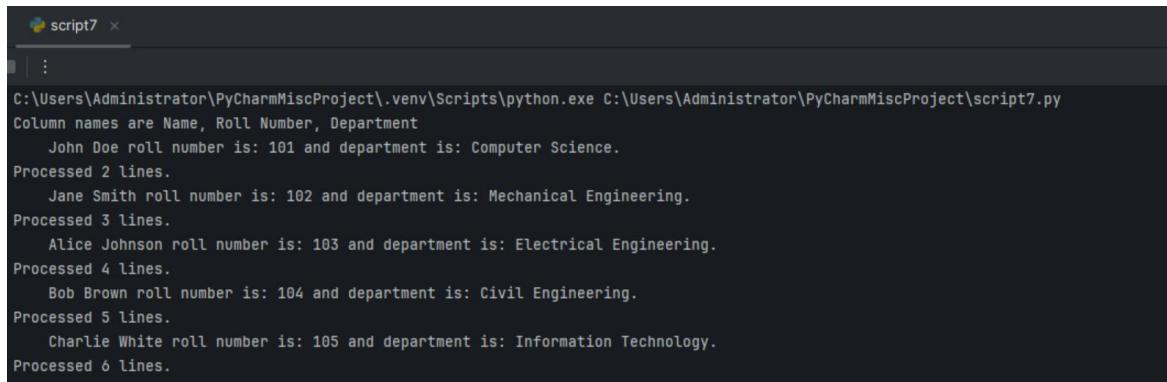
Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script6.py
a is c => True
a is not c => False
a is b => False
a is not b => True
a == b => True
a != b => False
```

8. Python program to read a ‘CSV file’

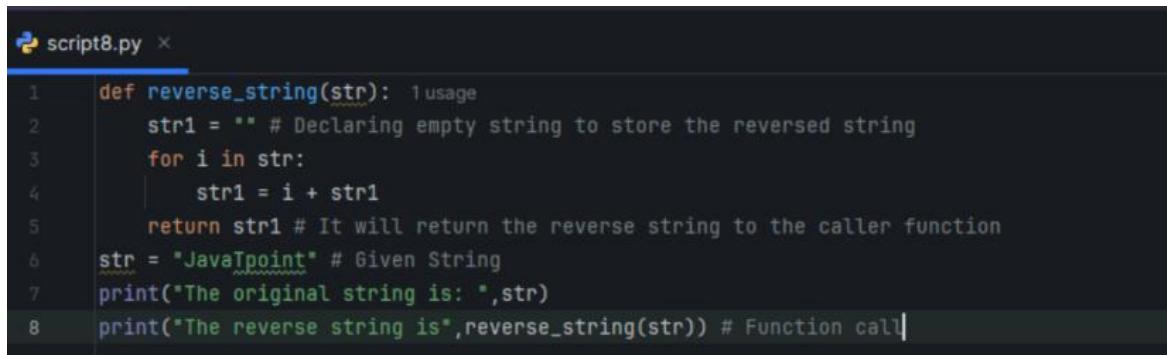
```
1 # Importing the csv module
2 import csv
3 # open file by passing the file path.
4 with open('C:\Users\Administrator\Downloads\student_records.csv') as csv_file:
5     csv_read = csv.reader(csv_file, delimiter=',') #Delimeter is comma
6     count_line = 0
7 # Iterate the file object or each row of the file
8     for row in csv_read:
9         if count_line == 0:
10             print(f'Column names are {", ".join(row)}')
11             count_line += 1
12         else:
13             print(f'\t{row[0]} roll number is: {row[1]} and department is: {row[2]}.')
14             count_line += 1
15     print(f'Processed {count_line} lines.') # This line will print number of line fro the file
```

Output:



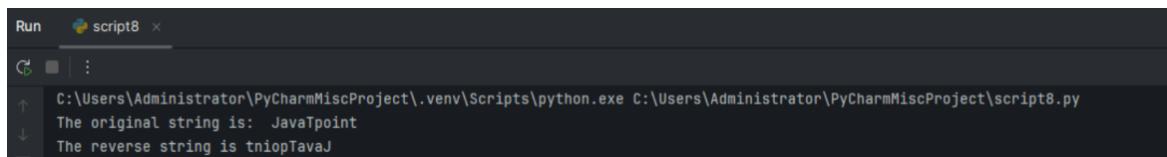
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script7.py
Column names are Name, Roll Number, Department
    John Doe roll number is: 101 and department is: Computer Science.
Processed 2 lines.
    Jane Smith roll number is: 102 and department is: Mechanical Engineering.
Processed 3 lines.
    Alice Johnson roll number is: 103 and department is: Electrical Engineering.
Processed 4 lines.
    Bob Brown roll number is: 104 and department is: Civil Engineering.
Processed 5 lines.
    Charlie White roll number is: 105 and department is: Information Technology.
Processed 6 lines.
```

9. Reverse a string using for loop in python.



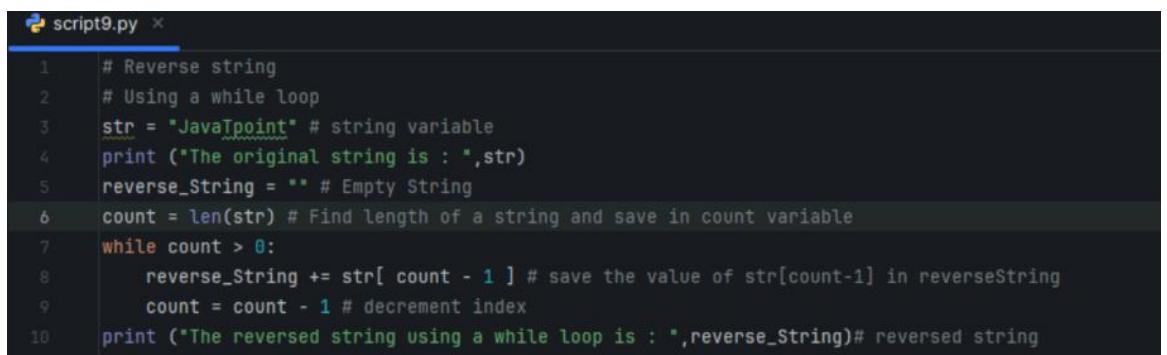
```
def reverse_string(str):  #usage
    str1 = "" # Declaring empty string to store the reversed string
    for i in str:
        str1 = i + str1
    return str1 # It will return the reverse string to the caller function
str = "JavaTpoint" # Given String
print("The original string is: ",str)
print("The reverse string is",reverse_string(str)) # Function call
```

Output:



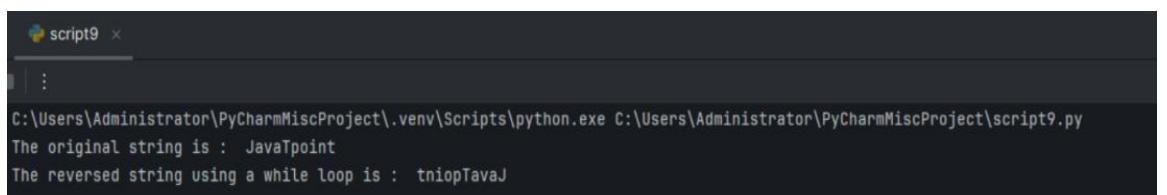
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script8.py
The original string is: JavaTpoint
The reverse string is tniopTavaJ
```

10. Reverse a string using while loop in python



```
# Reverse string
# Using a while loop
str = "JavaTpoint" # string variable
print ("The original string is : ",str)
reverse_String = "" # Empty String
count = len(str) # Find length of a string and save in count variable
while count > 0:
    reverse_String += str[ count - 1 ] # save the value of str[count-1] in reverseString
    count = count - 1 # decrement index
print ("The reversed string using a while loop is : ",reverse_String)# reversed string
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script9.py
The original string is : JavaTpoint
The reversed string using a while loop is : tniopTavaJ
```

11. Reverse a string using the slice ([]) operator in python

```
script10.py x
1 # Reverse a string
2 # using slice syntax
3 # reverse(str) Function to reverse a string
4 def reverse(str):
5     str = str[::-1]
6     return str
7 s = "JavaTpoint"
8 print ("The original string is : ",s)
9 print ("The reversed string using extended slice operator is : ",reverse(s))
```

Output:

```
script10 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script10.py
The original string is : JavaTpoint
The reversed string using extended slice operator is : tniopTavaJ
```

12. Reverse a string using reverse function with join in python

```
script11.py x
1 #reverse a string using reversed()
2 # Function to reverse a string
3 def reverse(str):
4     string = "".join(reversed(str)) # reversed() function inside the join() function
5     return string
6 s = "JavaTpoint"
7 print ("The original string is : ",s)
8 print ("The reversed string using reversed() is : ",reverse(s) )
```

Output:

```
script11 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script11.py
The original string is : JavaTpoint
The reversed string using reversed() is : tniopTavaJ
```

13. Reverse a string in using recursion function in python

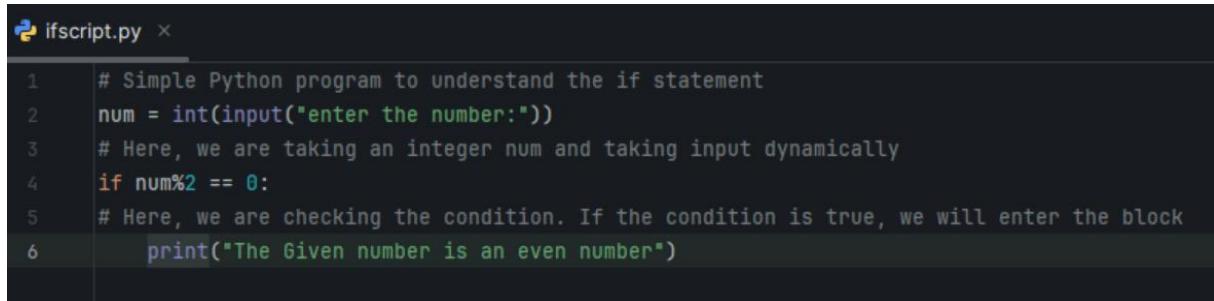
```
script12.py x
1 # reverse a string
2 # using recursion
3 def reverse(str):
4     if len(str) == 0:
5         return str
6     else:
7         return reverse(str[1:]) + str[0]
8 str = "Mamatha Eudulakanti"
9 print ("The original string is : ", str)
10 print ("The reversed string(using recursion) is : ", reverse(str))
```

Output:

```
script12 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\script12.py
The original string is : Mamatha Eudulakanti
The reversed string(using recursion) is : itnakaluduE ahtamaM
```

LOOPS

14. Python program to understand the ‘if statement’



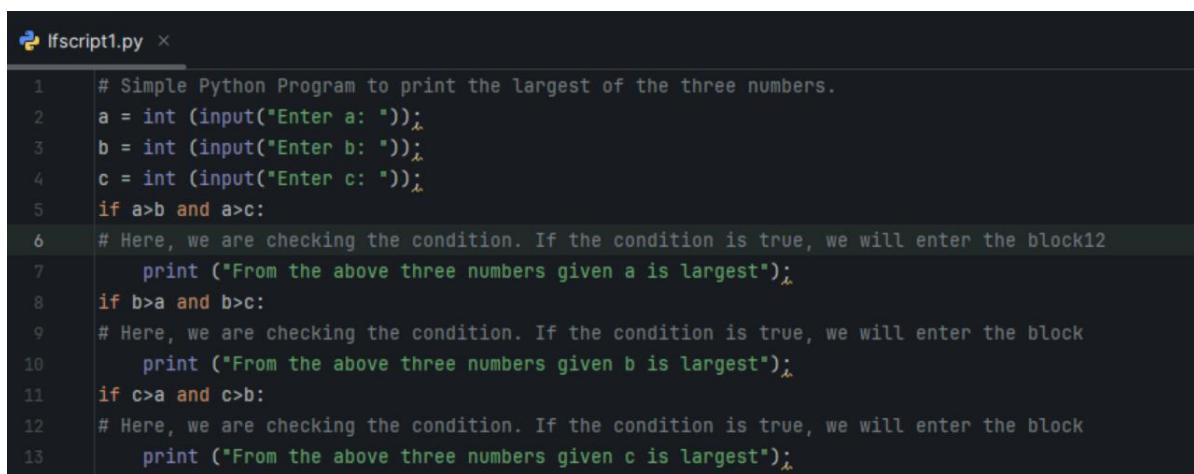
```
ifscript.py x
1 # Simple Python program to understand the if statement
2 num = int(input("enter the number:"))
3 # Here, we are taking an integer num and taking input dynamically
4 if num%2 == 0:
5     # Here, we are checking the condition. If the condition is true, we will enter the block
6     print("The Given number is an even number")
```

Output:



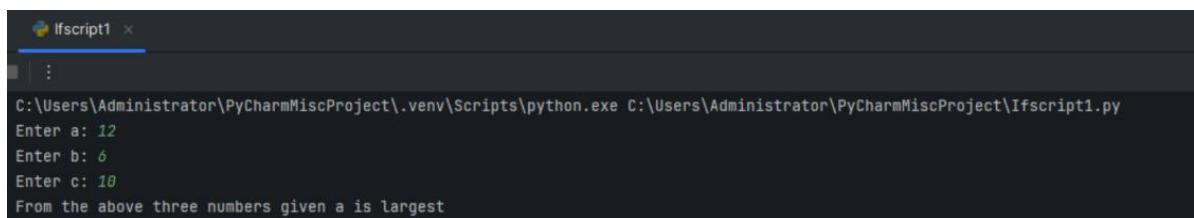
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\ifscript.py
enter the number:24
The Given number is an even number
```

15. Program to print the largest of the three numbers using ‘If Statement’



```
ifscript1.py x
1 # Simple Python Program to print the largest of the three numbers.
2 a = int (input("Enter a: "));
3 b = int (input("Enter b: "));
4 c = int (input("Enter c: "));
5 if a>b and a>c:
6     # Here, we are checking the condition. If the condition is true, we will enter the block12
7     print ("From the above three numbers given a is largest");
8 if b>a and b>c:
9     # Here, we are checking the condition. If the condition is true, we will enter the block
10    print ("From the above three numbers given b is largest");
11 if c>a and c>b:
12     # Here, we are checking the condition. If the condition is true, we will enter the block
13     print ("From the above three numbers given c is largest");
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\Ifscript1.py
Enter a: 12
Enter b: 6
Enter c: 10
From the above three numbers given a is largest
```

16. Program to check whether a person is eligible to vote or not using ‘If-else Statement’.

```
if-elsescript.py x
1 # Simple Python Program to check whether a person is eligible to vote or not.
2 age = int (input("Enter your age: "))
3 # Here, we are taking an integer num and taking input dynamically
4 if age>=18:
5     # Here, we are checking the condition. If the condition is true, we will enter the block
6     print("You are eligible to vote !!");
7 else:
8     print("Sorry! you have to wait !!");
```

Output:

```
if-elsescript x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\if-elsescript.py
Enter your age: 21
You are eligible to vote !!
```

17. Program to check whether a number is even or not using ‘If-else statement’.

```
if-elsescript1.py x
1 # Simple Python Program to check whether a number is even or not.
2 num = int(input("enter the number:"))
3 # Here, we are taking an integer num and taking input dynamically
4 if num%2 == 0:
5     # Here, we are checking the condition. If the condition is true, we will enter the block
6     print("The Given number is an even number")
7 else:
8     print("The Given Number is an odd number")
```

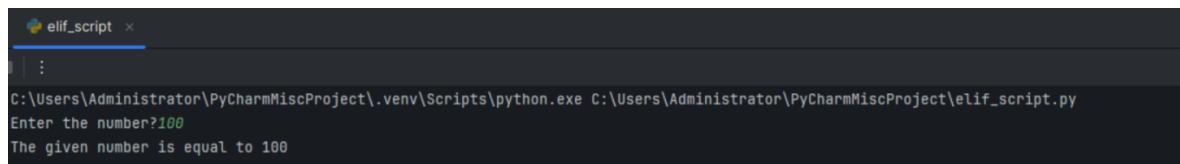
Output:

```
if-elsescript1 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\if-elsescript1.py
enter the number:24
The Given number is an even number
```

18. Simple Python program to understand ‘elif statement’.

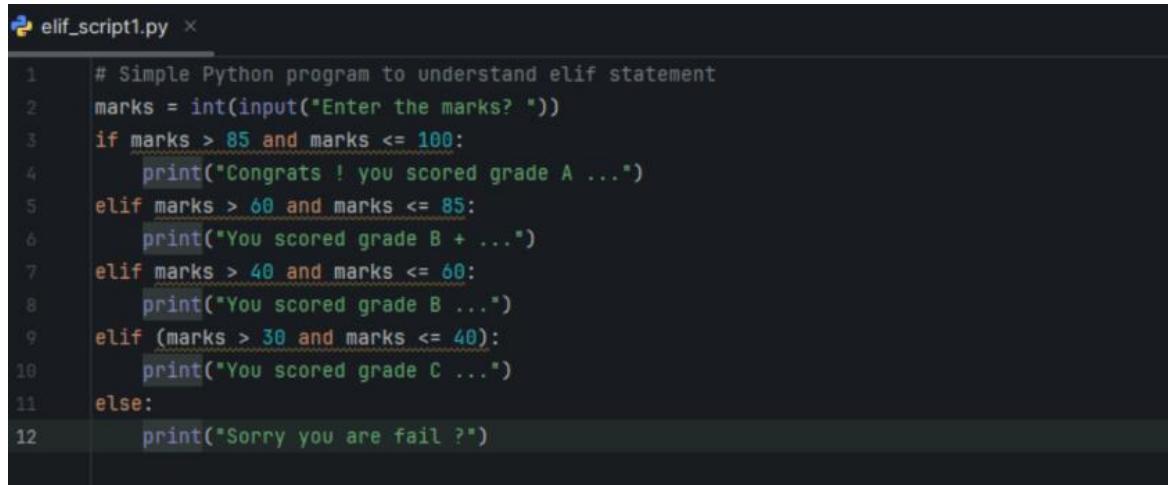
```
elif_script.py x
1 # Simple Python program to understand elif statement
2 number = int(input("Enter the number?"))
3 # Here, we are taking an integer number and taking input dynamically
4 if number==10:
5     # Here, we are checking the condition. If the condition is true, we will enter the block
6     print("The given number is equals to 10")
7 elif number==50:
8     # Here, we are checking the condition. If the condition is true, we will enter the block
9     print("The given number is equal to 50");
10 elif number==100:
11     # Here, we are checking the condition. If the condition is true, we will enter the block
12     print("The given number is equal to 100");
13 else:
14     print("The given number is not equal to 10, 50 or 100");
```

Output:



```
elif_script x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\elif_script.py
Enter the number?100
The given number is equal to 100
```

19. Python program to classify a student's grade based on their marks using 'elif statement'.



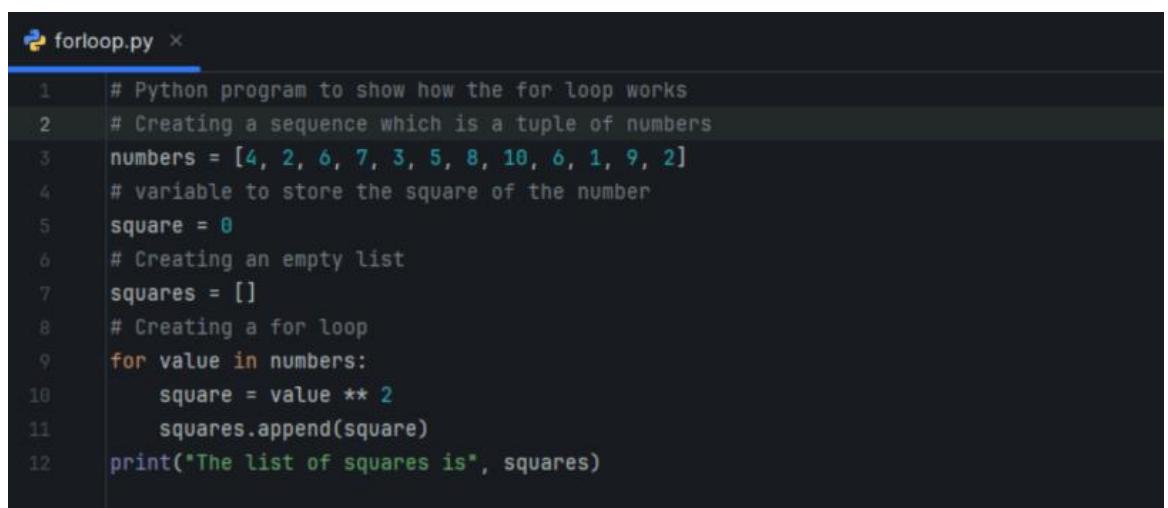
```
elif_script1.py x
1 # Simple Python program to understand elif statement
2 marks = int(input("Enter the marks? "))
3 if marks > 85 and marks <= 100:
4     print("Congrats ! you scored grade A ...")
5 elif marks > 60 and marks <= 85:
6     print("You scored grade B + ...")
7 elif marks > 40 and marks <= 60:
8     print("You scored grade B ...")
9 elif (marks > 30 and marks <= 40):
10    print("You scored grade C ...")
11 else:
12     print("Sorry you are fail ?")
```

Output:



```
elif_script1 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\elif_script1.py
Enter the marks? 96
Congrats ! you scored grade A ...
```

20. Python program that takes a list of numbers and creates a new list containing the square of each number using 'For Loop'.



```
forloop.py x
1 # Python program to show how the for loop works
2 # Creating a sequence which is a tuple of numbers
3 numbers = [4, 2, 6, 7, 3, 5, 8, 10, 6, 1, 9, 2]
4 # variable to store the square of the number
5 square = 0
6 # Creating an empty list
7 squares = []
8 # Creating a for loop
9 for value in numbers:
10    square = value ** 2
11    squares.append(square)
12 print("The list of squares is", squares)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\forloop.py
The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4]
```

21. Python program that loops through each character in a string. If the character is the letter 'o', print 'If block'. Otherwise, print the character."

```
# forloop1.py
# Python program to show how if-else statements work
string = "Python Loop"
# Initiating a loop
for s in string:
    if s == "o":
        print("If block")
    # if condition is not satisfied then else block will be executed
    else:
        print(s)
```

Output:

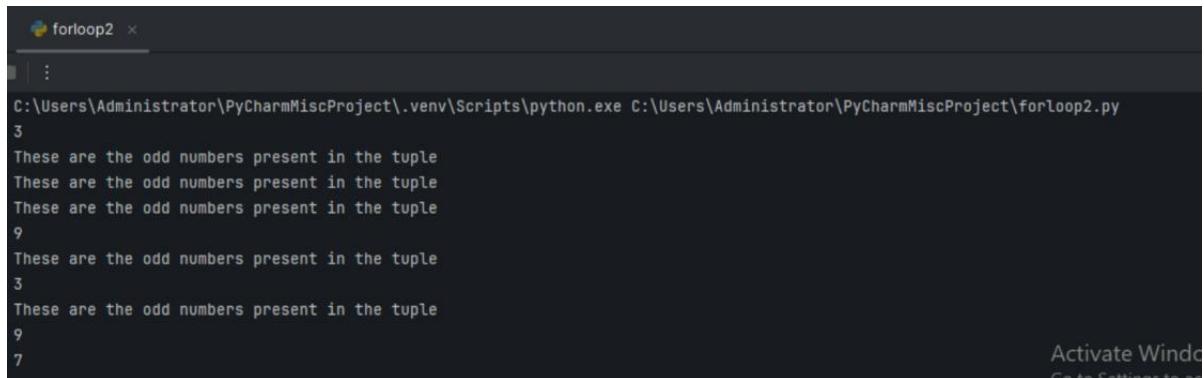
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\forloop1.py
P
y
t
h
If block
n

L
If block
If block
p
```

22. Python program to loop through a tuple and print all the odd numbers.

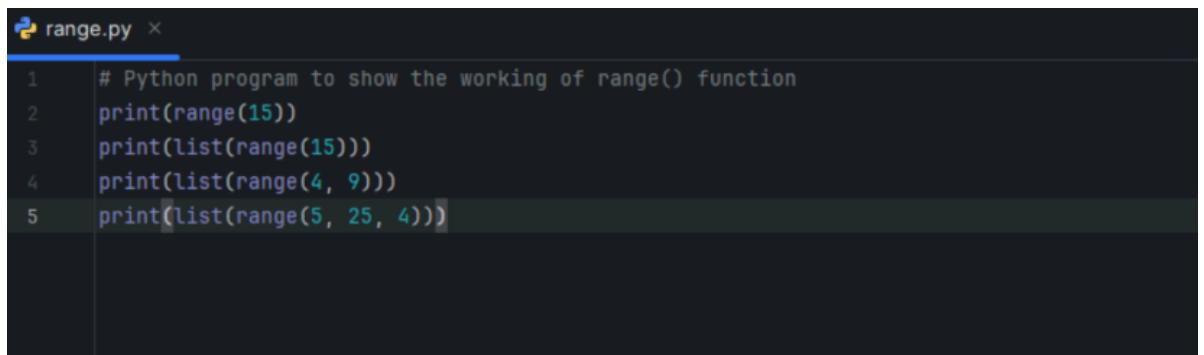
```
# forloop2.py
# Python program to show how to use else statement with for loop
# Creating a sequence
tuple_ = (3, 4, 6, 8, 9, 2, 3, 8, 9, 7)
# Initiating the loop
for value in tuple_:
    if value % 2 != 0:
        print(value)
    # giving an else statement
    else:
        print("These are the odd numbers present in the tuple")
```

Output:



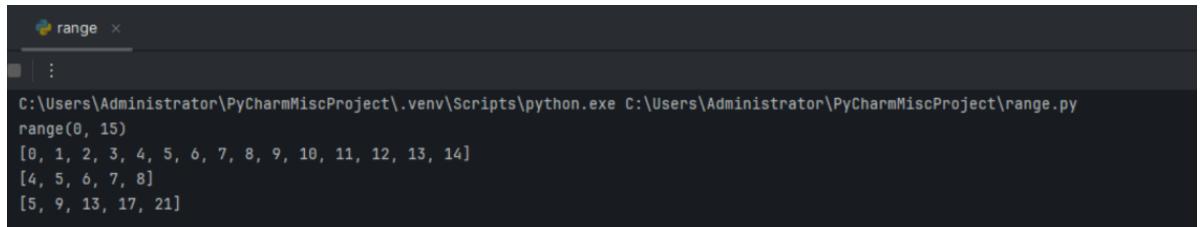
```
forloop2 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\forloop2.py
3
These are the odd numbers present in the tuple
These are the odd numbers present in the tuple
These are the odd numbers present in the tuple
9
These are the odd numbers present in the tuple
3
These are the odd numbers present in the tuple
9
7
```

23. Python program to demonstrate the usage of the range() function.



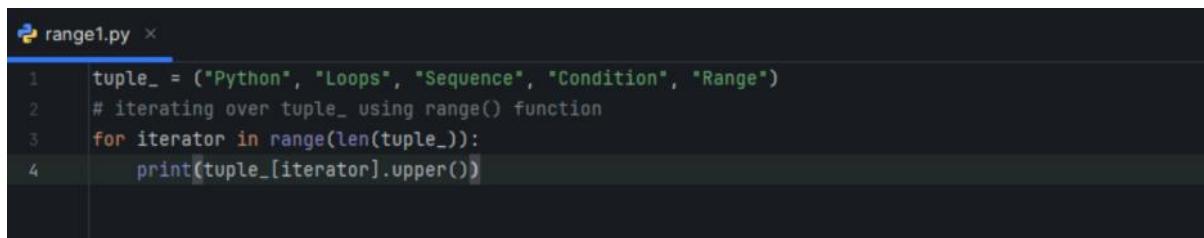
```
range.py x
1 # Python program to show the working of range() function
2 print(range(15))
3 print(list(range(15)))
4 print(list(range(4, 9)))
5 print(list(range(5, 25, 4)))
```

Output:



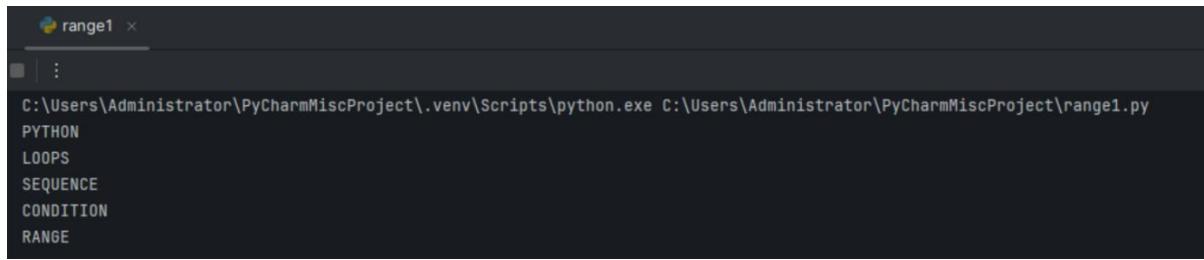
```
range x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\range.py
range(0, 15)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[4, 5, 6, 7, 8]
[5, 9, 13, 17, 21]
```

24. Python program to iterate through a tuple and print each element in uppercase. Use the range() function to iterate over the indices of the tuple.



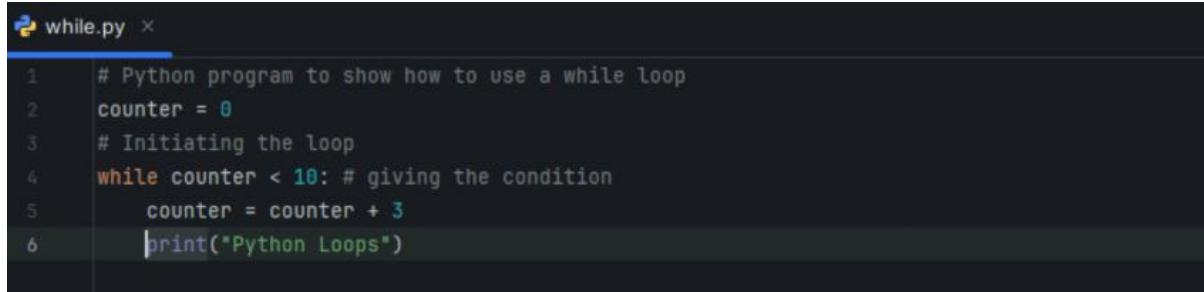
```
range1.py x
1 tuple_ = ("Python", "Loops", "Sequence", "Condition", "Range")
2 # iterating over tuple_ using range() function
3 for iterator in range(len(tuple_)):
4     print(tuple_[iterator].upper())
```

Output:



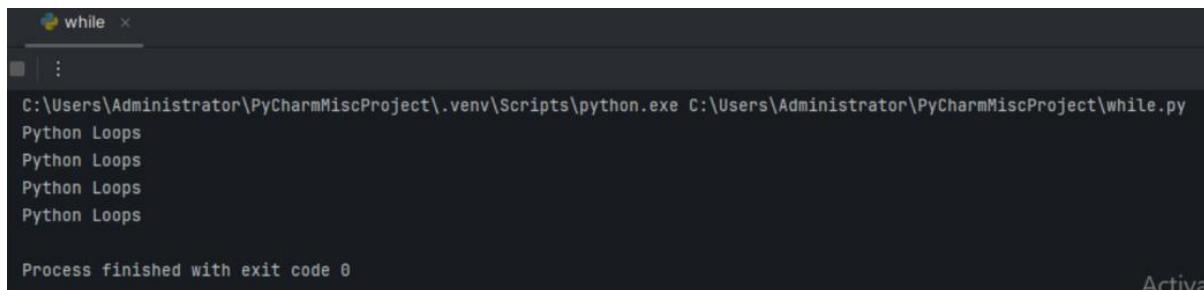
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\range1.py
PYTHON
LOOPS
SEQUENCE
CONDITION
RANGE
```

25. Python program that uses a while loop to print 'Python Loops' multiple times.



```
# Python program to show how to use a while loop
counter = 0
# Initiating the loop
while counter < 10: # giving the condition
    counter = counter + 3
    print("Python Loops")
```

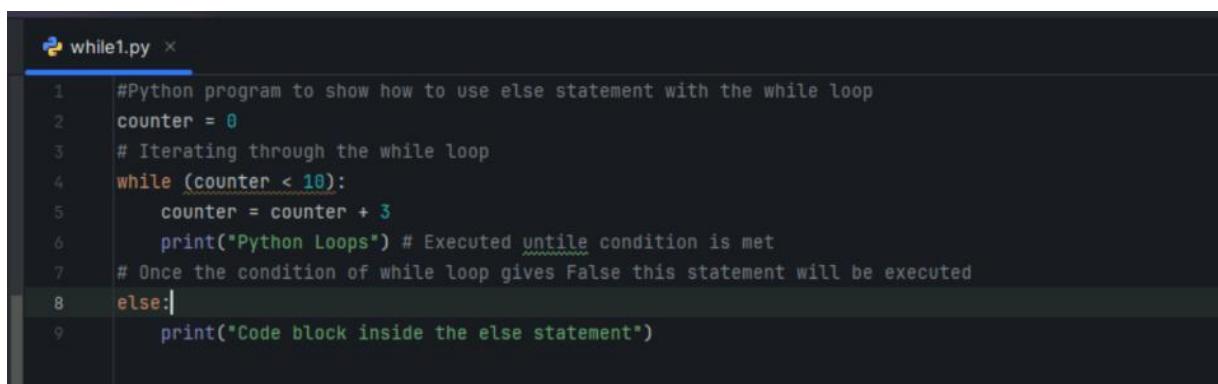
Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while.py
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops

Process finished with exit code 0
```

26. Python program using a while loop to print 'Python Loops' and then print a message from the else block after the loop ends.



```
# Python program to show how to use else statement with the while loop
counter = 0
# Iterating through the while loop
while (counter < 10):
    counter = counter + 3
    print("Python Loops") # Executed until condition is met
# Once the condition of while loop gives False this statement will be executed
else:
    print("Code block inside the else statement")
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while1.py
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Code block inside the else statement
```

27. Python program to print each character in 'Python Loops' except 'o', 'p', and 't' using continue statement.

```
# Python program to show how the continue statement works
# Initiating the loop
for string in "Python Loops":
    if string == "o" or string == "p" or string == "t":
        continue
    print('Current Letter:', string)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\continue.py
Current Letter: P
Current Letter: y
Current Letter: h
Current Letter: n
Current Letter: t
Current Letter: L
Current Letter: s
```

28. Python program to print each character in 'Python Loops' until the letter 'L' is encountered using break statement.

```
# Python program to show how the break statement works
# Initiating the loop
for string in "Python Loops":
    if string == 'L':
        break
    print('Current Letter: ', string)
```

Output:

```
break x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break.py
Current Letter: P
Current Letter: y
Current Letter: t
Current Letter: h
Current Letter: o
Current Letter: n
Current Letter:
```

29. Python program to demonstrate the use of the pass statement inside a loop.

```
pass.py x
1 # Python program to show how the pass statement works
2 for string in "Python Loops":
3     pass
4     print('Last Letter:', string)
```

Output:

```
pass x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\pass.py
Last Letter: s
```

30. Python program to calculate the sum of squares of all numbers in a list.

```
for.py x
1 # Code to find the sum of squares of each element of the list using for loop
2 # creating the list of numbers
3 numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8]
4 # initializing a variable that will store the sum
5 sum_ = 0
6 # using for loop to iterate over the list
7 for num in numbers:
8     sum_ = sum_ + num ** 2
9     print("The sum of squares is: ", sum_)
```

Output:

```
for x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\for.py
The sum of squares is: 774
```

31. Python program to iterate over a list and append each element plus 2 to the list, avoid modifying the list during iteration.

```
	range2.py x
1 my_list = [3, 5, 6, 8, 4]
2 for iter_var in range( len( my_list ) ):
3     my_list.append(my_list[iter_var] + 2)
4 print( my_list )
```

Output:

```
	range2 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\range2.py
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]
```

32. Python program to find the sum of squares of each element of the list using for loop.

```
	for1.py x
1 # Code to find the sum of squares of each element of the list using for loop
2 # creating the list of numbers
3 numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8]
4 # initializing a variable that will store the sum
5 sum_ = 0
6 # using for loop to iterate over list
7 for num in range( len(numbers) ):
8     sum_ = sum_ + numbers[num] ** 2
9 print("The sum of squares is: ", sum_)
```

Output:

```
	for1 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\for1.py
The sum of squares is: 774
```

33. Python program to look up and print the marks of a student from a dictionary, and handle the case when the student is not found.

```
# for2.py x
1 # Code to print marks of a student from the records
2 student_name_1 = 'Itika'
3 student_name_2 = 'Parker'
4 # Creating a dictionary of records of the students
5 records = {'Itika': 90, 'Arshia': 92, 'Peter': 46}
6 def marks( student_name ): 2 usages
7     for a_student in records: # for loop will iterate over the keys of the dictionary
8         if a_student == student_name:
9             return records[ a_student ]
10            break
11        else:
12            return f'There is no student of name {student_name} in the records'
13 # giving the function marks() name of two students
14 print( f"Marks of {student_name_1} are: ", marks( student_name_1 ) )
15 print( f"Marks of {student_name_2} are: ", marks( student_name_2 ) )
```

Output:

```
n for2 x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\for2.py
Marks of Itika are: 90
Marks of Parker are: There is no student of name Parker in the records
```

34. Python program to generate 11 random numbers between 0 and 11, and print the numbers that appear in both the range 0-10 and generate list.

```
n nested.py x
1 import random
2 numbers = [ ]
3 for val in range(0, 11):
4     numbers.append( random.randint( a: 0, b: 11 ) )
5 for num in range( 0, 11 ):
6     for i in numbers:
7         if num == i:
8             print( num, end = ' ' )
```

Output:

```
n nested x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\nested.py
0 0 1 6 6 6 7 9 10
Process finished with exit code 0
```

35. Python program to print the numbers from 1 to 10 in a single line, separated by spaces.

```
while2.py
```

```
1 i=1
2 while i<=10:
3     print(i, end=' ')
4     i+=1
```

Output:

```
while2
```

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while2.py
1 2 3 4 5 6 7 8 9 10
Process finished with exit code 0
```

36. Python program to print all numbers between 1 and 50 that are divisible by 5 or 7, separated by spaces.

```
while3.py
```

```
1 i=1
2 while i < 51:
3     if i%5 == 0 or i%7==0 :
4         print(i, end=' ')
5     i+=1
```

Output:

```
while3
```

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while3.py
5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50
Process finished with exit code 0
```

37. Python program to calculate the sum of squares of all integers from 1 to 15.

```
# Python program example to show the use of while loop
num = 15
# initializing summation and a counter for iteration
summation = 0
c = 1
while c <= num: # specifying the condition of the loop
    # begining the code block
    summation = c**2 + summation
    c = c + 1 # incrementing the counter
# print the final sum
print("The sum of squares is", summation)
```

Output:

```
Run while4 x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while4.py
The sum of squares is 1240
```

38. Python program to check if each number in a list is prime or not.

```
num = [34, 12, 54, 23, 75, 34, 11]
def prime_number(number):
    usage
    condition = 0
    iteration = 2
    while iteration <= number / 2:
        if number % iteration == 0:
            condition = 1
            break
        iteration = iteration + 1
    if condition == 0:
        print(f"{number} is a PRIME number")
    else:
        print(f"{number} is not a PRIME number")
for i in num:
    prime_number(i)
```

Output:

```
while5 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while5.py
34 is not a PRIME number
12 is not a PRIME number
54 is not a PRIME number
23 is a PRIME number
75 is not a PRIME number
34 is not a PRIME number
11 is a PRIME number
```

39. Python program to check if a number is an Armstrong number.

```
while6.py x
1 n = int(input("Enter a number: 300")) # Input number
2 n1 = str(n) # Convert number to string to get the length (number of digits)
3 l = len(n1) # Number of digits in the number
4 temp = n # Store original number
5 s = 0 # Initialize sum of powers
6
7 # Loop through each digit in the number
8 while n != 0:
9     r = n % 10 # Get the last digit
10    s = s + (r ** l) # Raise the digit to the power of the number of digits and add to sum
11    n = n // 10 # Remove the last digit
12
13 # Check if the sum is equal to the original number
14 if s == temp:
15     print("It is an Armstrong number")
16 else:
17     print("It is not an Armstrong number")
```

Output:

```
while6 x
:
Enter a number: 342
It is not an Armstrong number
```

40. Python program to print the multiplication table of a given number.

```
while7.py x
1 num = 21
2 counter = 1
3 # we will use a while loop for iterating 10 times for the multiplication table
4 print("The Multiplication Table of: ", num)
5 while counter <= 10: # specifying the condition
6     ans = num * counter
7     print (num, 'x', counter, '=', ans)
8     counter += 1 # expression to increment the counter
```

Output:

```
while7 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while7.py
The Multiplication Table of: 21
21 x 1 = 21
21 x 2 = 42
21 x 3 = 63
21 x 4 = 84
21 x 5 = 105
21 x 6 = 126
21 x 7 = 147
21 x 8 = 168
21 x 9 = 189
21 x 10 = 210
Activate
```

41. Python program that squares each element of a list and stores the results in a new list using a while loop.

```
while8.py x
1 list_ = [3, 5, 1, 4, 6]
2 squares = []
3 # programing a while loop
4 while list_: # until list is not empty this expression will give boolean True after that False
5     squares.append( (list_.pop())**2)
6 # Print the squares of all numbers.
7 print( squares )
```

Output:

```
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while8.py
[36, 16, 1, 25, 9]
```

42. Python program to check if each element in a list is even or odd and print the appropriate message.

```
while9.py x
1 list_ = [3, 4, 8, 10, 34, 45, 67, 80] # Initialize the list
2 index = 0
3 while index < len(list_):
4     element = list_[index]
5     if element % 2 == 0:
6         print('It is an even number') # Print if the number is even.
7     else:
8         print('It is an odd number') # Print if the number is odd.
9     index += 1
```

Output:

```
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while9.py
It is an odd number
It is an even number
It is an odd number
It is an odd number
It is an even number
```

43. Python program to print the length of each element in a list.

```
while10.py
```

```
1 list_=['Priya','Neha','Cow','To']
2 index=0
3 while index<len(list_):
4     element=list_[index]
5     print(len(element))
6     index+=1
```

Output:

```
while10
```

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while10.py
5
4
3
2
```

44. Python program to modify two numbers using a while loop with conditions.

```
while11.py
```

```
1 num1=17
2 num2=-12
3 while num1 >5 and num2 < -5:
4     num1 -=2
5     num2 +=3
6     print((num1,num2))
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while11.py
(15, -9)
(13, -6)
(11, -3)
```

45. Python program to modify two numbers using a while loop with OR operator.

```
while12.py
1 num1=17
2 num2=-12
3 while num1 >5 or num2 < -5:
4     num1 -=2
5     num2 +=3
6     print((num1,num2))
```

Output:

```
while12
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while12.py
(15, -9)
(13, -6)
(11, -3)
(9, 0)
(7, 3)
(5, 6)
```

46. Python program to print the number of iterations using a while loop with multiple conditions.

```
while13.py
1 num1=9
2 num2=14
3 maximum_value=4
4 counter=0
5 while(counter < num1 or counter < num2) and not counter >= maximum_value:
6     print(f"Number of iterations: {counter}")
7     counter+=1
```

Output:

```
while13
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\while13.py
Number of iterations: 0
Number of iterations: 1
Number of iterations: 2
Number of iterations: 3
```

47. Python program to iterate over the string 'While Loops' and skip the vowels using a continue statement.

```
continue1.py
```

```
1 for string in "While Loops":  
2     if string == "o" or string == "i" or string == "e":  
3         continue  
4     print('Current Letter:', string)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\continue1.py  
Current Letter: W  
Current Letter: h  
Current Letter: l  
Current Letter:  
Current Letter: L  
Current Letter: p  
Current Letter: s
```

48. Python program to iterate over the string 'Python Loops' and stop the loop when the letter 'n' is encountered.

```
break1.py
```

```
1 for string in "Python Loops":  
2     if string == "n":  
3         break  
4     print('Current Letter:', string)
```

Output:

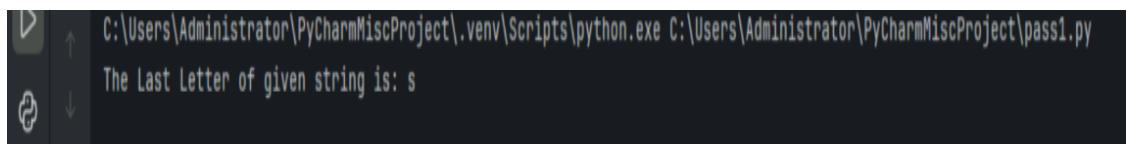
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break1.py  
Current Letter: P  
Current Letter: y  
Current Letter: t  
Current Letter: h  
Current Letter: o
```

49. Python program to iterate over the string 'Python Loops' using the pass statement and print the last letter.

```
pass1.py
```

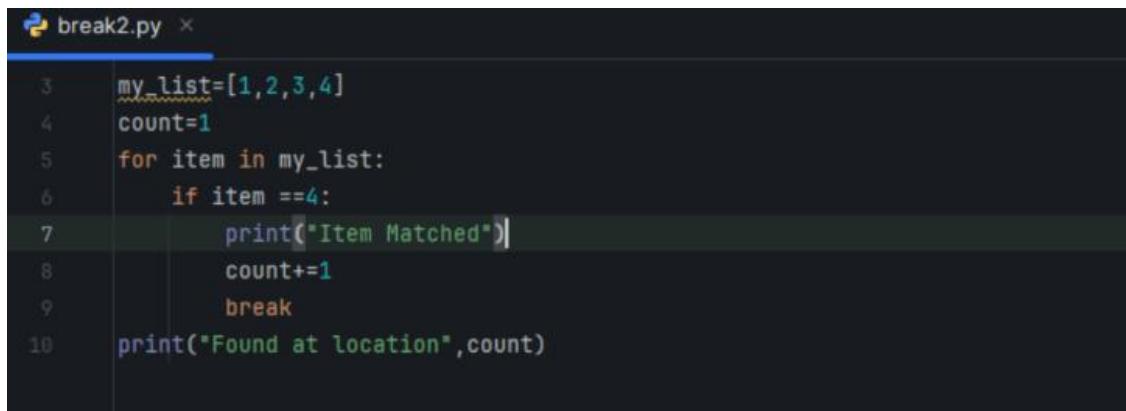
```
1 for string in "Python Loops":  
2     pass  
3     print('The Last Letter of given string is:',string)
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\pass1.py
The Last Letter of given string is: s
```

50. Python program to break the loop when a specific item is found in a list



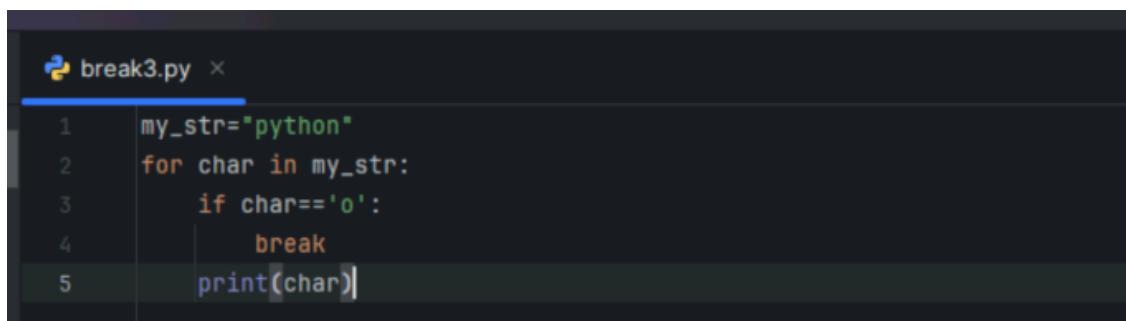
```
break2.py
1 my_list=[1,2,3,4]
2 count=1
3 for item in my_list:
4     if item ==4:
5         print("Item Matched")
6         count+=1
7         break
8
9 print("Found at location",count)
```

Output:



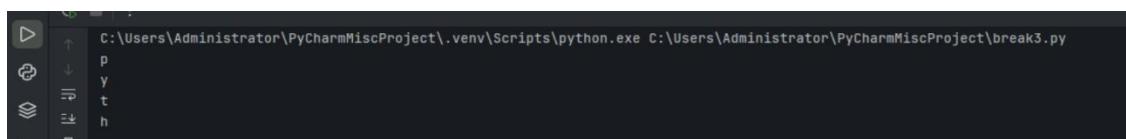
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break2.py
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]
Item Matched
Found at location 2
```

51. Python program to break the loop when the character 'o' is found in the string 'python'.



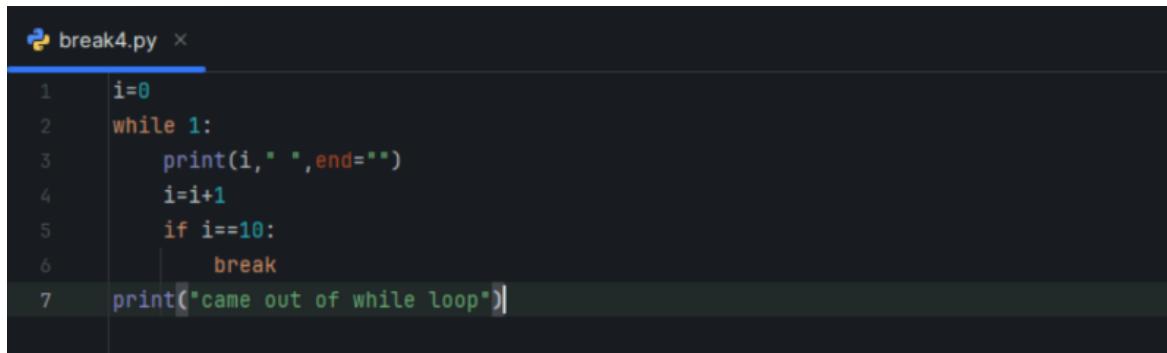
```
break3.py
1 my_str="python"
2 for char in my_str:
3     if char=='o':
4         break
5     print(char)
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break3.py
p
y
t
h
```

52. Python program to exit a while loop when a variable equals 10.



```
i=0
while 1:
    print(i, " ",end="")
    i=i+1
    if i==10:
        break
print("came out of while loop")
```

Output:



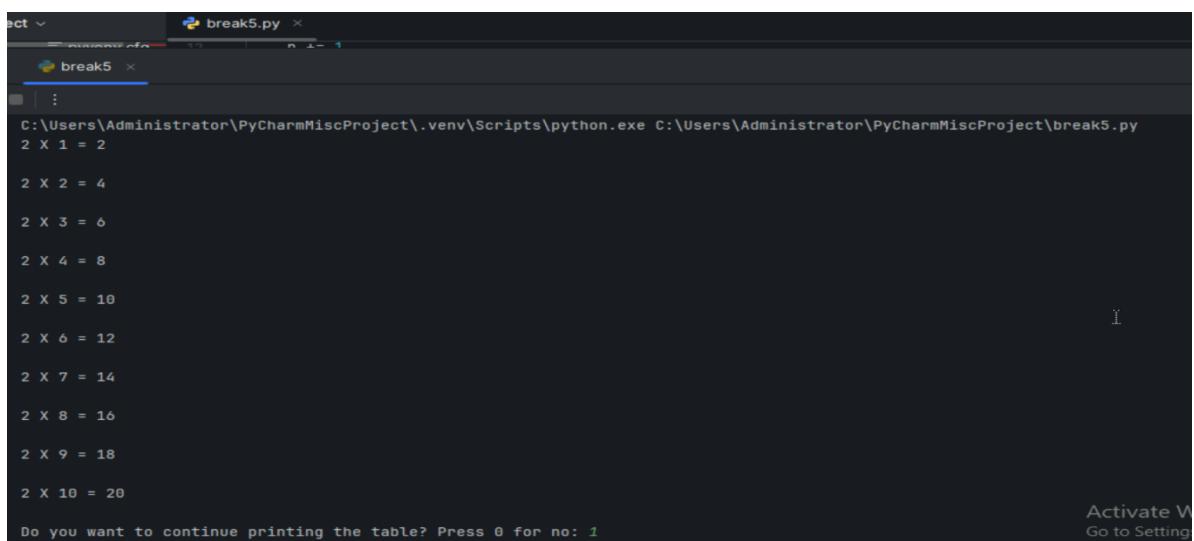
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break4.py
0 1 2 3 4 5 6 7 8 9 came out of while loop
```

53. Python program to print multiplication tables and exit when the user chooses to stop.



```
# break statement example
n = 2
while True:
    i = 1
    while i <= 10:
        print("%d X %d = %d\n" % (n, i, n * i))
        i += 1
    choice = int(input("Do you want to continue printing the table? Press 0 for no: "))
    if choice == 0:
        print("Exiting the program...")
        break
    n += 1
print("Program finished successfully.")
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\break5.py
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: 1
```

```
PC PyCharmMiscProject Version control
break5.py

Do you want to continue printing the table? Press 0 for no: 1
3 X 1 = 3

3 X 2 = 6

3 X 3 = 9

3 X 4 = 12

3 X 5 = 15

3 X 6 = 18

3 X 7 = 21

3 X 8 = 24

3 X 9 = 27

3 X 10 = 30

Do you want to continue printing the table? Press 0 for no: 0

Activate Windows
Go to Settings to activate Windows.

Do you want to continue printing the table? Press 0 for no: 0
Exiting the program...
Program finished successfully.
```

54. Python program to loop from 10 to 20 and skip printing 15 using the continue statement.

```
continue2.py

1 # looping from 10 to 20
2 for iterator in range(10, 21):
3     # If iterator is equals to 15, loop will continue to the next iteration
4     if iterator == 15:
5         continue
6     # otherwise printing the value of iterator
7     print(iterator)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\continue2.py
10
11
12
13
14
15
16
17
18
19
20
```

55. Python program to loop through a string and skip the letter 'a'.

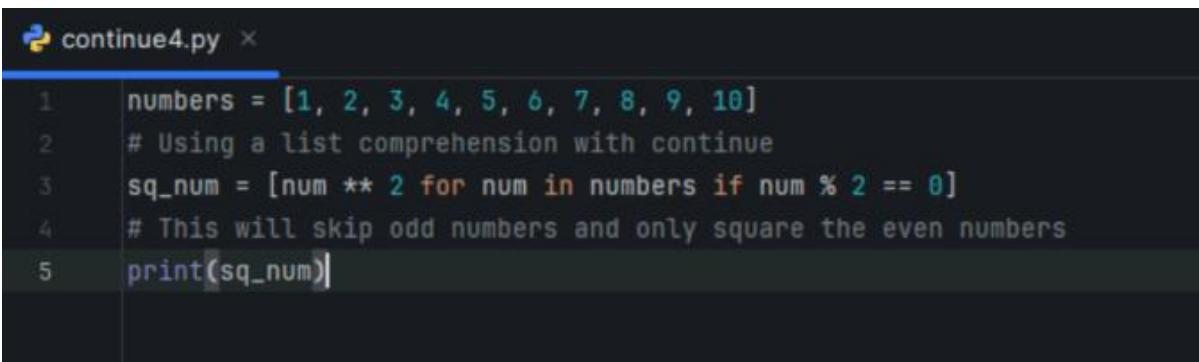
```
continue3.py

1     string = "JavaTpoint"
2     iterator = 0
3     while iterator < len(string):
4         if string[iterator] == 'a':
5             iterator += 1
6             continue
7         print(string[iterator])
8         iterator += 1
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\continue3.py
J
v
T
P
o
i
n
t
```

56. Python program to square only the even numbers in a list using list comprehension.



```
continue4.py ×

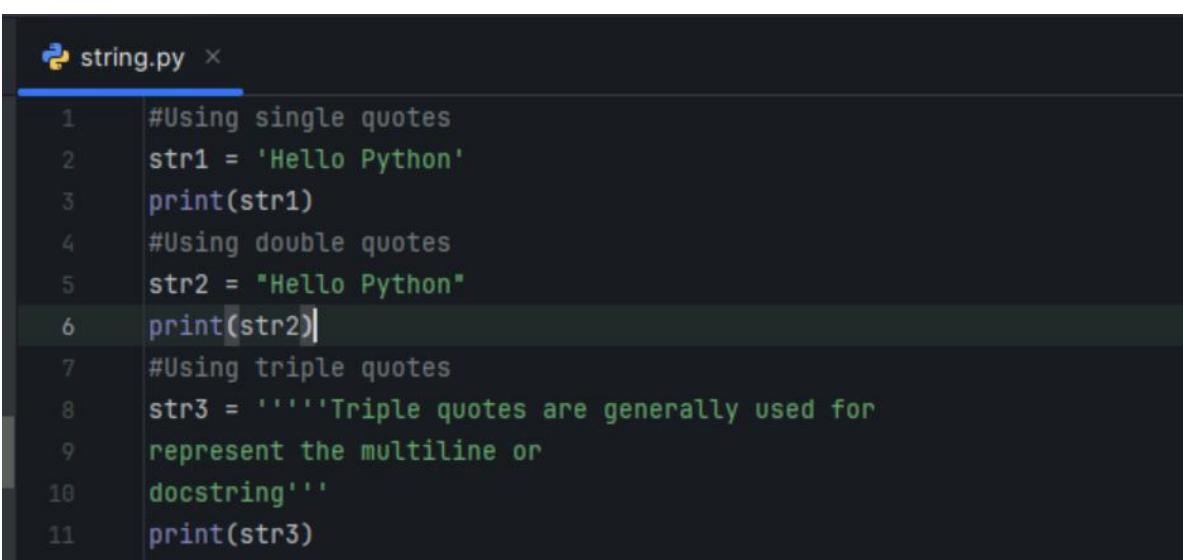
1  numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2  # Using a list comprehension with continue
3  sq_num = [num ** 2 for num in numbers if num % 2 == 0]
4  # This will skip odd numbers and only square the even numbers
5  print(sq_num)
```

Output:

```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\continue4.py
[4, 16, 36, 64, 100]
```

STRING

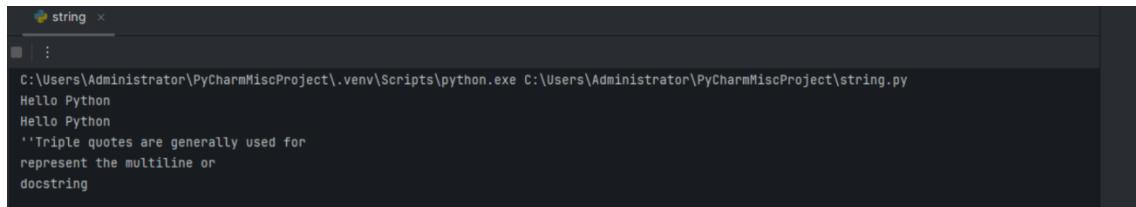
57. Create a string by enclosing the characters in single-quotes or double-quotes.



```
string.py ×

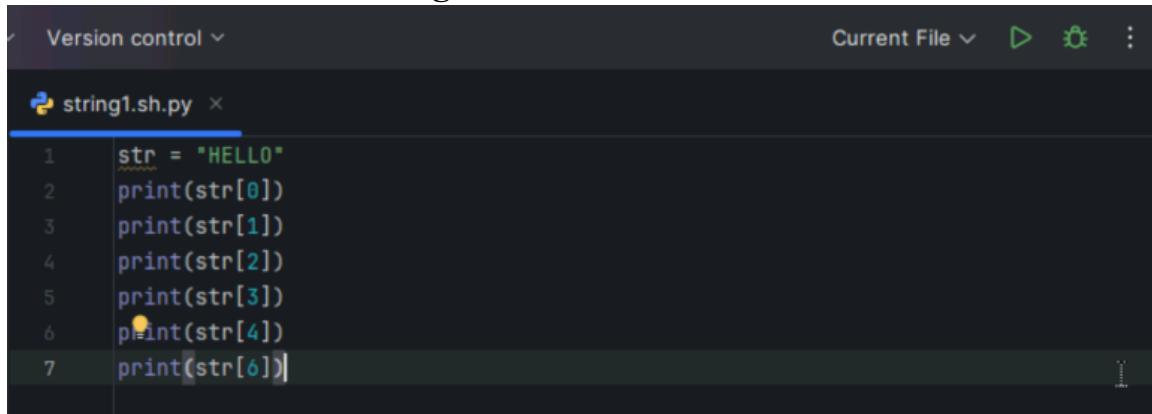
1  #Using single quotes
2  str1 = 'Hello Python'
3  print(str1)
4  #Using double quotes
5  str2 = "Hello Python"
6  print(str2)
7  #Using triple quotes
8  str3 = '''Triple quotes are generally used for
9  represent the multiline or
10 docstring'''
11 print(str3)
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string.py
Hello Python
Hello Python
'''Triple quotes are generally used for
represent the multiline or
docstring
```

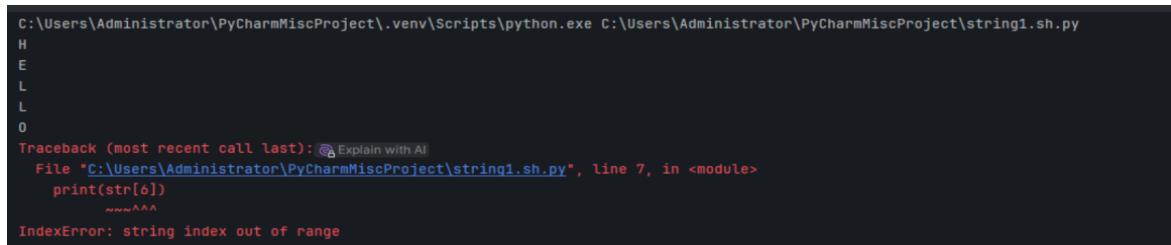
58. Python program to access each character of a string and handle the Index Error when accessing an invalid index.



```
Version control Current File ▶ ⚡ ⋮
string1.sh.py ×

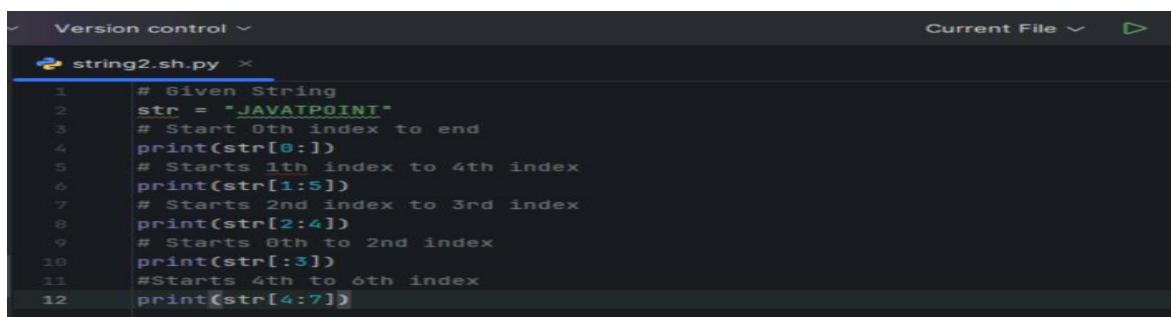
1 str = "HELLO"
2 print(str[0])
3 print(str[1])
4 print(str[2])
5 print(str[3])
6 print(str[4])
7 print(str[6])
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string1.sh.py
H
E
L
L
O
Traceback (most recent call last): Explain with AI
  File "C:\Users\Administrator\PyCharmMiscProject\string1.sh.py", line 7, in <module>
    print(str[6])
               ^
IndexError: string index out of range
```

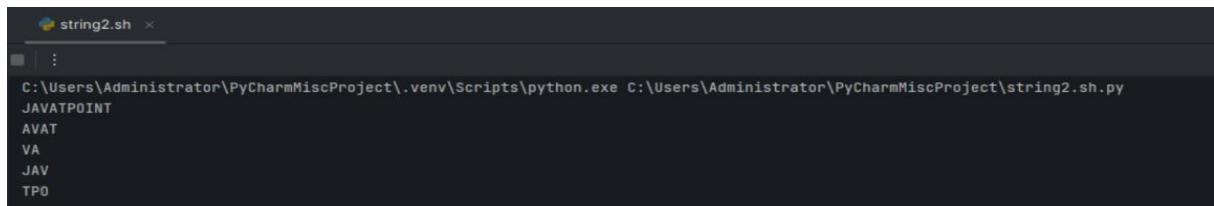
59. Python program to slice a string in different ways using index ranges



```
Version control Current File ▶ ⚡ ⋮
string2.sh.py ×

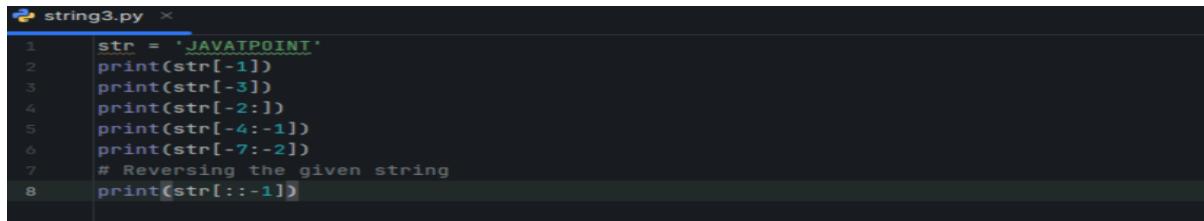
1 # Given String
2 str = "JAVATPOINT"
3 # Start 0th index to end
4 print(str[0:])
5 # Starts 1th index to 4th index
6 print(str[1:5])
7 # Starts 2nd index to 3rd index
8 print(str[2:4])
9 # Starts 0th to 2nd index
10 print(str[:3])
11 #Starts 4th to 6th index
12 print(str[4:7])
```

Output:



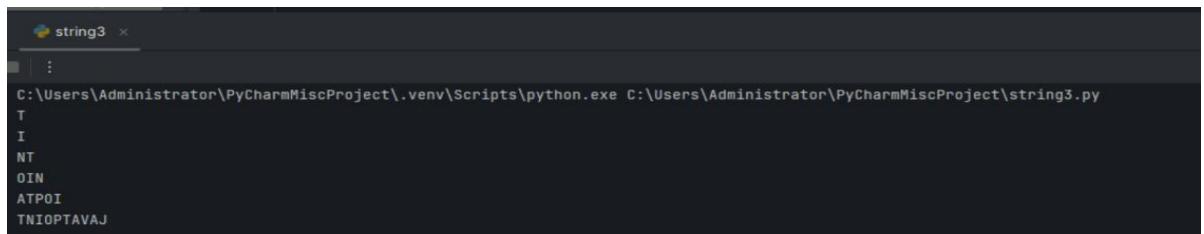
```
string2.sh ×
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string2.sh.py
JAVATPOINT
AVAT
VA
JAV
TPO
```

60. Python program to access elements from a string using negative indexing and slice the string in various ways.



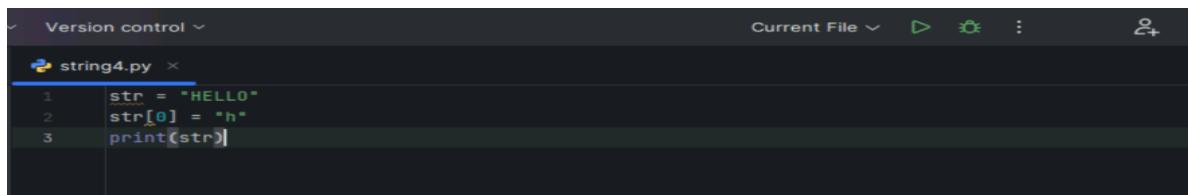
```
string3.py
1 str = 'JAVATPOINT'
2 print(str[-1])
3 print(str[-3])
4 print(str[-2:])
5 print(str[-4:-1])
6 print(str[-7:-2])
7 # Reversing the given string
8 print(str[::-1])
```

Output:



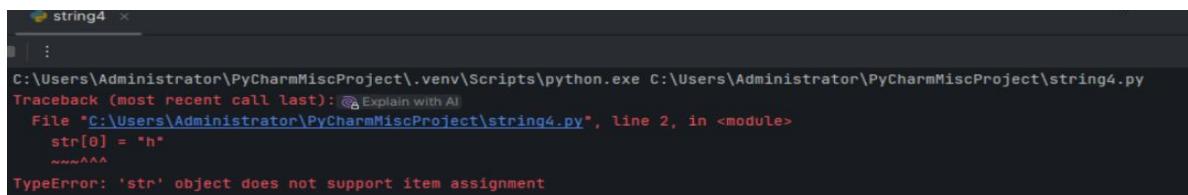
```
string3
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string3.py
T
I
N
O
A
P
T
O
V
A
J
```

61. Python program to try modifying a character of a string.



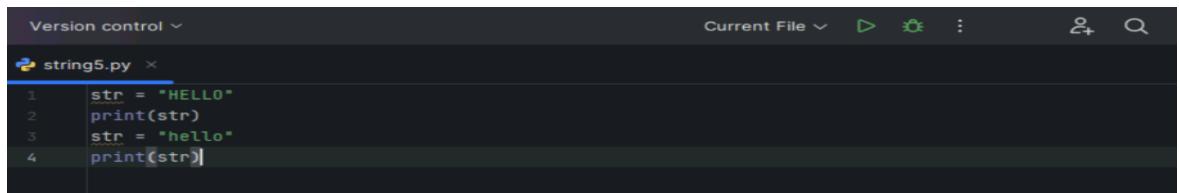
```
string4.py
1 str = "HELLO"
2 str[0] = "h"
3 print(str)
```

Output:



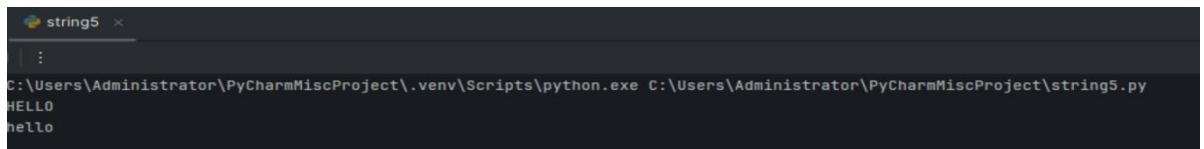
```
string4
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string4.py
Traceback (most recent call last):
  File "C:\Users\Administrator\PyCharmMiscProject\string4.py", line 2, in <module>
    str[0] = "h"
    ^^^^^^
TypeError: 'str' object does not support item assignment
```

62. Python program to modify a character in a Python string



```
string5.py
1 str = "HELLO"
2 print(str)
3 str = "hello"
4 print(str)
```

Output:



```
string5
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string5.py
HELLO
hello
```

63. Python program to delete the string.

```
Version control ~
string6.py ×
1 str = "JAVATPOINT"
2 del str[1]
```

Output:

```
string6 ×
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string6.py
Traceback (most recent call last): Explain with AI
  File "C:\Users\Administrator\PyCharmMiscProject\string6.py", line 2, in <module>
    del str[1]
    ^^^^
TypeError: 'str' object doesn't support item deletion
```

64. Python program to delete the entire string.

```
Version control ~
string7.py ×
1 str1 = "JAVATPOINT"
2 del str1
3 print(str1)
```

Output:

```
string7 ×
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string7.py
Traceback (most recent call last): Explain with AI
  File "C:\Users\Administrator\PyCharmMiscProject\string7.py", line 3, in <module>
    print(str1)
    ^^^^
NameError: name 'str1' is not defined. Did you mean: 'str'?
```

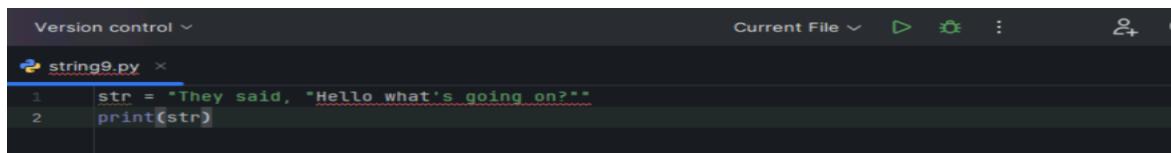
65. Python program to understand the real use of Python operators.

```
Version control ~
Current File ~
string8.py ×
1 str = "Hello"
2 str1 = " world"
3 print(str*3) # prints HelloHelloHello
4 print(str+str1) # prints Hello world
5 print(str[4]) # prints o
6 print(str[2:4]) # prints ll
7 print('w' in str) # prints false as w is not present in str
8 print('wo' not in str1) # prints false as wo is present in str1.
9 print(r'C://python37') # prints C://python37 as it is written
10 print("The string str : %s"%(str)) # prints The string str : Hello
```

Output:

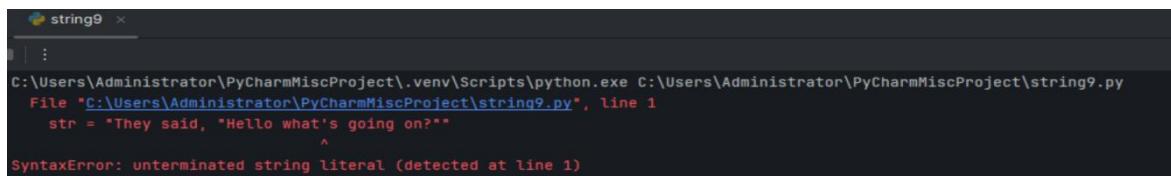
```
string8 ×
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string8.py
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello
```

66. Python program to understand the string formatting.



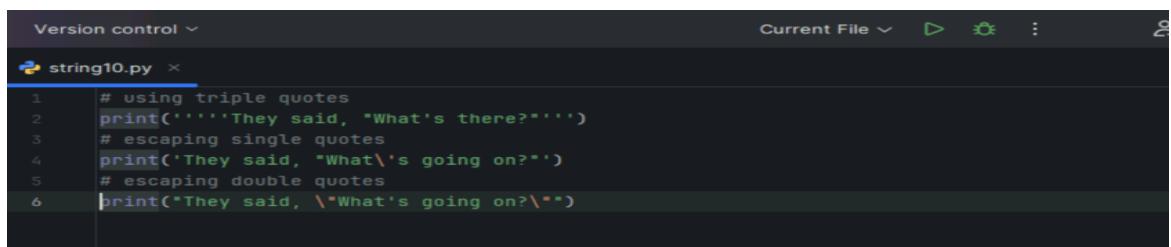
```
Version control Current File > ⚡ ⋮ +  
string9.py  
1 str = "They said, "Hello what's going on?""  
2 print(str)
```

Output:



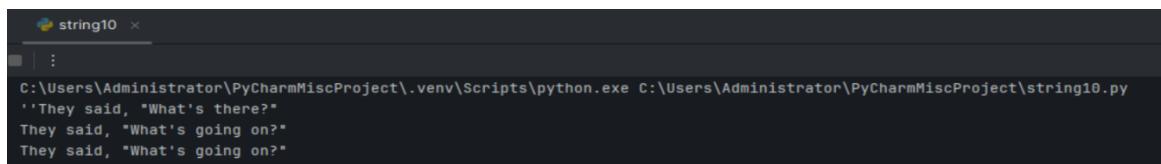
```
string9  
:  
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string9.py  
File "C:\Users\Administrator\PyCharmMiscProject\string9.py", line 1  
    str = "They said, "Hello what's going on?"  
          ^  
SyntaxError: unterminated string literal (detected at line 1)
```

67. Python code to handle quotes in strings



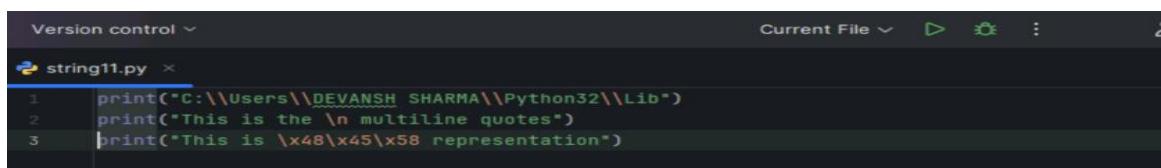
```
Version control Current File > ⚡ ⋮ +  
string10.py  
1 # using triple quotes  
2 print('''They said, "What's there?'''')  
3 # escaping single quotes  
4 print('They said, "What\'s going on?"')  
5 # escaping double quotes  
6 print("They said, \"What's going on?\"")
```

Output:



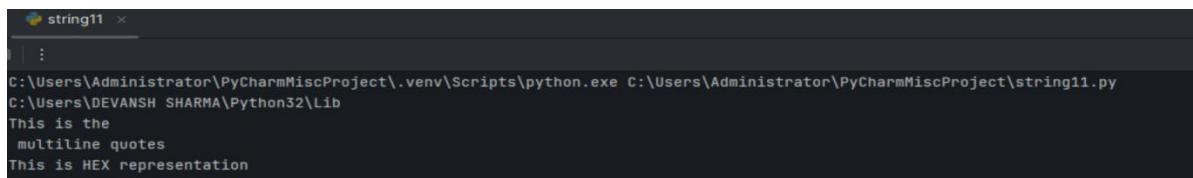
```
string10  
:  
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string10.py  
'They said, "What's there?"  
They said, "What's going on?"  
They said, "What's going on?"
```

68. Python program to perform escape sequence in strings



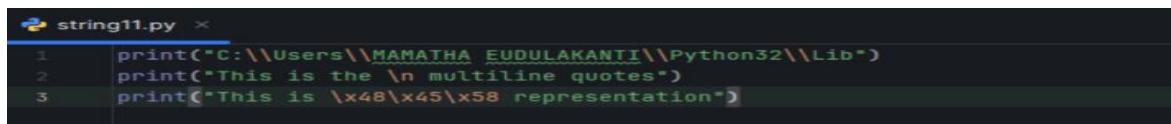
```
Version control Current File > ⚡ ⋮ +  
string11.py  
1 print("C:\\\\Users\\\\DEVANSH SHARMA\\\\Python32\\\\Lib")  
2 print("This is the \\n multiline quotes")  
3 print("This is \\x48\\x45\\x58 representation")
```

Output:



```
string11  
:  
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string11.py  
C:\\\\Users\\\\DEVANSH SHARMA\\\\Python32\\\\Lib  
This is the  
multiline quotes  
This is HEX representation
```

69. Python code to demonstrate different ways to format strings using curly braces {}.



```
string11.py  
1 print("C:\\\\Users\\\\MAMATHA EUDULAKANTI\\\\Python32\\\\Lib")  
2 print("This is the \\n multiline quotes")  
3 print("This is \\x48\\x45\\x58 representation")
```

Output:

```
Run string11 ×  
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string11.py  
C:\Users\MAMATHA_EUDULAKANTI\Python32\Lib  
This is the  
    multiline quotes  
This is HEX representation
```

70. Python code to demonstrate different ways to format strings using curly braces {}.

```
string12.py x
1 # Using Curly braces
2 print("{} and {} both are the best friend".format( "args: "Devansh", "Abhishek"))
3 #Positional Argument
4 print("{} and {} best players ".format( "args: "Virat", "Rohit"))
5 #Keyword Argument
6 print("{a},{b},{c}".format(a = "James", b = "Peter", c = "Ricky"))
```

Output:

```
string12 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string12.py
Devansh and Abhishek both are the best friend
Rohit and Virat best players
James,Peter,Ricky
```

71. Python String Formatting Using % Operator.

The screenshot shows a code editor window with the following details:

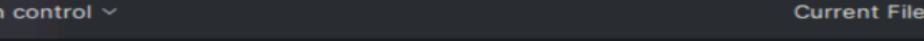
- Version control:** Version control dropdown.
- Current File:** Current File dropdown.
- File Operations:** Save (green arrow), Undo (left arrow), Redo (right arrow), Copy (copy icon), Paste (paste icon), Find (magnifying glass), Replace (crossed-out magnifying glass), and Delete (trash bin).
- Code Area:** The file is named "string13.py". The code contains:

```
1 Integer = 10;
2 Float = 1.290
3 String = "Devansh"
4 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%(In
```
- Status Bar:** Status bar showing line 1, column 1, and a warning icon with "1" and "1" next to it.

Output:

```
string13 x
:
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\string13.py
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
```

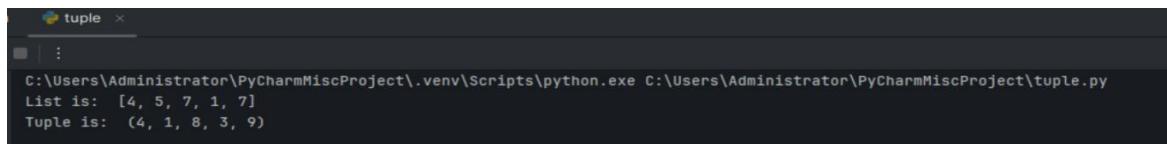
72. Python code to show the difference between creating a list and a tuple



A screenshot of a code editor window titled "tuple.py". The code defines a list and a tuple, then prints them both.

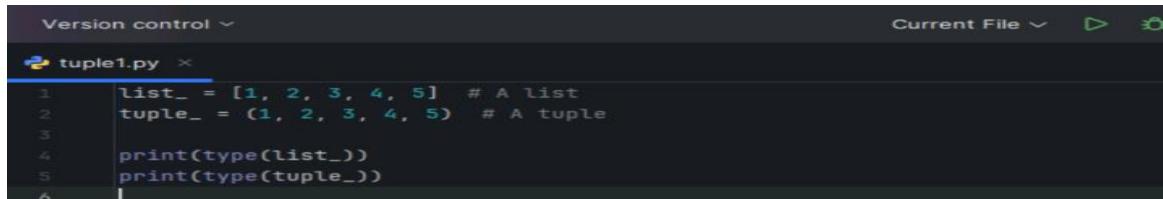
```
list_ = [4, 5, 7, 1, 7]
tuple_ = (4, 1, 8, 3, 9)
print("List is: ", list_)
print("Tuple is: ", tuple_)
```

Output:



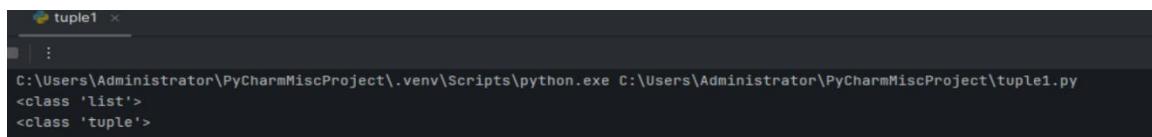
```
tuple x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple.py
List is: [4, 5, 7, 1, 9]
Tuple is: (4, 1, 8, 3, 9)
```

73. Python code to print the data type of the data structure using the type() function



```
tuple1.py x
1 list_ = [1, 2, 3, 4, 5] # A list
2 tuple_ = (1, 2, 3, 4, 5) # A tuple
3
4 print(type(list_))
5 print(type(tuple_))
```

Output:



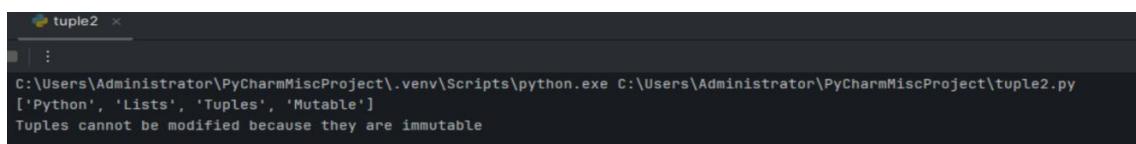
```
tuple1 x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple1.py
<class 'list'>
<class 'tuple'>
```

74. Python code to Update the element of list and tuple at a particular index



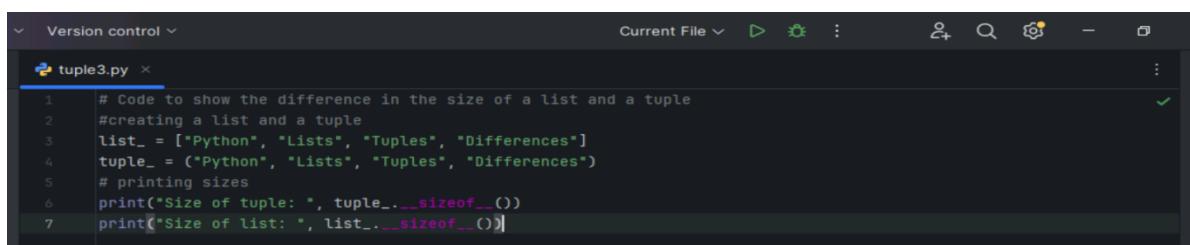
```
tuple2.py x
1 # Updating the element of list and tuple at a particular index
2 # creating a list and a tuple
3 list_ = ["Python", "Lists", "Tuples", "Differences"]
4 tuple_ = ("Python", "Lists", "Tuples", "Differences")
5 # modifying the last string in both data structures
6 list_[3] = "Mutable"
7 print( list_ )
8 try:
9     tuple_[3] = "Immutable"
10    print( tuple_ )
11 except TypeError:
12     print( "Tuples cannot be modified because they are immutable" )
```

Output:



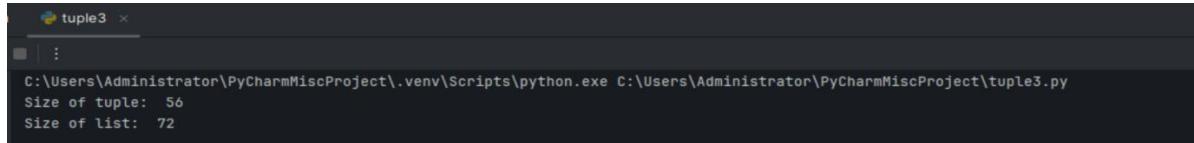
```
tuple2 x
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple2.py
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable
```

75. Python Code to show the difference in the size of a list and a tuple



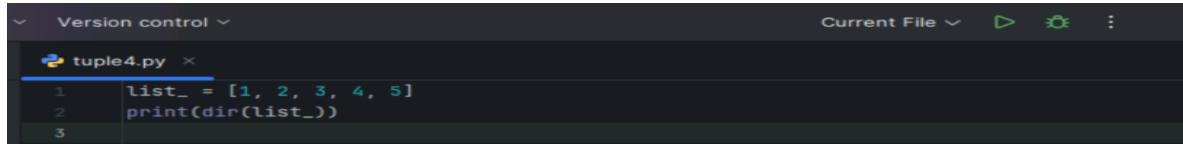
```
tuple3.py x
1 # Code to show the difference in the size of a list and a tuple
2 #creating a list and a tuple
3 list_ = ["Python", "Lists", "Tuples", "Differences"]
4 tuple_ = ("Python", "Lists", "Tuples", "Differences")
5 # printing sizes
6 print("Size of tuple: ", tuple_.__sizeof__())
7 print("Size of list: ", list_.__sizeof__())
```

Output:



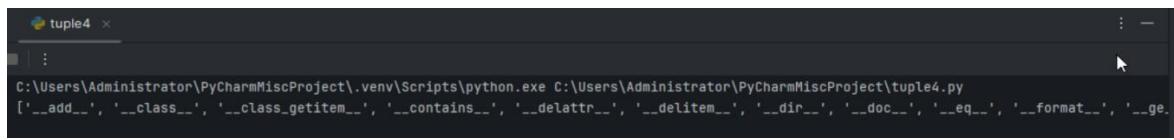
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple3.py
Size of tuple: 56
Size of list: 72
```

76. Python code to understand the basic inbuilt functions.



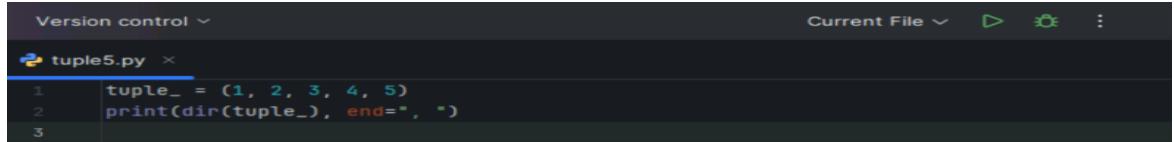
```
list_ = [1, 2, 3, 4, 5]
print(dir(list_))
```

Output:



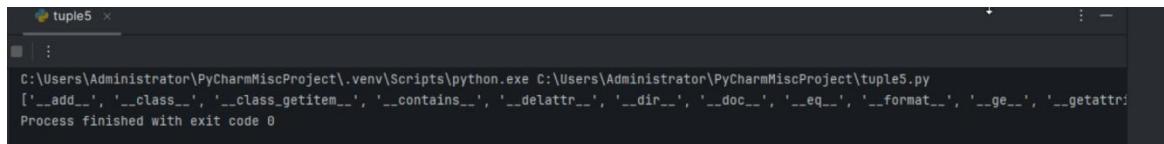
```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple4.py
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
```

77. Python program to print the directory of a tuple.



```
tuple_ = (1, 2, 3, 4, 5)
print(dir(tuple_), end="")
```

Output:



```
C:\Users\Administrator\PyCharmMiscProject\.venv\Scripts\python.exe C:\Users\Administrator\PyCharmMiscProject\tuple5.py
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__',
```