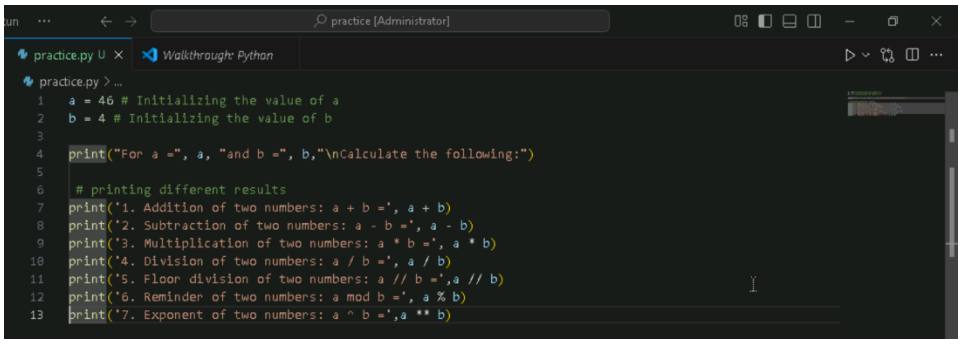
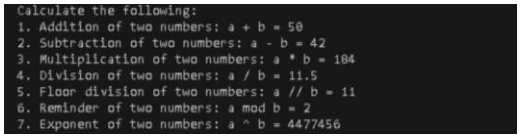
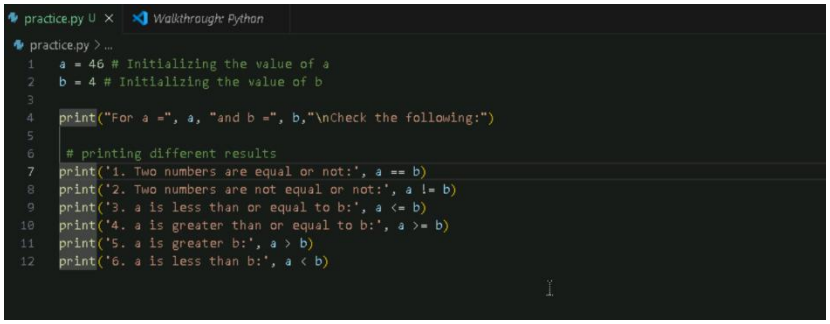
## 1) Arithmetic Operators



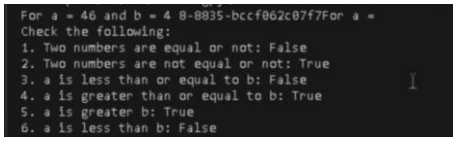

**Output:**




## 2) Comparison Operators




**Output:**




## 3) Assignment Operators

```python
a = 34 # Initialize the value of a
b = 6 # Initialize the value of b

# printing the different results
print('a += b:', a + b)
print('a -= b:', a - b)
print('a *= b:', a * b)
print('a /= b:', a / b)
print('a %= b:', a % b)
print('a **= b:', a ** b)
print('a //= b:', a // b)
```

**Output:**

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

### 4) Bitwise Operators

```python
a = 7 # initializing the value of a
b = 8 # initializing the value of b

# printing different results
print('a & b :', a & b)
print('a | b :', a | b)
print('a ^ b :', a ^ b)
print('~a :', ~a)
print('a << b :', a << b)
print('a >> b :', a >> b)
```

**Output:**

```
PS C:\Users\Administrator\Documents\practice> c:; cd 'c:\Users\Administrator\Documents\practice'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '49571' '--' 'C:\Users\Administrator\Documents\practice\practice.py'
x5cadapter/../..\x5cdebugpy\x5claucher' '49571' '--' 'C:\x5cUsers\x5cAdministrator\x5cDocuments\x5cpractice\x5cpractice.p
a & b : 0    52a6-48f8-8835-bccf862c07f7a & b :
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0
```

### 5) Logical Operators

```python
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:**

```
x5cadapter/../..\x5cdebugpy\x5clauncher' '49613' '--' 'C:\x5cUsers\x5cAdministrator\x5cDocuments\x5cpractice\x5cpractice.p
y' ;ec1bef83-52a6-48f8-8835-bccf062c07f7For 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) Membership Operators

```python
myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]

# initializing x and y with some values
x = 31
y = 28

# printing the given list
print("Given List:", myList)

# checking if x is present in the list or not
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.")

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

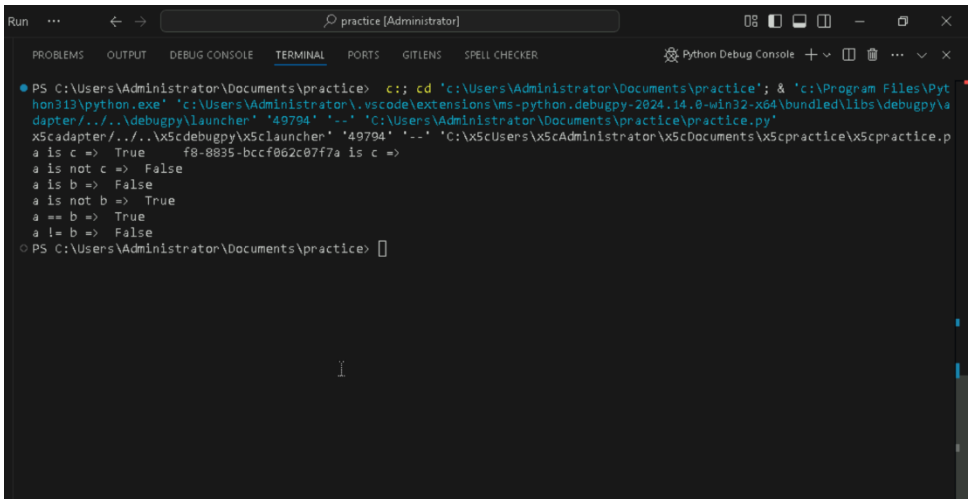
**Output:**

```
bugpy\x5clauncher' '49721' '--' 'C:\x5cUsers\x5cAdministrator\x5cDocuments\x5cpractice\x5cpractice.py' ;ec1bef83-52a6-48f8
-8835-bccf062c07f7
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.
```

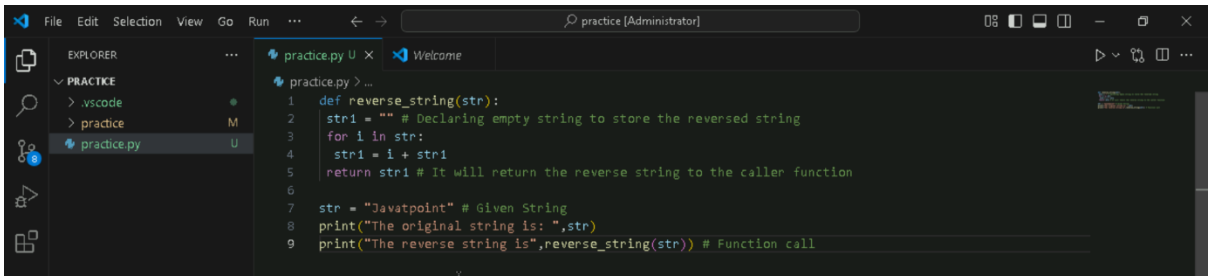
## 7) Identity Operators

```python
a = ["Rose", "Lotus"]
b = ["Rose", "Lotus"]

# initializing a variable c and storing the value of a in c
c = a

# printing the different results
print("a is c => ", a is c)
print("a is not c => ", a is not c)
print("a is b => ", a is b)
print("a is not b => ", a is not b)
print("a == b => ", a == b)
print("a != b => ", a != b)
```

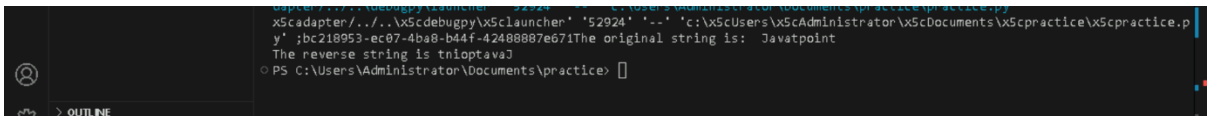
**Output:**



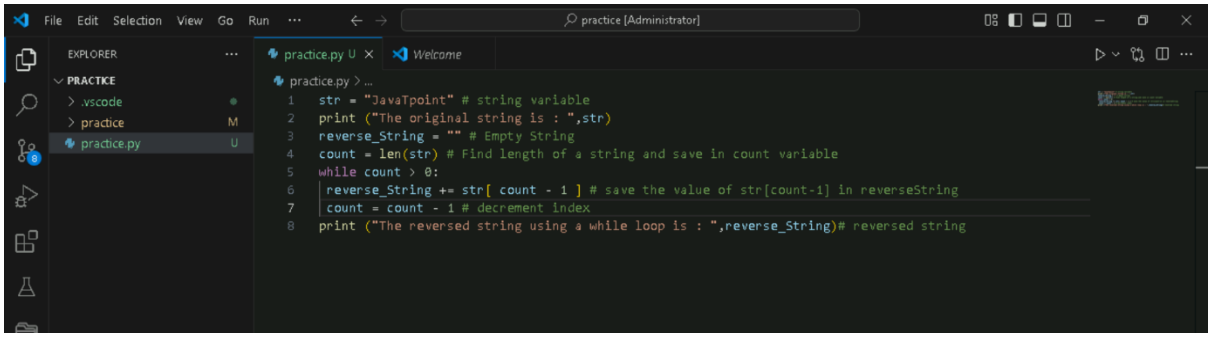

### 8) Using for loop



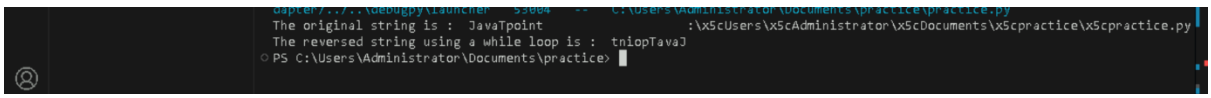

**Output:**




### 9) Using while loop




**Output:**




### 10) Using the slice ([]) operator

```
def reverse(str):
    str = str[::-1]
    return str

s = "JavaTpoint"
print ("The original string is : ",s)
print ("The reversed string using extended slice operator is : ",reverse(s))
```

**Output:**

```
The original string is :  JavaTpoint
The reversed string using extended slice operator is :  tniopTavaJ
PS C:\Users\Administrator\Documents\practice>
```

### 11) Using reverse function with join

```
def reverse(str):
    string = "".join(reversed(str)) # reversed() function inside the join() function
    return string

s = "JavaTpoint"

print ("The original string is : ",s)
print ("The reversed string using reversed() is : ",reverse(s) )
```

**Output:**

```
The original string is :  JavaTpoint
The reversed string using reversed() is :  tniopTavaJ
PS C:\Users\Administrator\Documents\practice>
```

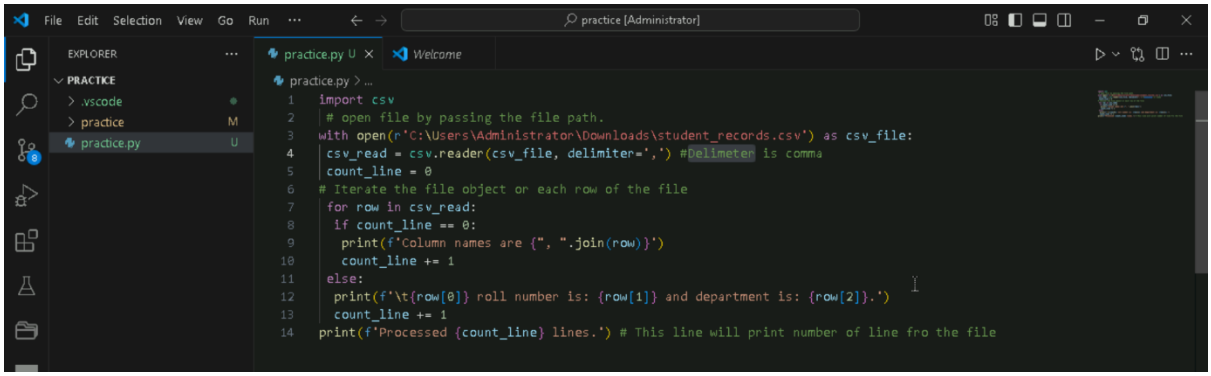
### 12) Using recursion()

```
def reverse(str):
    if len(str) == 0: # Checking the length of string
        return str
    else:
        return reverse(str[1:]) + str[0]

str = "Devansh shams"
print ("The original string is : ", str)
print ("The reversed string(using recursion) is : ", reverse(str))
```

**Output:**

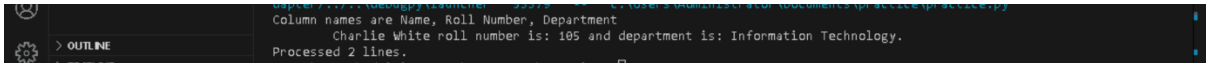
```
The original string is :  Devansh shams
The reversed string(using recursion) is :  smahs hsnaveD
PS C:\Users\Administrator\Documents\practice>
```

### 13) To read CSV file in Python



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

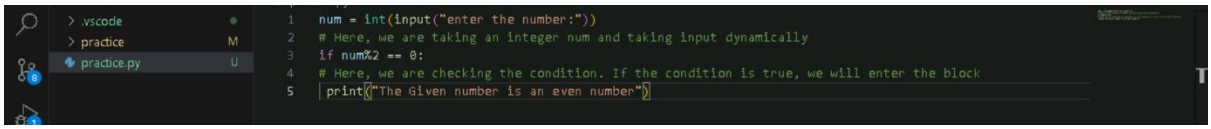
**Output:**



```
Column names are Name, Roll Number, Department
        Charlie White roll number is: 105 and department is: Information Technology.
Processed 2 lines.
```

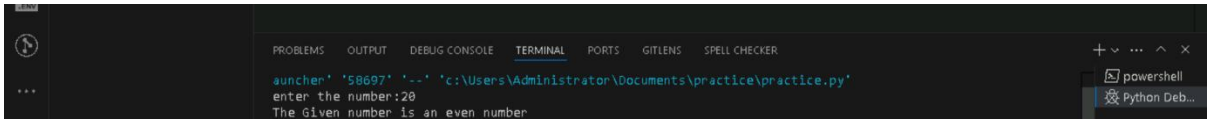
# If-else statements

### 14) If statement



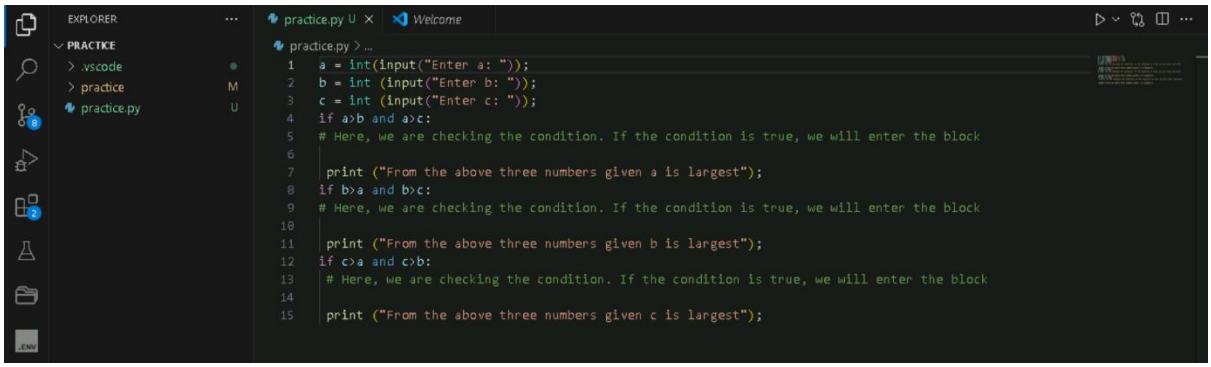
```
num = int(input("enter the number:"))
# Here, we are taking an integer num and taking input dynamically
if num%2 == 0:
# Here, we are checking the condition. If the condition is true, we will enter the block
 print("The Given number is an even number")
```

**Output:**



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  SPELL CHECKER
auncher' '58697' '--' 'c:\Users\Administrator\Documents\practice\practice.py'
enter the number:20
The Given number is an even number
```

### 15) Program to print the largest of the three numbers.

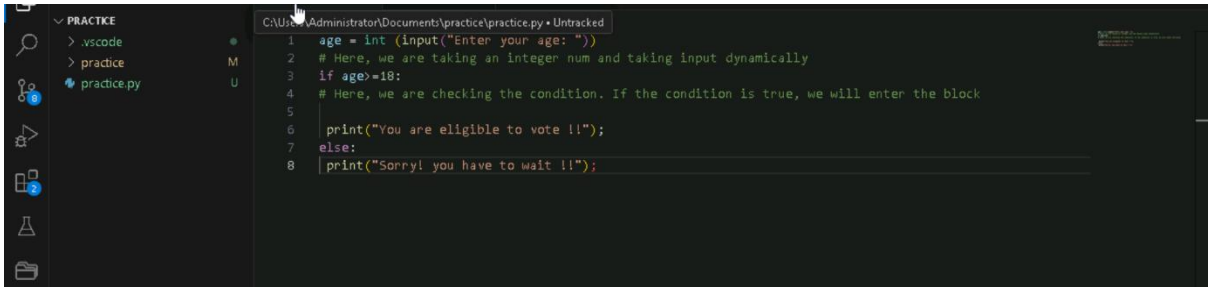


```
a = int(input("Enter a: "));
b = int (input("Enter b: "));
c = int (input("Enter c: "));
if a>b and a>c:
# Here, we are checking the condition. If the condition is true, we will enter the block

 print ("From the above three numbers given a is largest");
if b>a and b>c:
# Here, we are checking the condition. If the condition is true, we will enter the block

 print ("From the above three numbers given b is largest");
if c>a and c>b:
 # Here, we are checking the condition. If the condition is true, we will enter the block

 print ("From the above three numbers given c is largest");
```

**Output:**

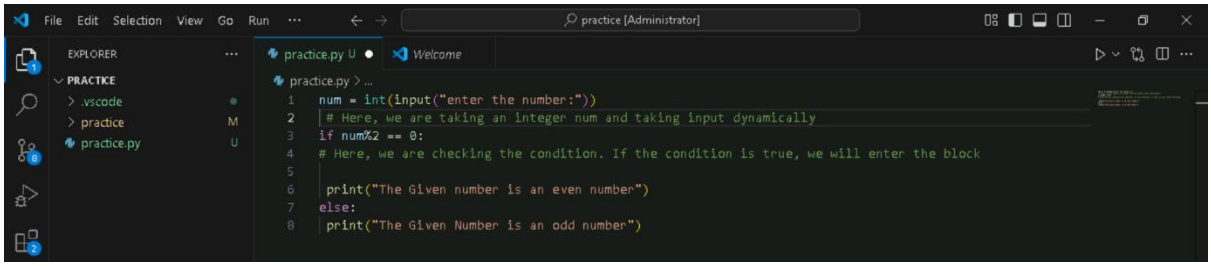
```
Enter a: 20
Enter b: 27
Enter c: 23
From the above three numbers given b is largest
```

### 16) Program to check whether a person is eligible to vote or not.



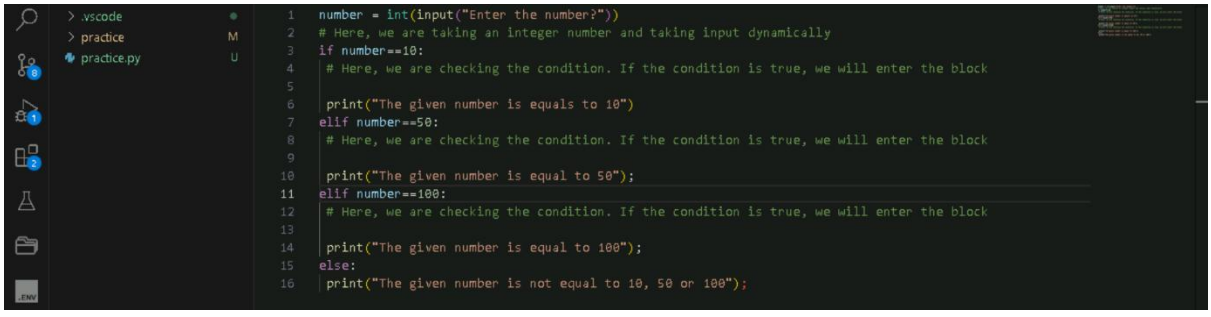

**Output:**


**17) Program to check whether a number is even or not.**




**Output:**


**18) Elif statement**




**Output:**


**19) Elif statement**

```python
marks = int(input("Enter the marks? "))
if marks > 85 and marks <= 100:
# Here, we are checking the condition. If the condition is true, we will enter the block

 print("Congrats ! you scored grade A ...")
elif marks > 60 and marks <= 85:
 # Here, we are checking the condition. If the condition is true, we will enter the block

 print("You scored grade B + ...")
elif marks > 40 and marks <= 60:
# Here, we are checking the condition. If the condition is true, we will enter the block

 print("You scored grade B ...")
elif (marks > 30 and marks <= 40):
# Here, we are checking the condition. If the condition is true, we will enter the block

 print("You scored grade C ...")
else:
```

**Output:**

```
Enter the marks? 85
You scored grade B + ...
PS C:\Users\Administrator\Documents\practice>  c:; cd 'c:\Users\Administrator\Documents\practice'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '59071' '--' 'c:\Users\Administrator\Documents\practice\practice.py'
x5cadapter/../..\x5cdebugpy\x5clauncher' '59071' '--' 'c:\x5cUsers\x5cAdministrator\x5cDocuments\x5cpractice\x5cpractice.py' ;d11dc8b3-3ca1-44d6-a22a-2a26ea78d530Enter the marks? 86
Congrats ! you scored grade A ...
```

# Python Loops

### 20) For Loop

```python
numbers = [4, 2, 6, 7, 3, 5, 8, 10, 6, 1, 9, 2]

# variable to store the square of the number
square = 0

# Creating an empty list
squares = []

# Creating a for loop
for value in numbers:
 square = value ** 2
 squares.append(square)
print("The list of squares is", squares)
```

**Output:**

```
The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4]
```

### 21) Using else Statement with for Loop

```python
string = "Python Loop"

# Initiating a loop
for s in string:
# giving a condition in if block
 if s == "o":
  print("If block")
# if condition is not satisfied then else block will be executed
 else:
  print(s)
```

**Output:**

```
P
y
t
h
If block
n

L
If block
If block
p
```

### 22) Using else Statement with for Loop

```python
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:**

```
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) Range() Function

```python
print(range(15))

print(list(range(15)))

print(list(range(4, 9)))

print(list(range(5, 25, 4)))
```

**Output:**

```
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) Range() Function

```python
tuple_ = ("Python", "Loops", "Sequence", "Condition", "Range")

# iterating over tuple_ using range() function
for iterator in range(len(tuple_)):
 print(tuple_[iterator].upper())
```

**Output:**

```
PYTHON
LOOPS
SEQUENCE
CONDITION
RANGE
```

### 25) While Loop

```python
counter = 0
# Initiating the loop
while counter < 10: # giving the condition
 counter = counter + 3
 print("Python Loops")
```

**Output:**

```
Python Loops
Python Loops
Python Loops
Python Loops
```

## 26) Using else Statement with while Loops

```
counter = 0

# Iterating through the while loop
while (counter < 10):
 counter = counter + 3
 print("Python Loops") # Executed untile condition is met     "untile": Unknown word.
# Once the condition of while loop gives False this statement will be executed
else:
 print("Code block inside the else statement")
```

**Output:**

```
e\x5cpractice.py' ;d11dc8b3-3ca1-44d6-a22a-2a26ea78d530Python Loops
Python Loops
Python Loops
Python Loops
Code block inside the else statement
```

## 27) Single statement while Block

```
for string in "Python Loops":
 if string == "o" or string == "p" or string == "t":
  continue
 print('Current Letter:', string)
```

**Output:**

```
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
Python Loops
```

## 28) Continue Statement

```
for string in "Python Loops":
 if string == "o" or string == "p" or string == "t":
  continue
 print('Current Letter:', string)
```

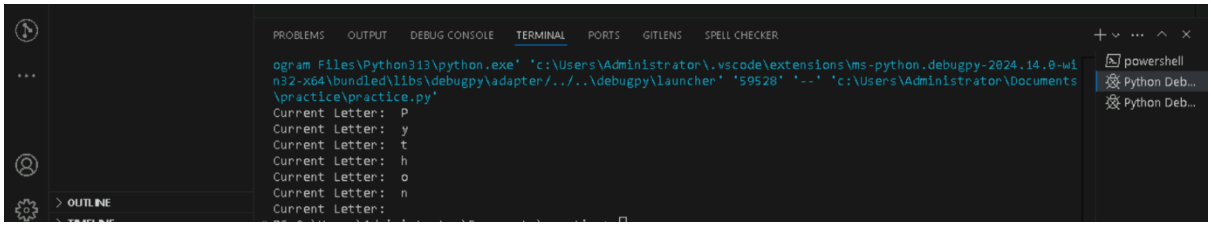
**Output:**

```
ogram Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '59498' '--' 'c:\Users\Administrator\Documents\practice\practice.py'
Current Letter: P
Current Letter: y
Current Letter: h
Current Letter: n
Current Letter:
Current Letter: L
Current Letter: s
```

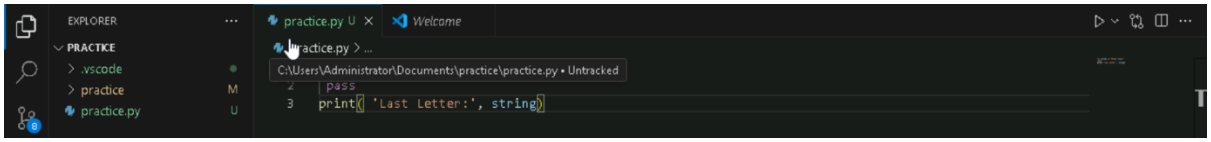
## 29) Break Statement

```
for string in "Python Loops":
 if string == 'L':
  break
 print('Current Letter: ', string)
```

**Output:**

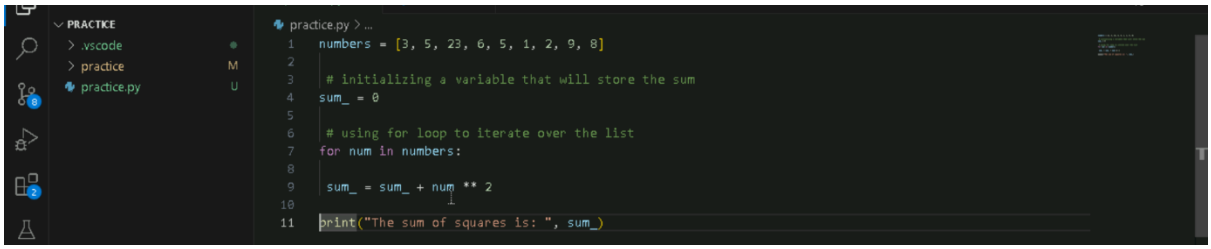



### 30) Pass Statement




**Output:**


### 31) for Loop

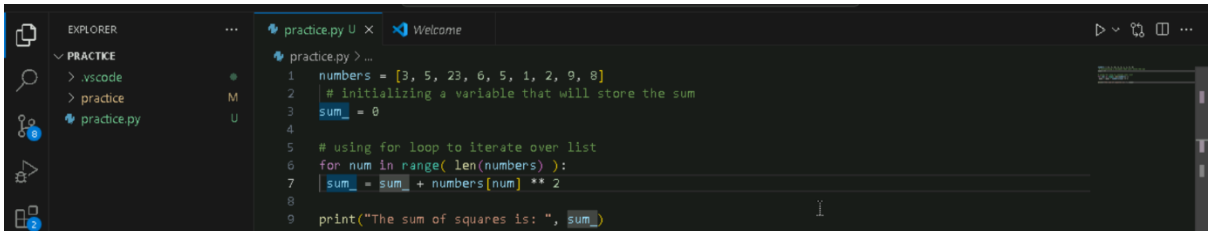



**Output:**

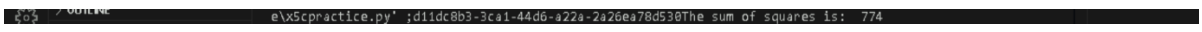

### 32) range() Function


**Output:**

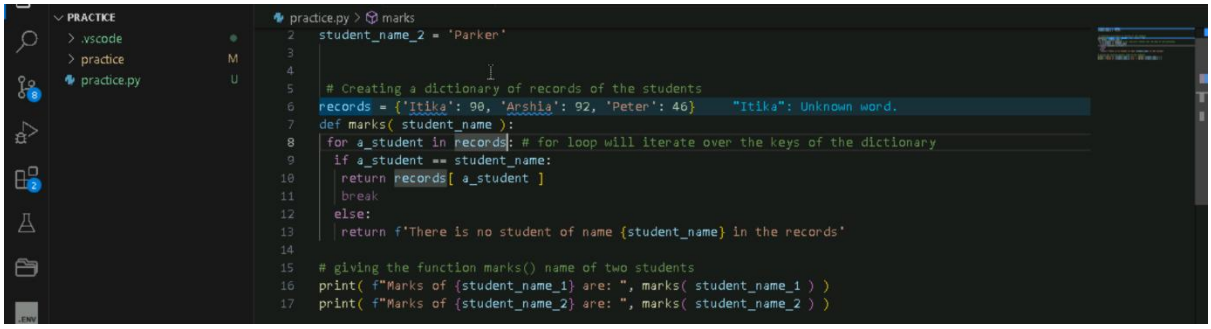

### 33) Iterating by Using Index of Sequence




**Output:**

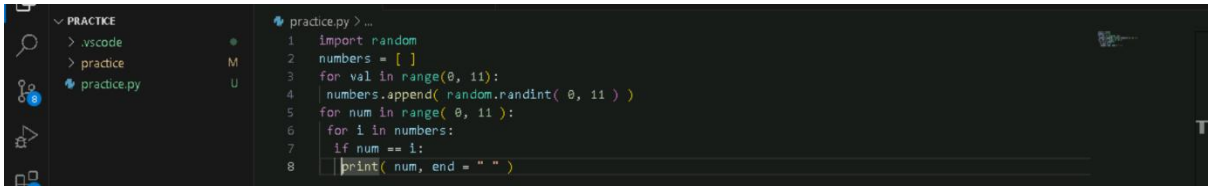



### 34) Using else Statement with for Loop

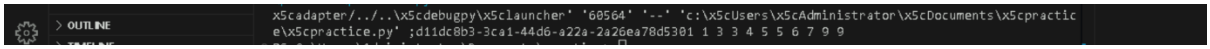



**Output:**

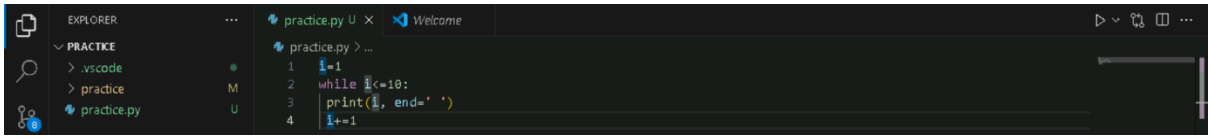

**35) Nested Loops**




**Output:**

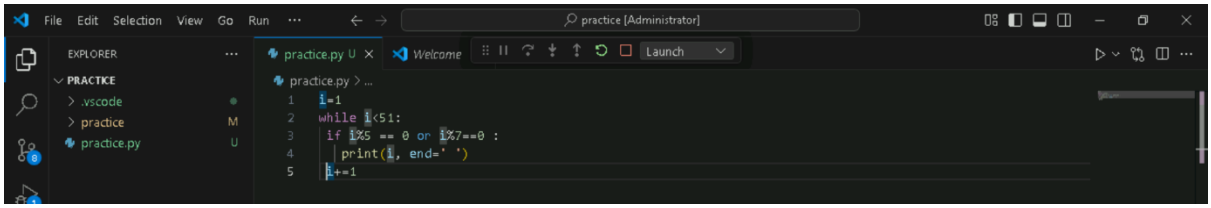



**36) Printing numbers from 1 to 10**



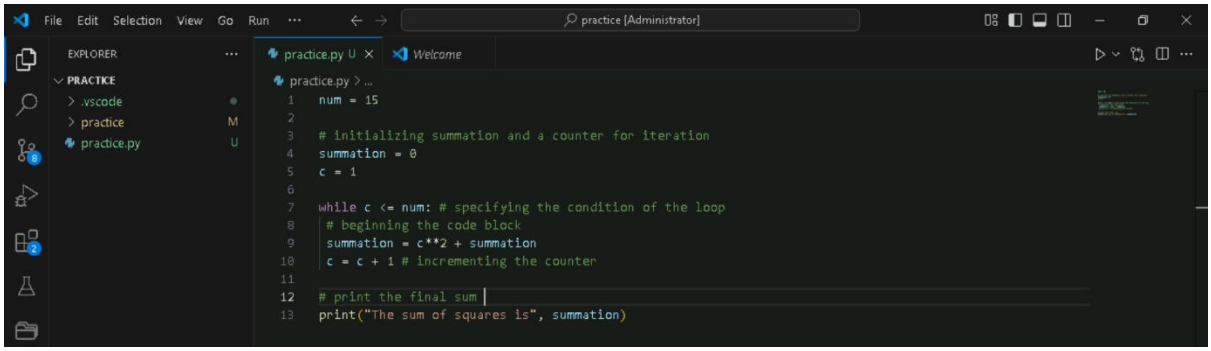

**Output:**


**37) Printing those numbers divisible by either 5 or 7 within 1 to 50 using a while loop.**



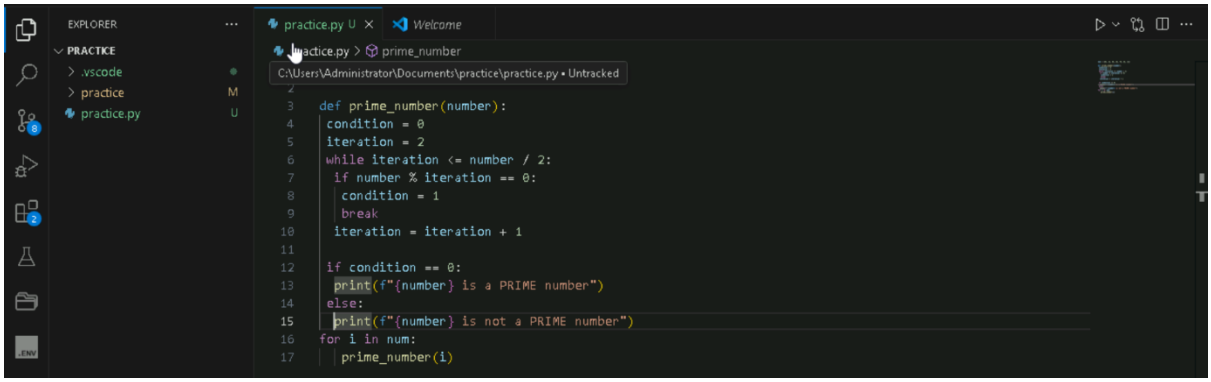

**Output:**


**38) The sum of squares of the first 15 natural numbers using a while loop.**



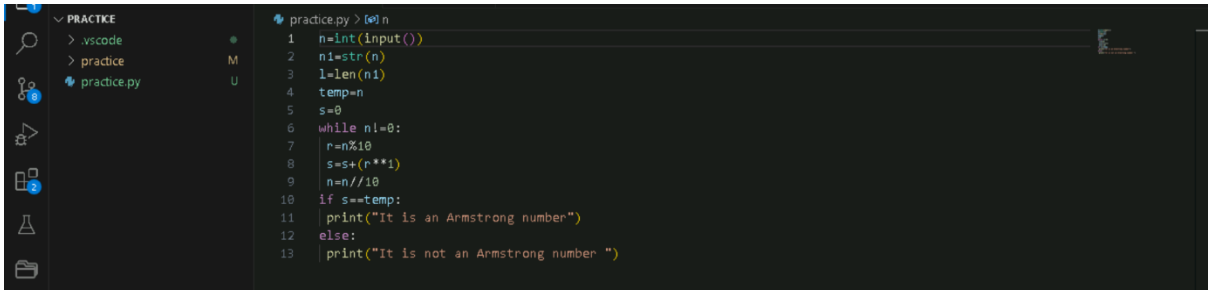

**Output:**


### 39) While loops in Python for a number is Prime number or not.




**Output:**


### 40) Armstrong and Python While Loop




**Output:**


### 41) Multiplication Table using While Loop

```python
num = 21
counter = 1
# we will use a while loop for iterating 10 times for the multiplication table
print("The Multiplication Table of: ", num)
while counter <= 10: # specifying the condition
 ans = num * counter
 print (num, 'x', counter, '=', ans)
 counter += 1 # expression to increment the counter
```

**Output:**

```
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
```

## 42) Python While Loop with List

```python
squares = []
# programing a while loop     "programing": Forbidden word.
while list_: # until list is not empty this expression will give boolean True after that False
 squares.append( (list_.pop())**2)
# Print the squares of all numbers.
print( squares )
```

**Output:**

```
[36, 16, 1, 25, 9]
```

## 43) Determine odd and even number

```python
index = 0
while index < len(list_):
 element = list_[index]
 if element % 2 == 0:
  print('It is an even number') # Print if the number is even.
 else:
  print('It is an odd number') # Print if the number is odd.
 index += 1
```

**Output:**

```
e\x5cpractice.py' ;d11dc8b3-3ca1-44d6-a22a-2a26ea78d530It is an odd number
It is an even number
It is an even number
It is an even number
It is an even number
It is an odd number
It is an odd number
It is an even number
```

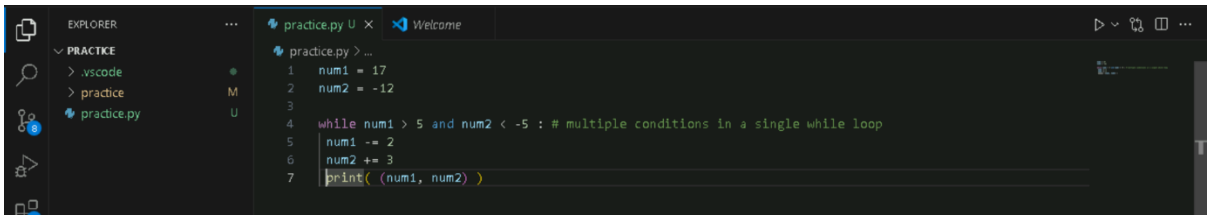
## 44) Determine the number letter

```python
List_= ['Priya', 'Neha', 'Cow', 'To']     "Priya": Unknown word.
index = 0
while index < len(List_):
 element = List_[index]
 print(len(element))
 index += 1
```

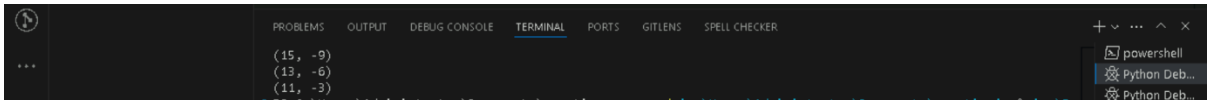
**Output:**

```
5
4
3
2
```

### 45) While Loop Multiple Conditions

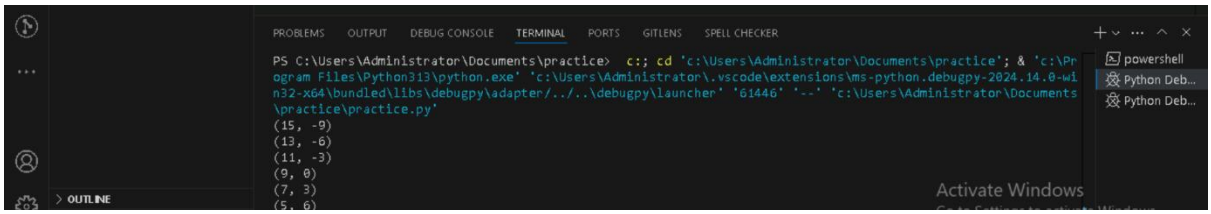



**Output:**

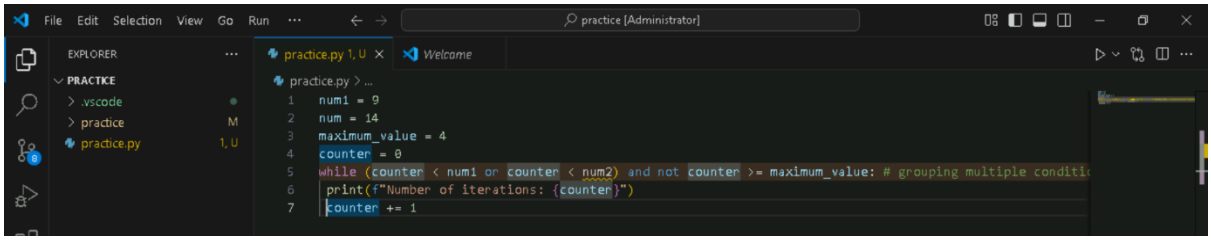



### 46) While Loop Multiple Conditions


**Output:**

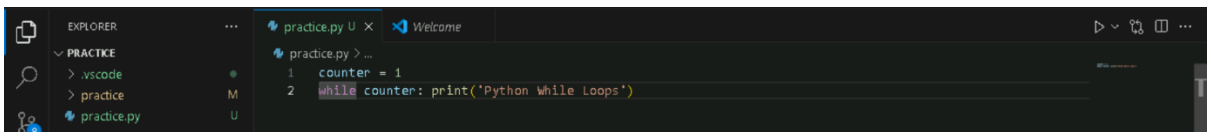



### 47) While Loop Multiple Conditions

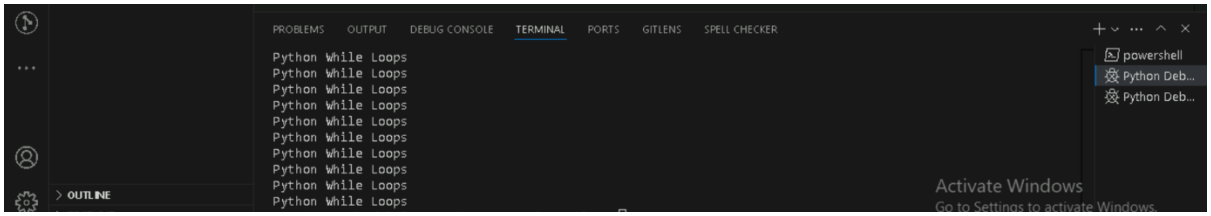



**Output:**

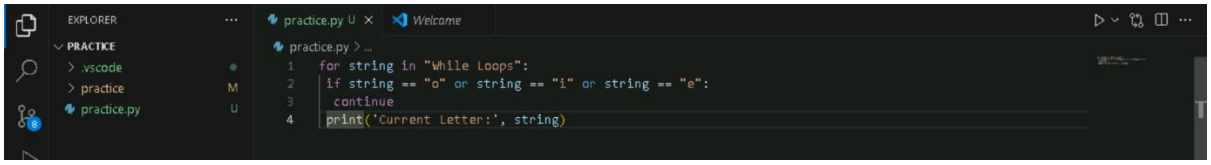

### 48) Single Statement While Loop



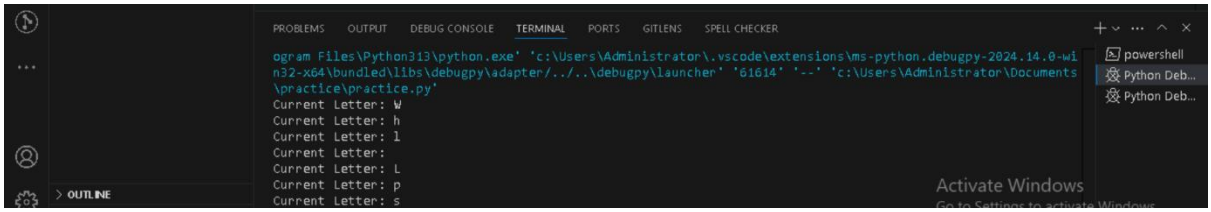

**Output:**



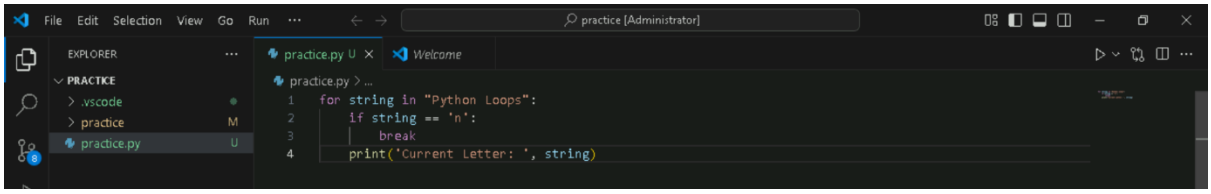

### 49) Loop Control Statements



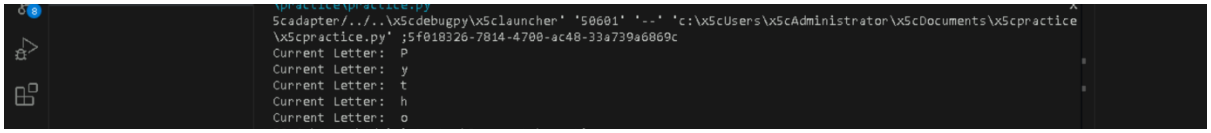

**Output:**




### 50) Break Statement

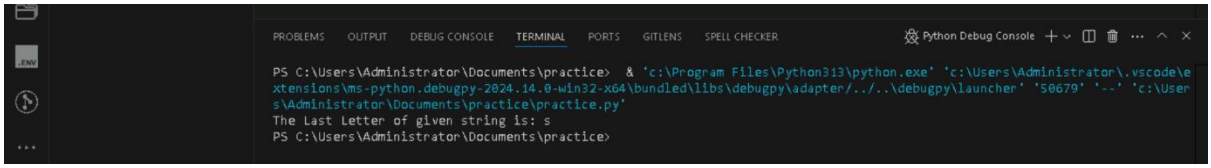



**Output:**



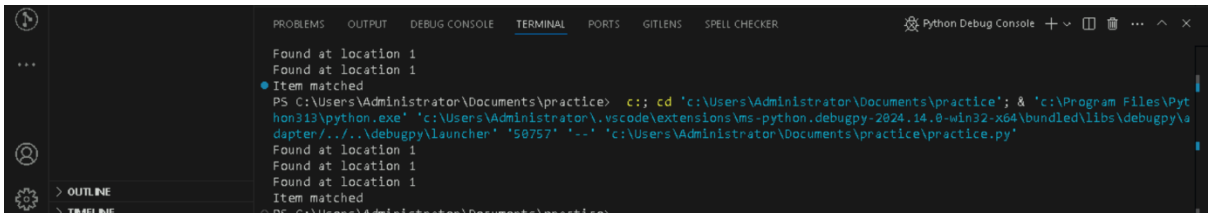

### 51) Pass Statement


**Output:**

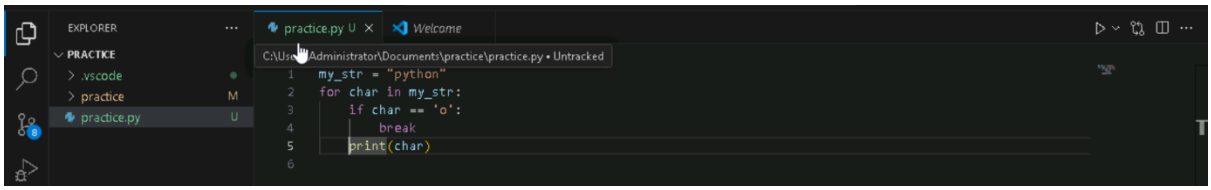



### 52) break statement with for loop

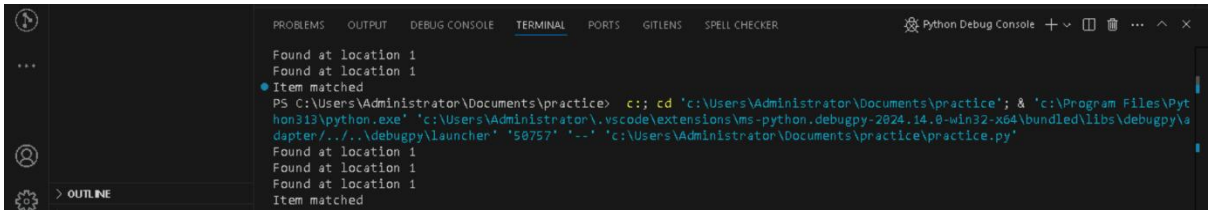

**Output:**




## 53) Breaking out of a loop early

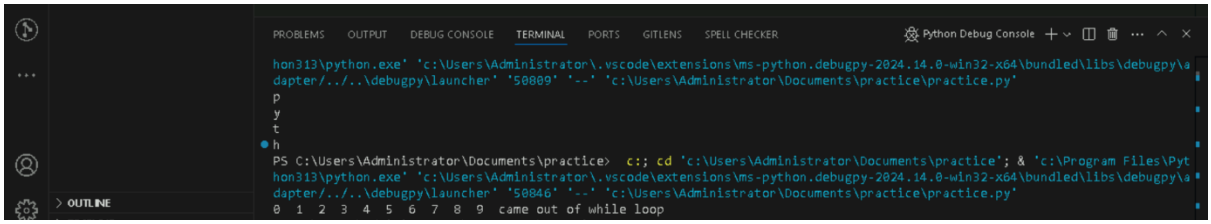



**Output:**




## 54) break statement with while loop


**Output:**




## 55) break statement with nested loops

```python
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:**

```
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:
```

### 56) Continue Statements in for Loop

```python
for iterator in range(10, 21):

# If iterator is equals to 15, loop will continue to the next iteration
    if iterator == 15:
        continue
# otherwise printing the value of iterator
    print( iterator )
```

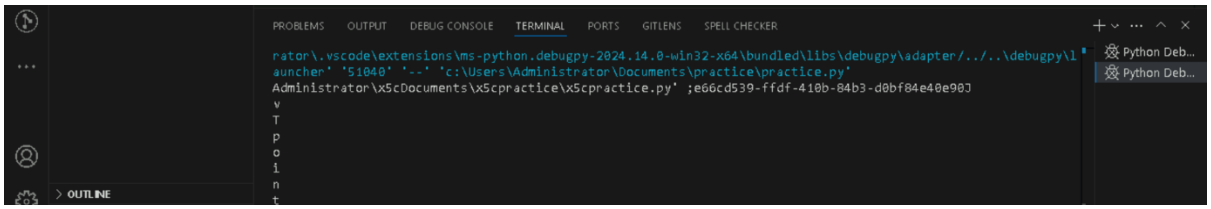
**Output:**

```
10
11
12
13
14
16
17
18
19
20
```

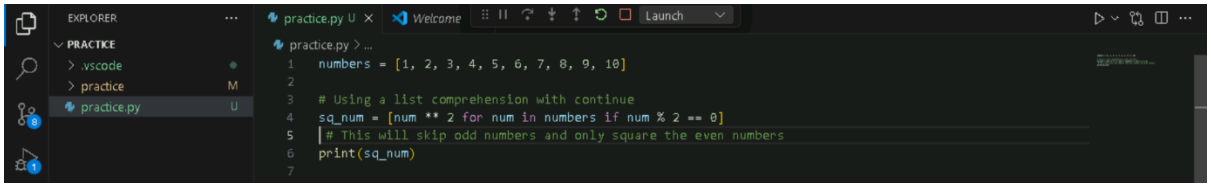
### 57) Continue Statements in while Loop

```python

# starting a while loop
while iterator < len(string):
# if loop is at letter a it will skip the remaining code and go to next iteration
    if string[iterator] == 'a':
        iterator += 1
        continue
# otherwise it will print the letter
    print(string[ iterator ])
    iterator += 1

```

**Output:**



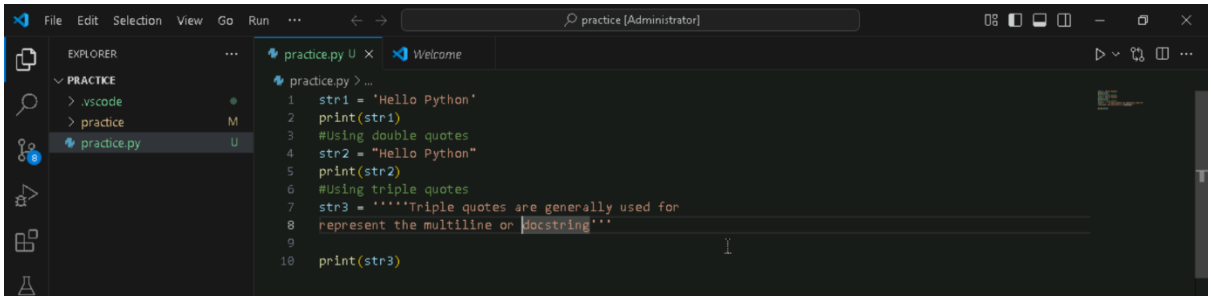

### 58) Continue statement in list comprehension



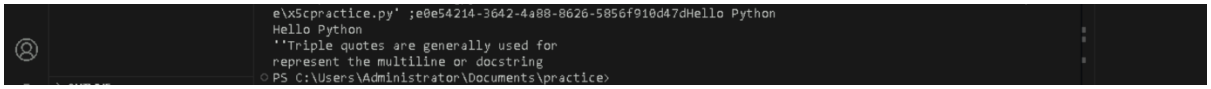

**Output:**


# Python String

### 59) Creating String in Python

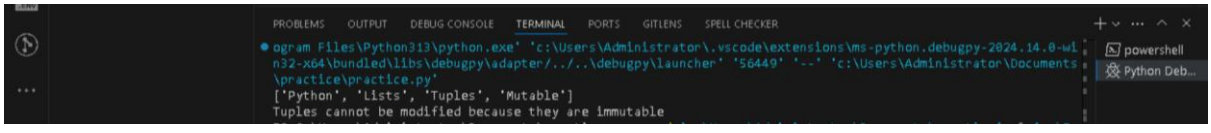



**Output:**

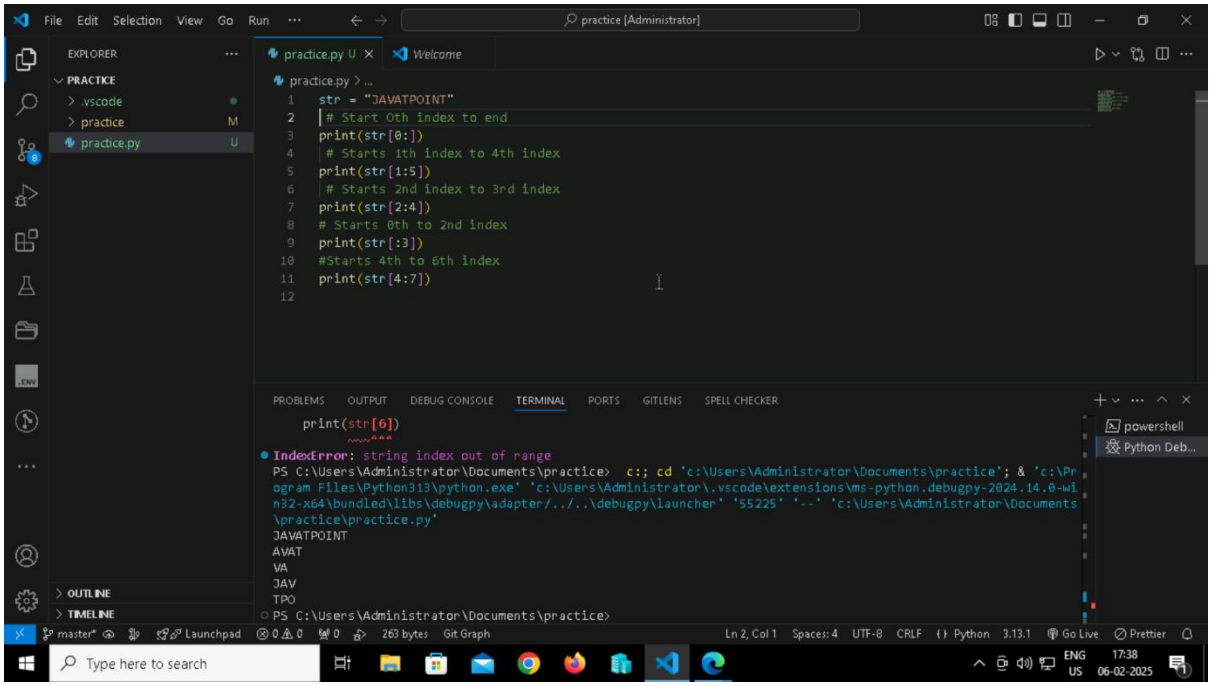



### 60) Strings indexing and splitting

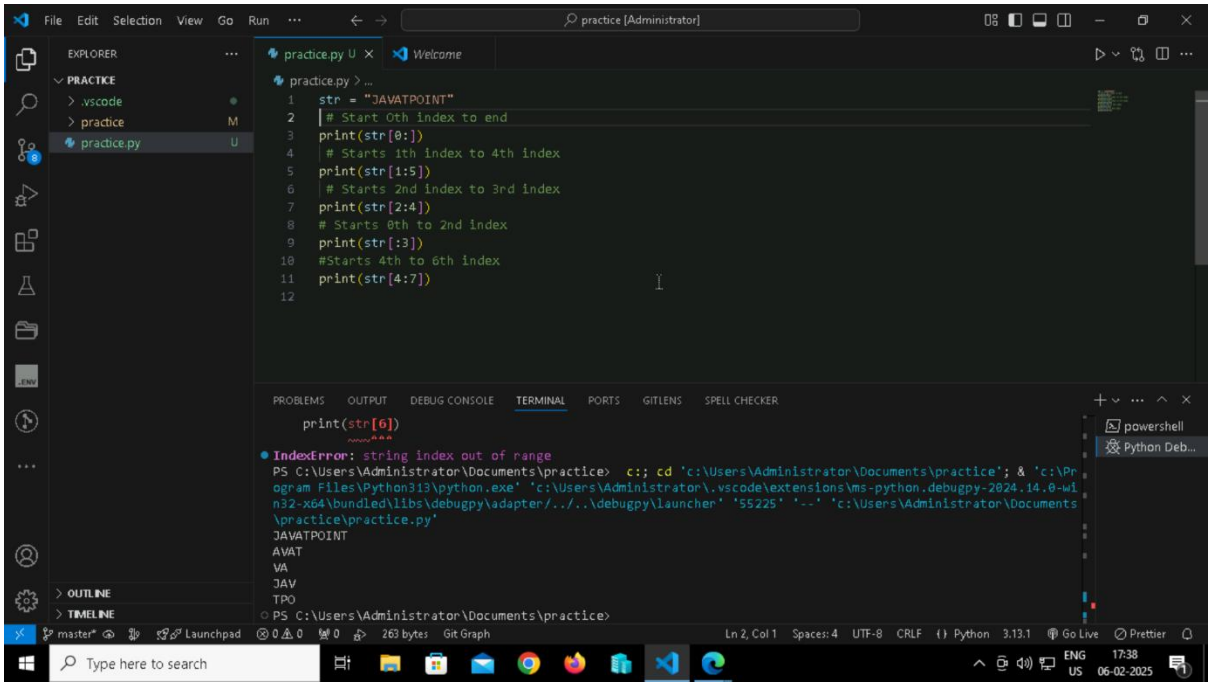

**Output:**



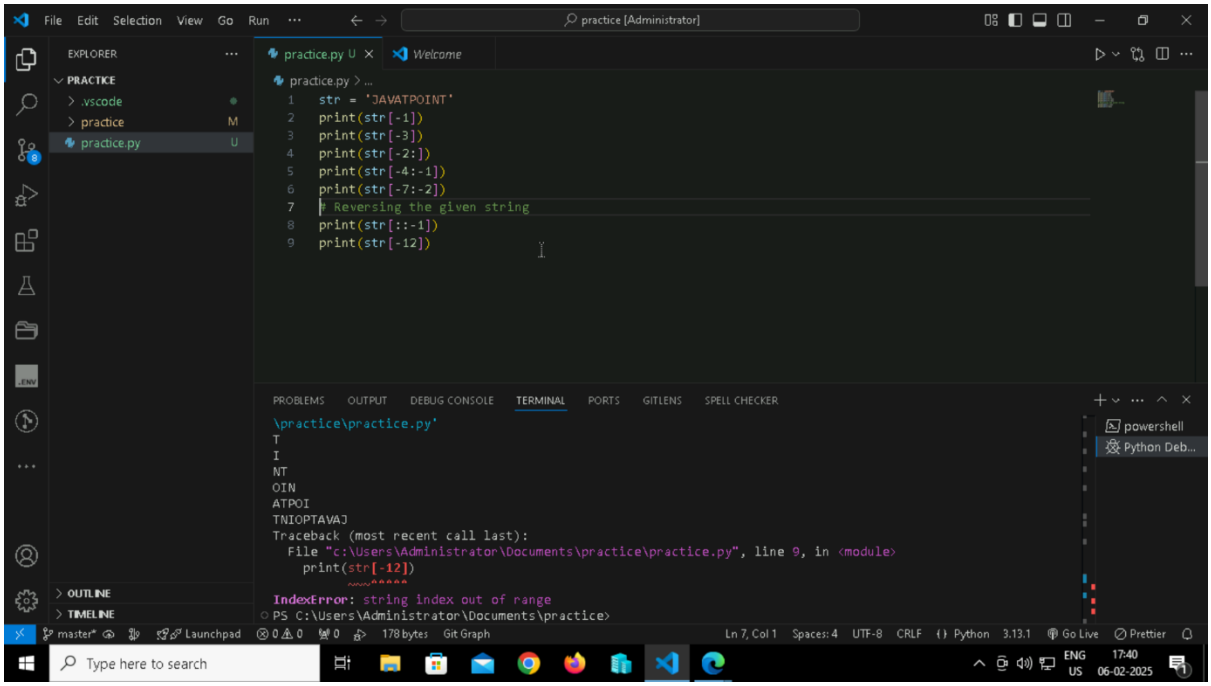

### 61) Strings indexing and splitting



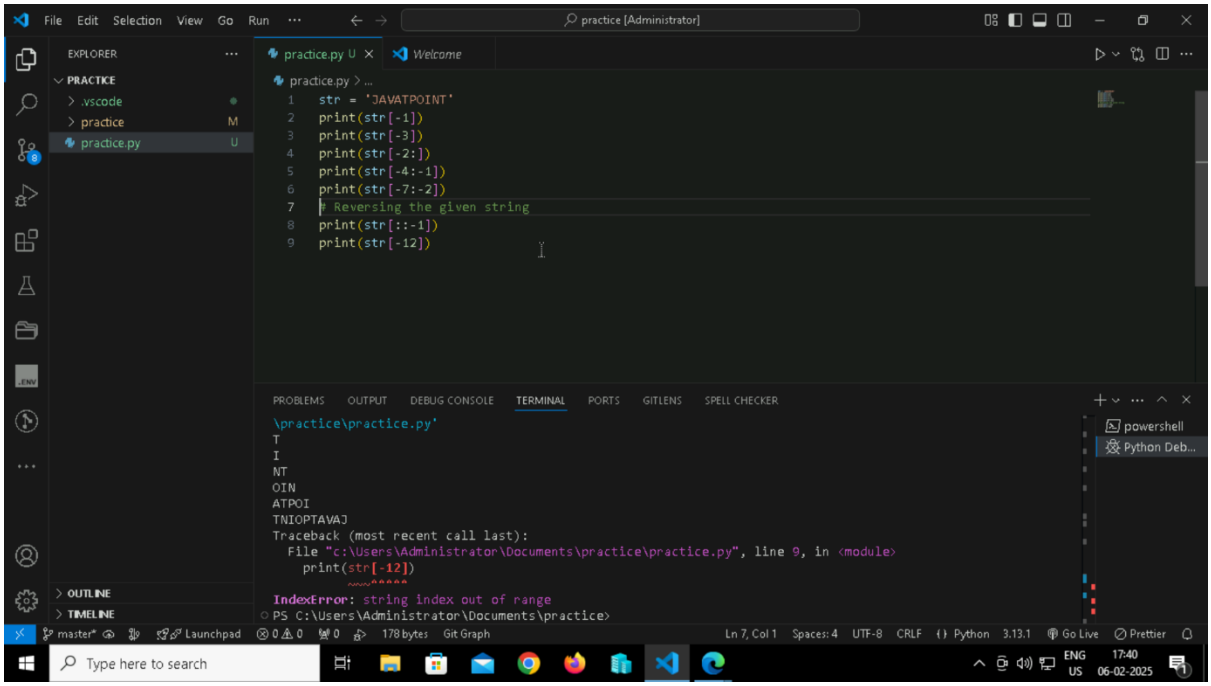

**Output:**



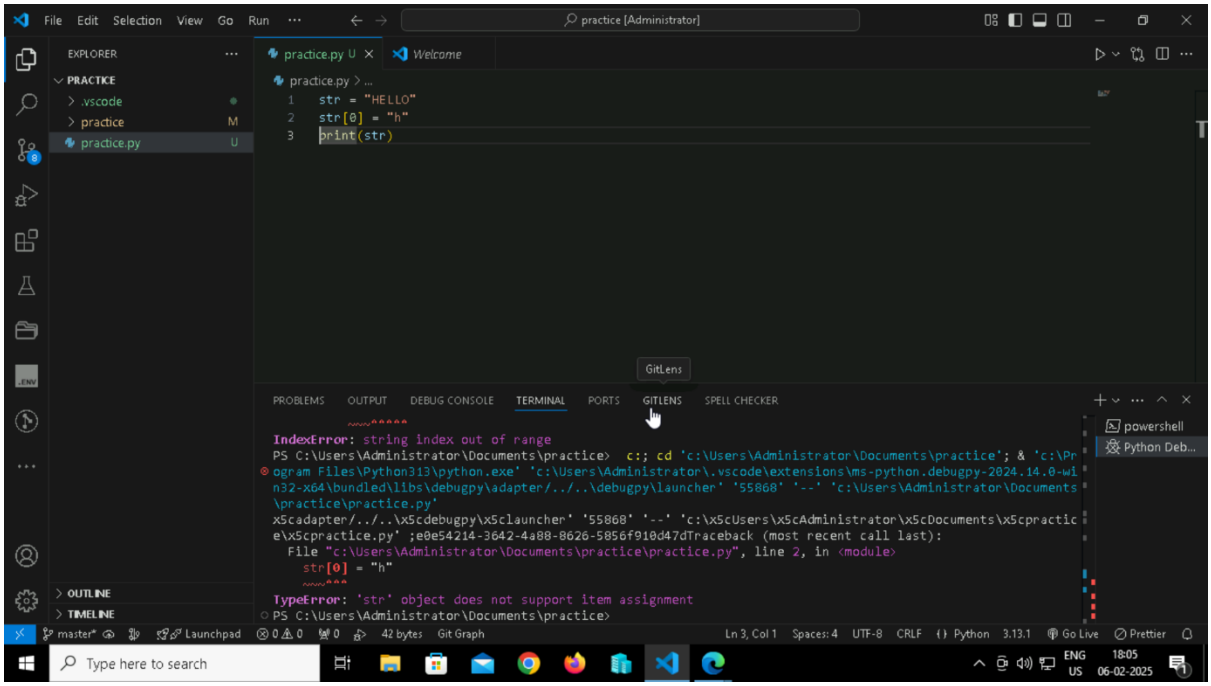

## 62) Strings indexing and splitting



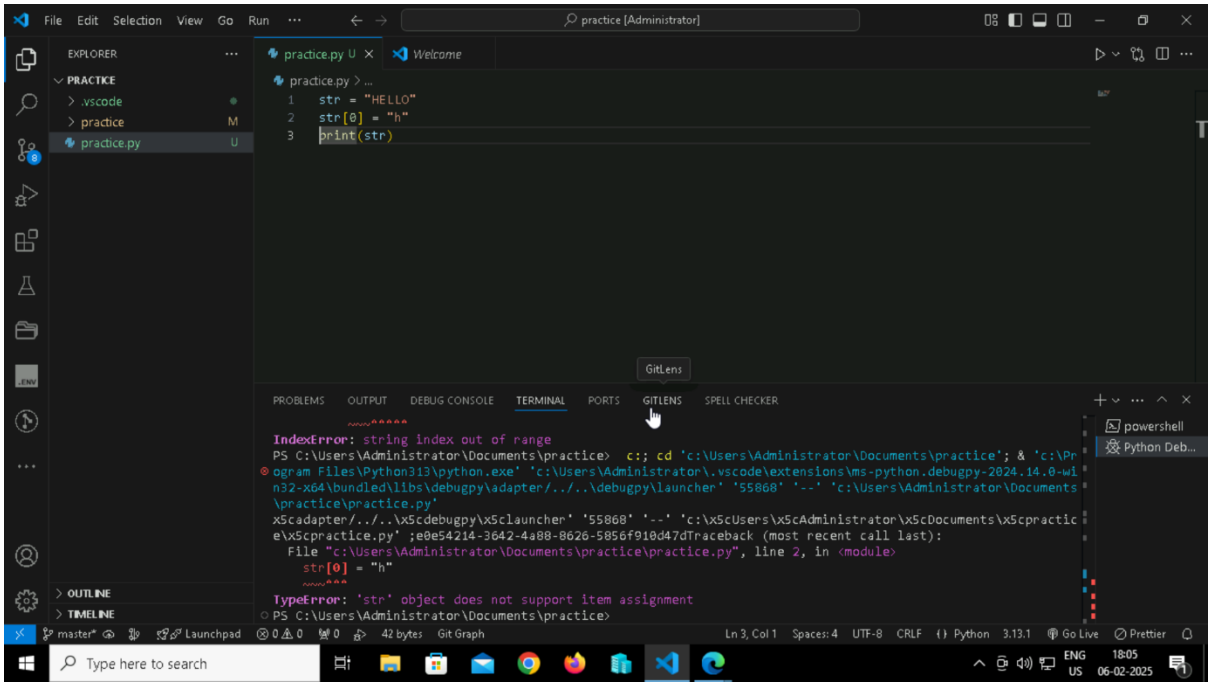

**Output:**



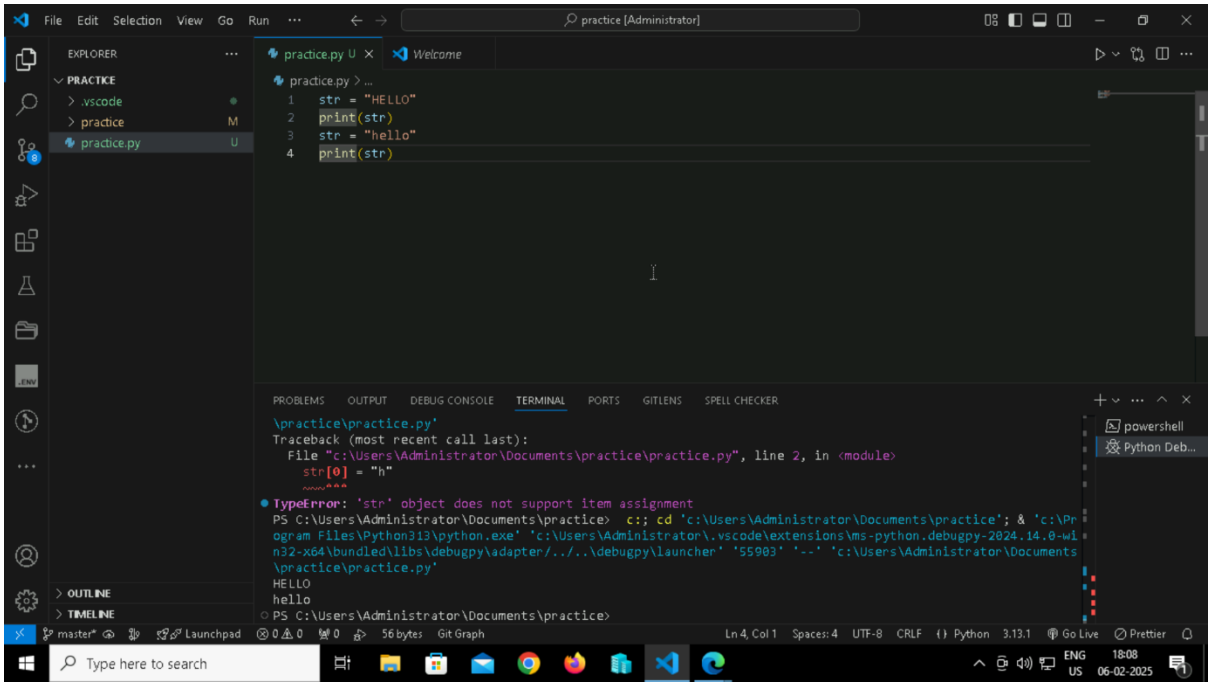

## 63) Reassigning Strings



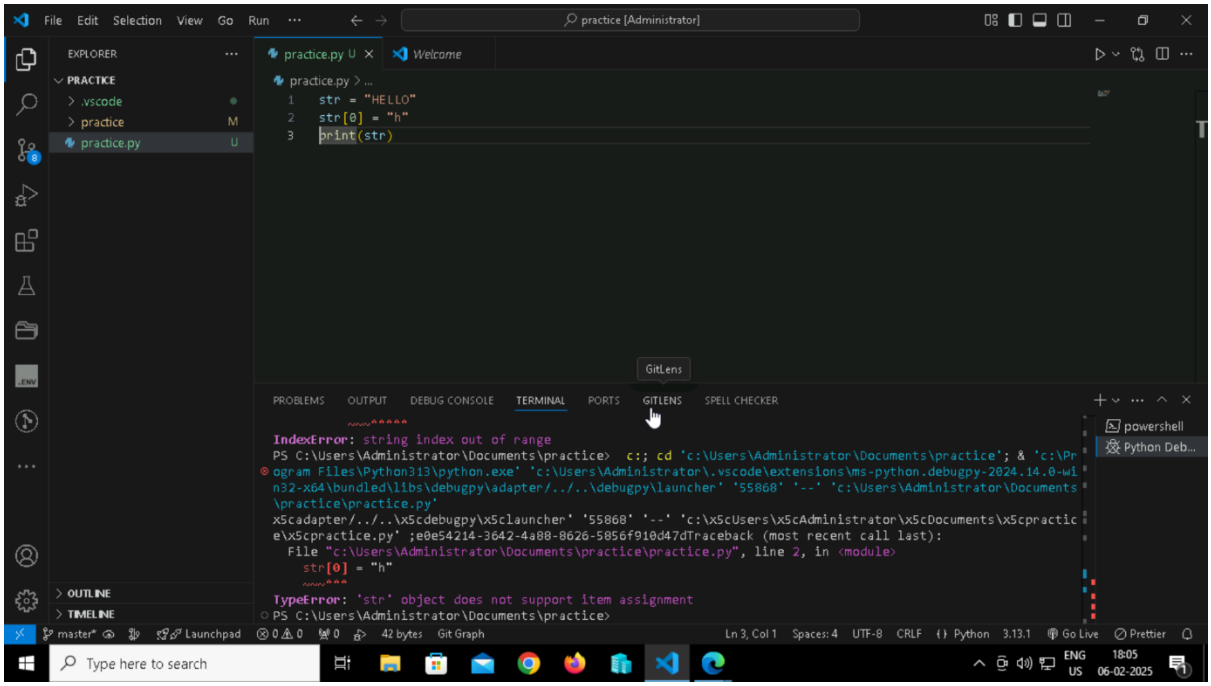

**Output:**



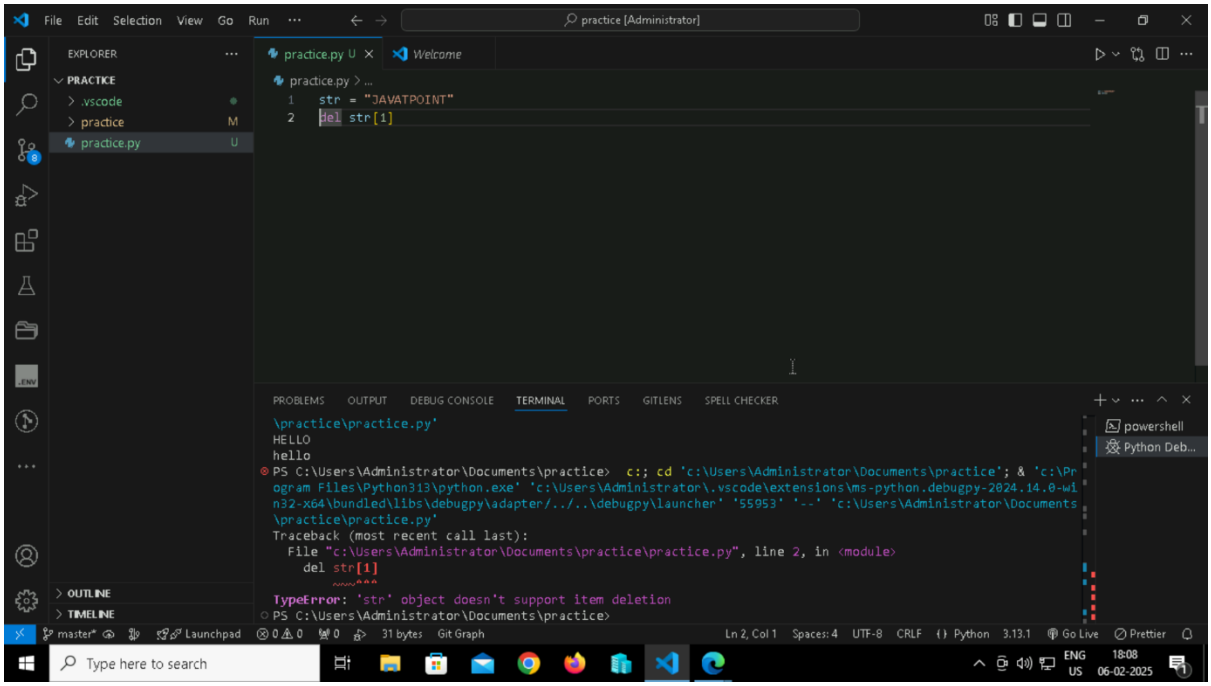

## 64) Reassigning Strings



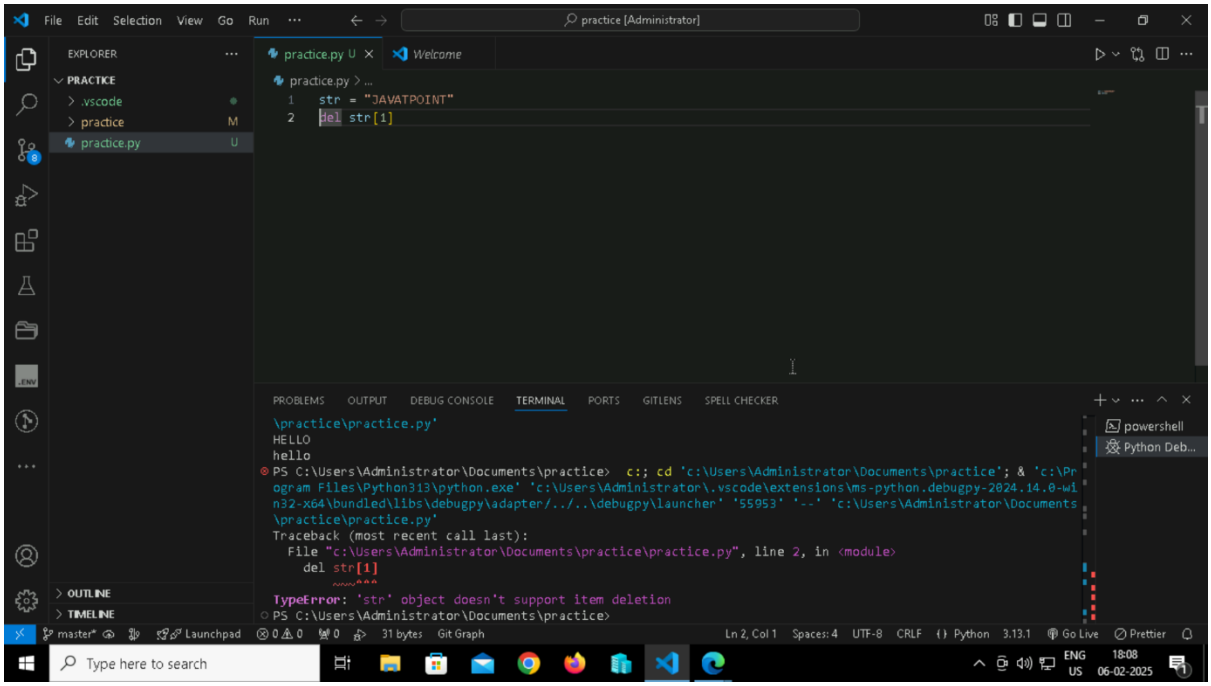

**Output:**



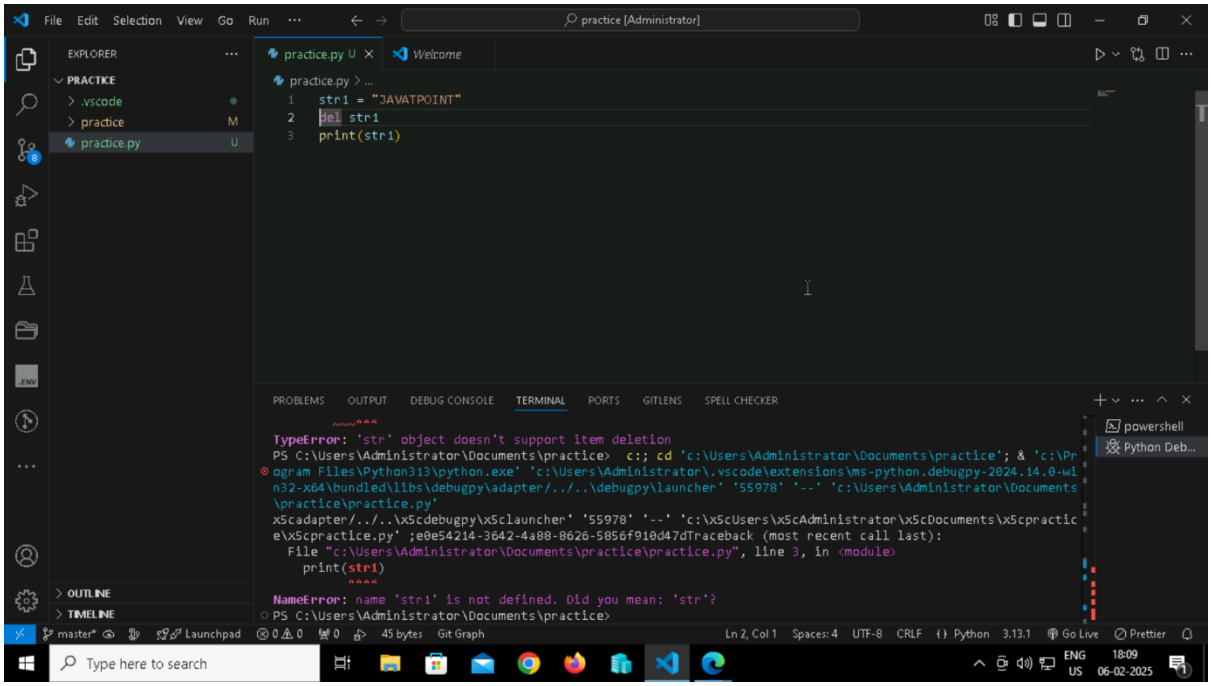

## 65) Deleting the String



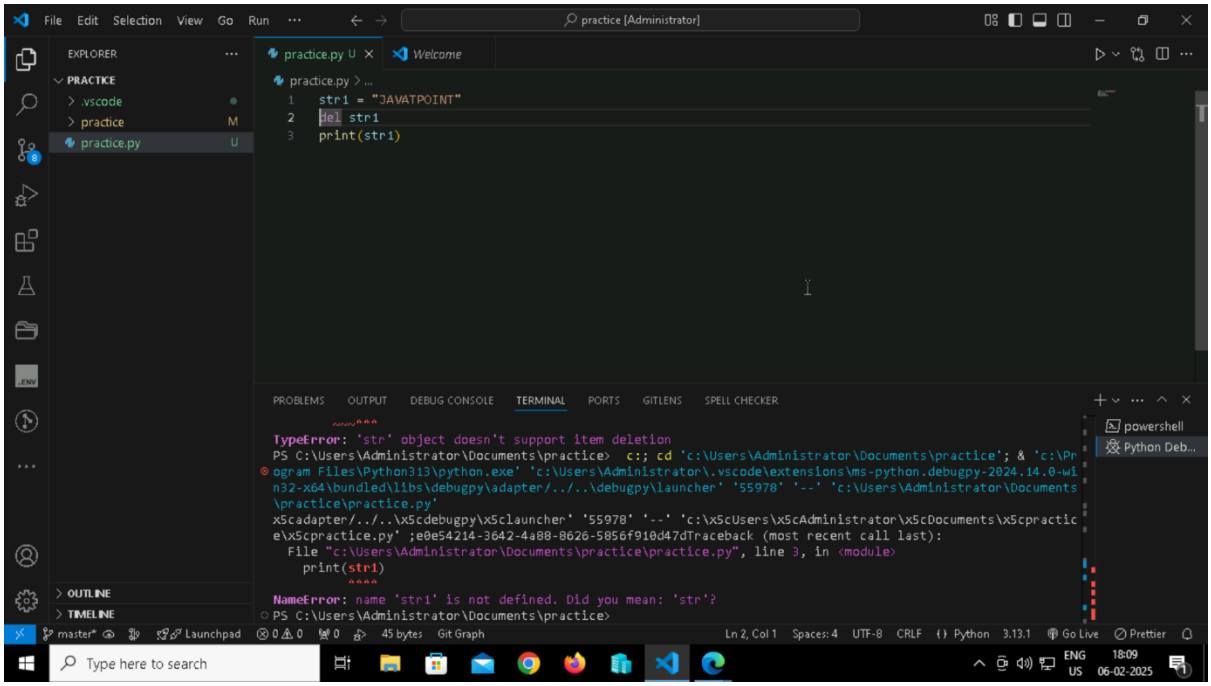

**Output:**



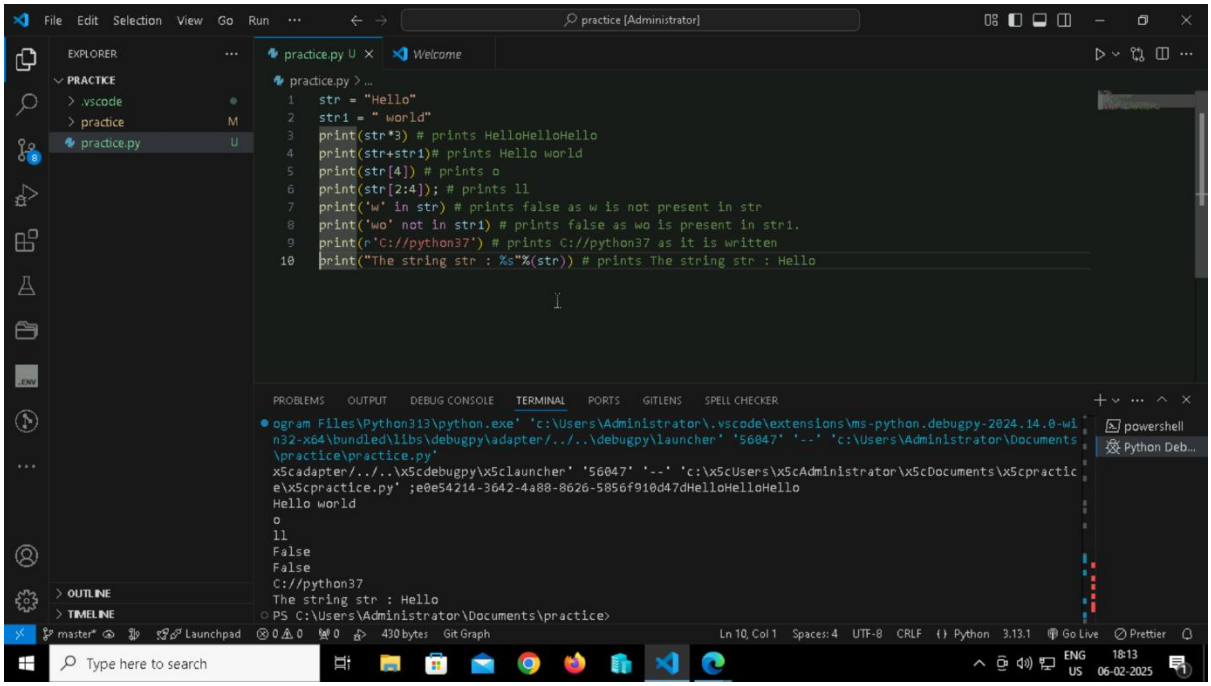

## 66) Python operators.



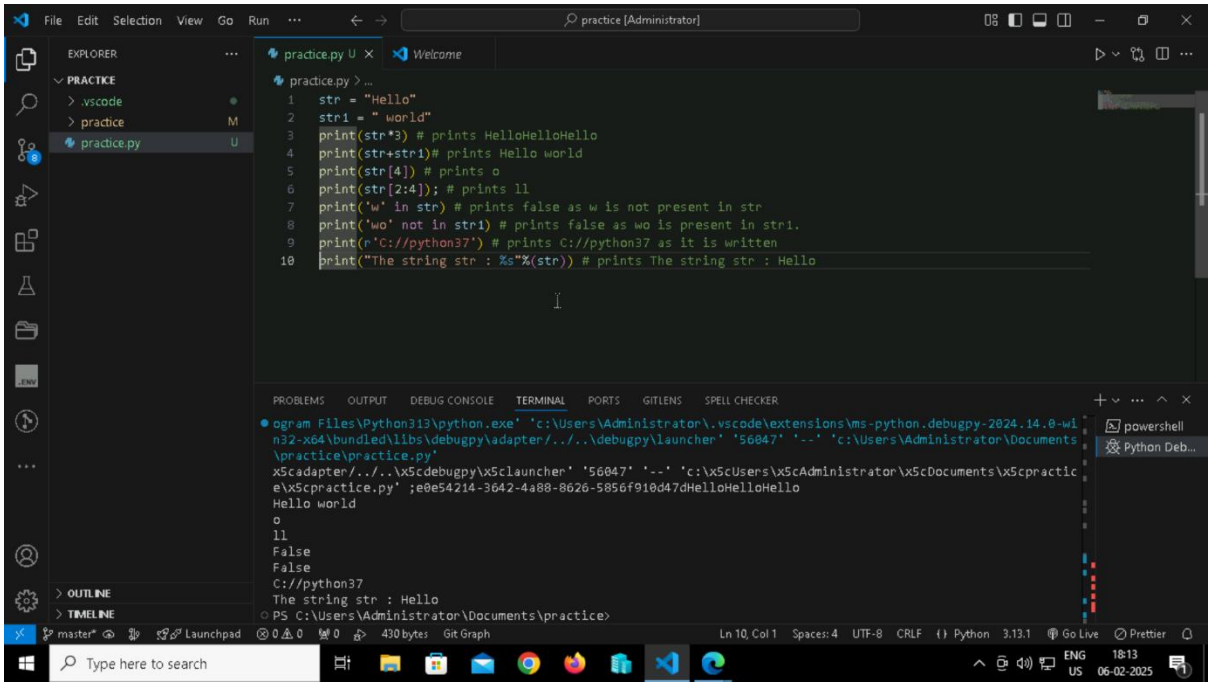

**Output:**



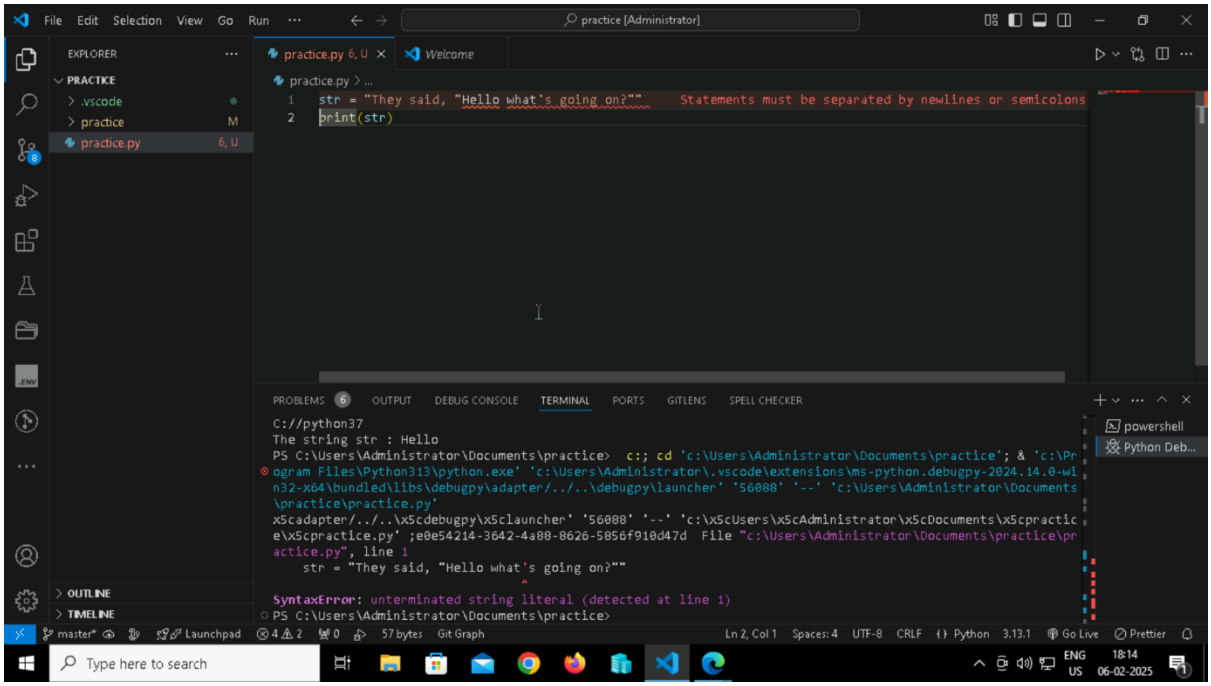

## 67) Python String Formatting



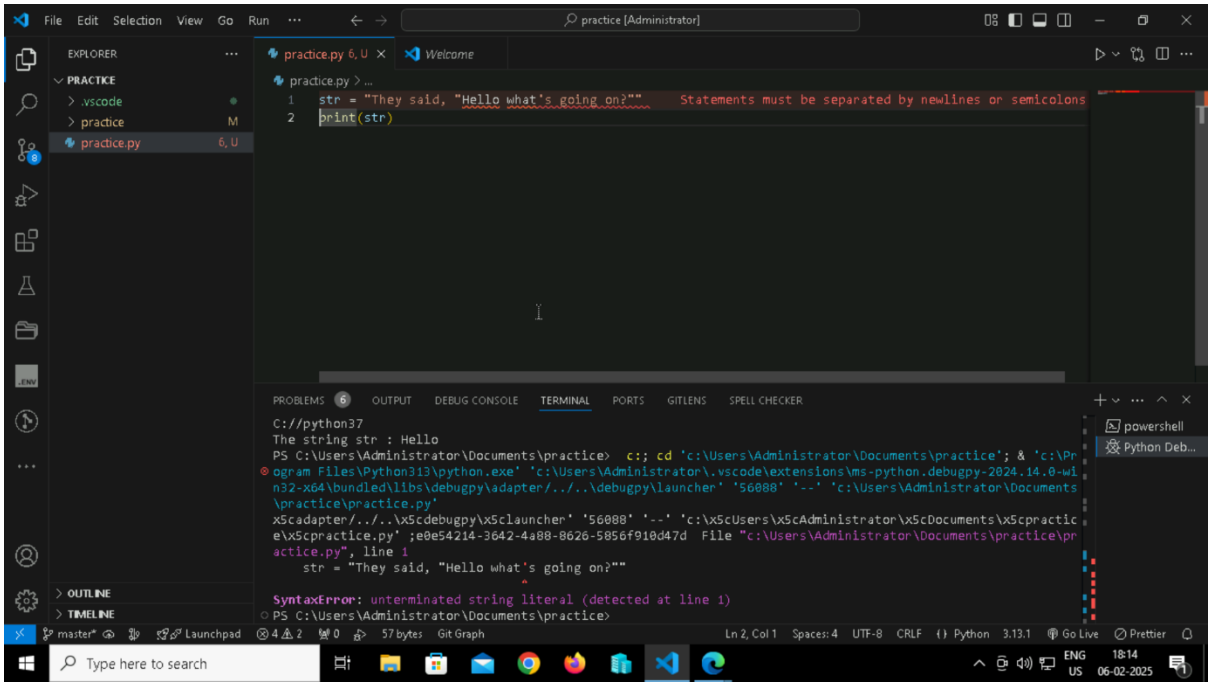

**Output:**



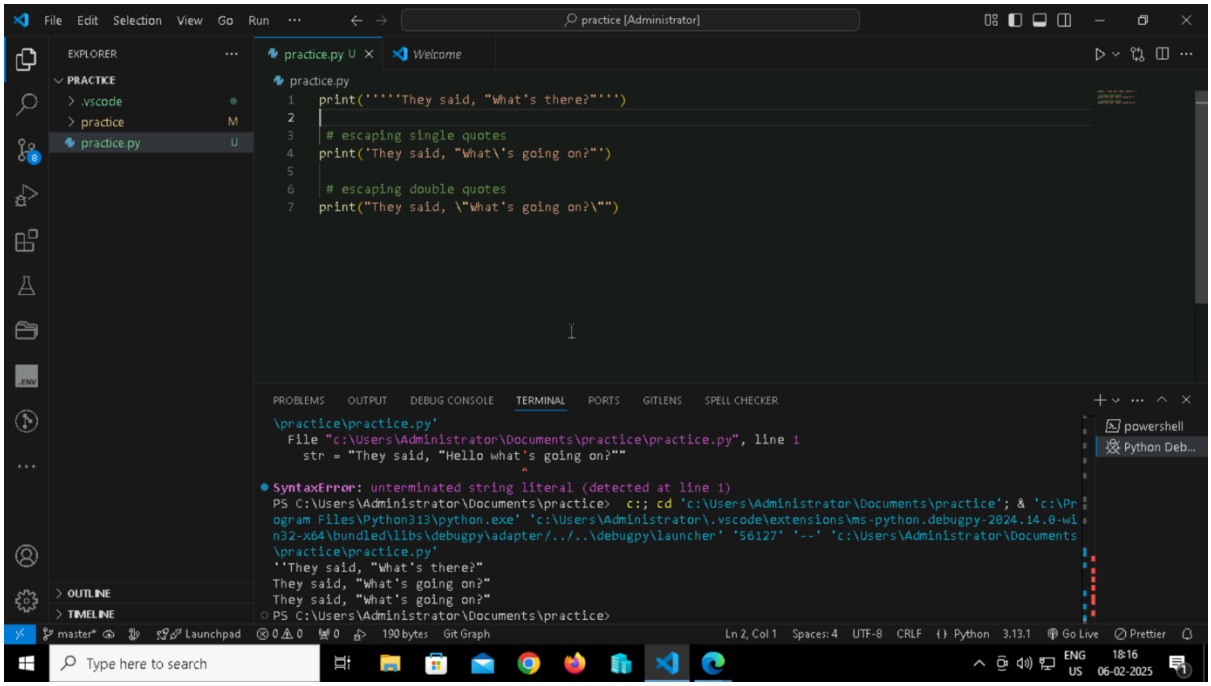

## 68) Python String Formatting



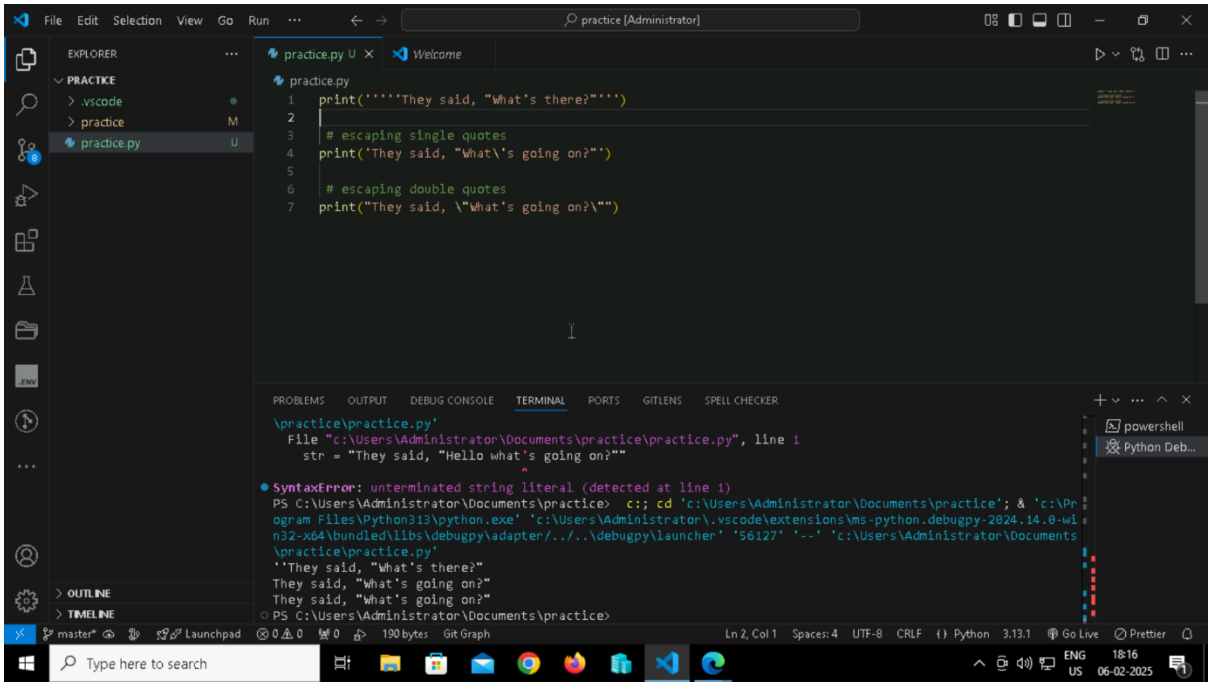

**Output:**



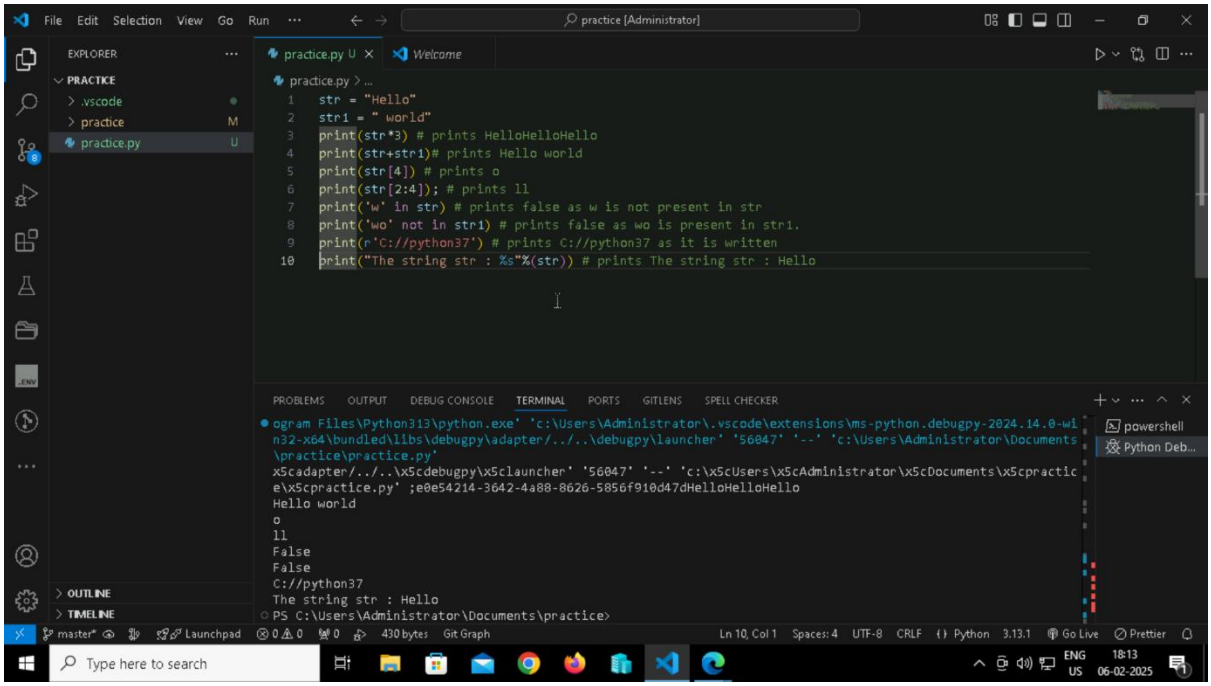

### 69) format() method



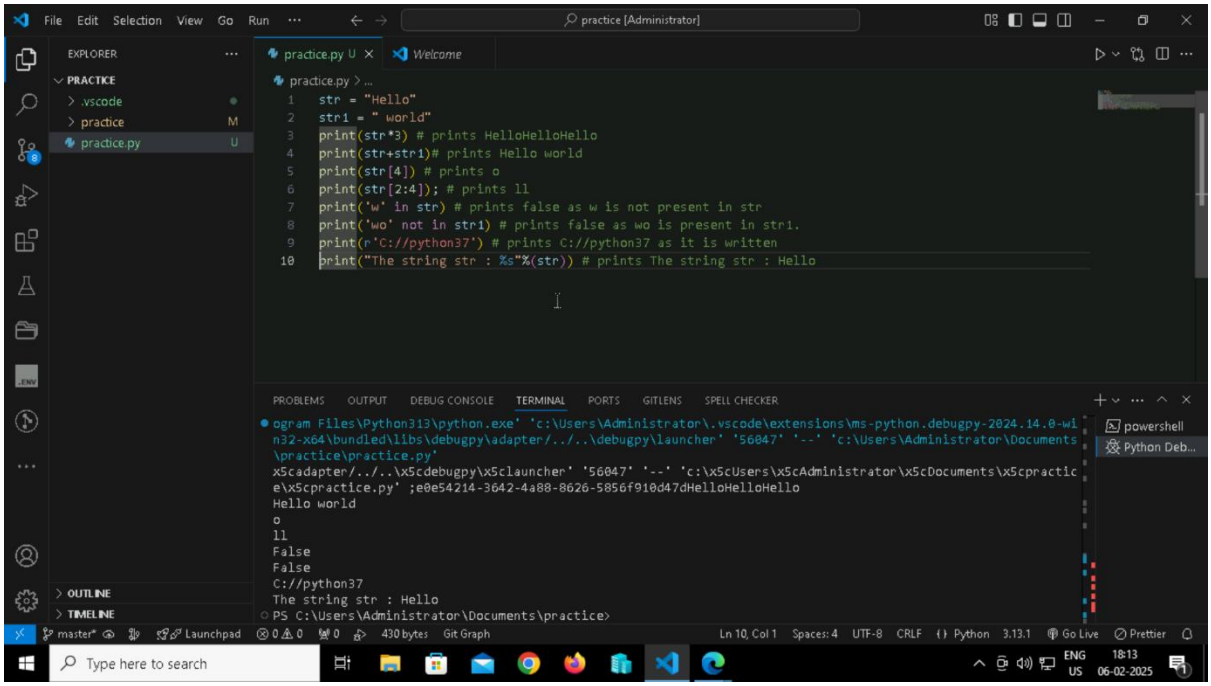

**Output:**



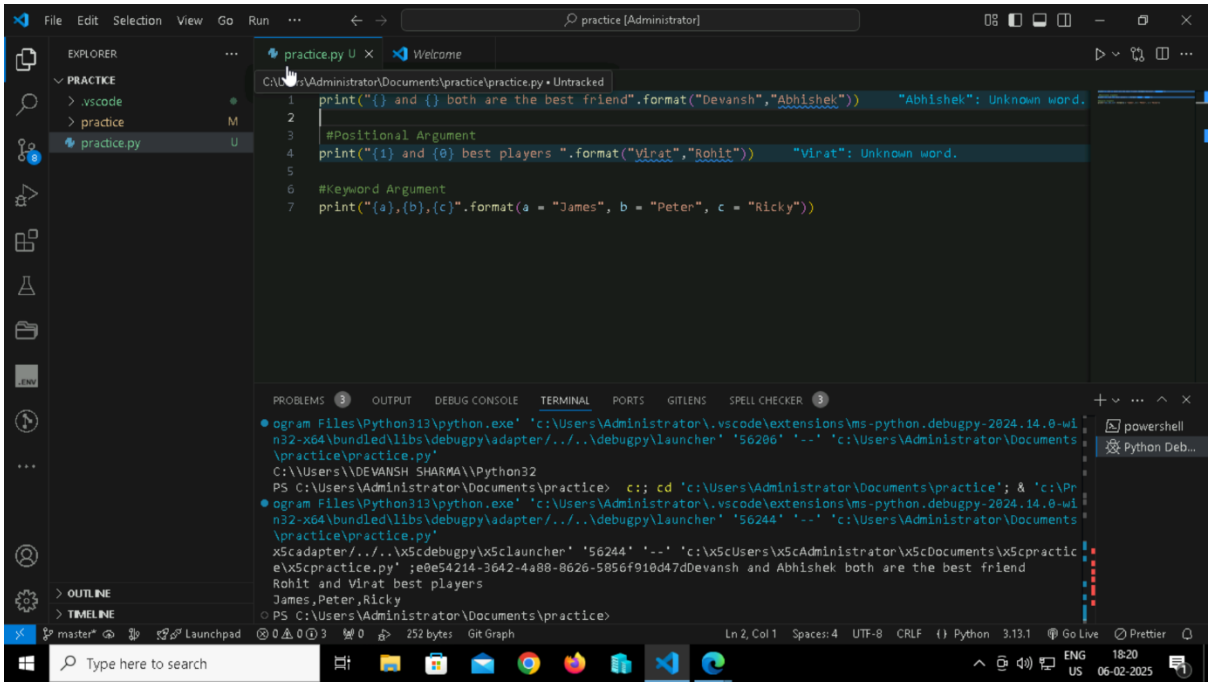

### 70) format() method



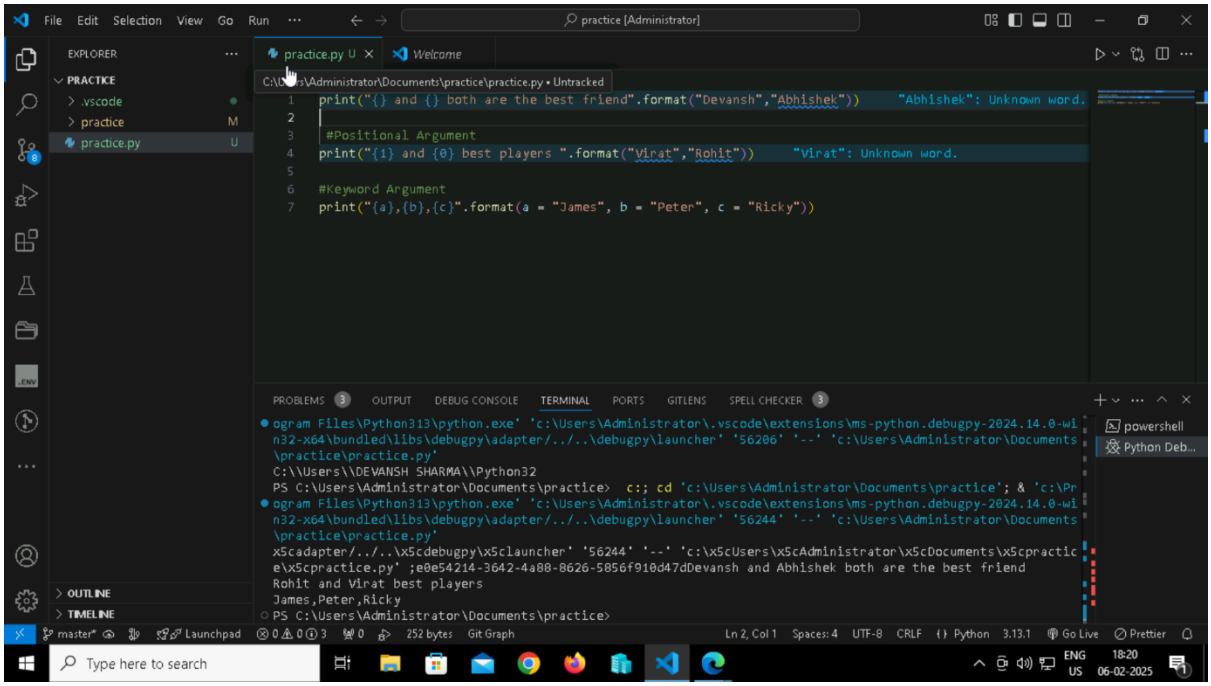

**Output:**



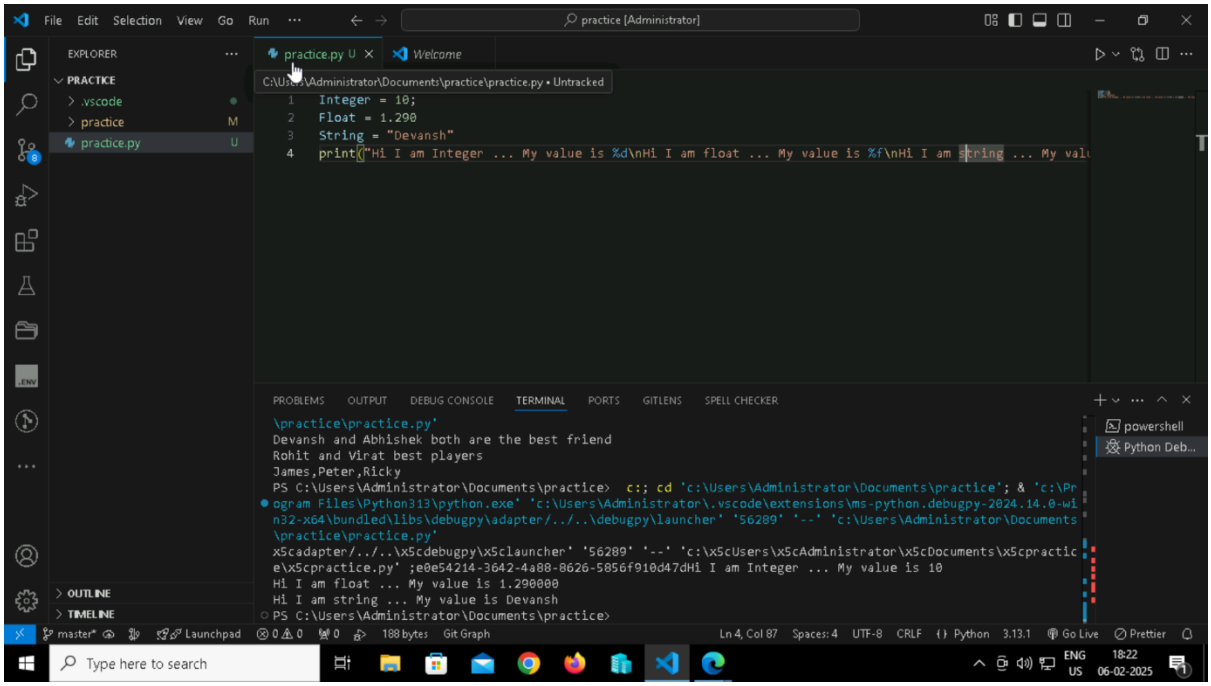

### 71) Formatting Using % Operator



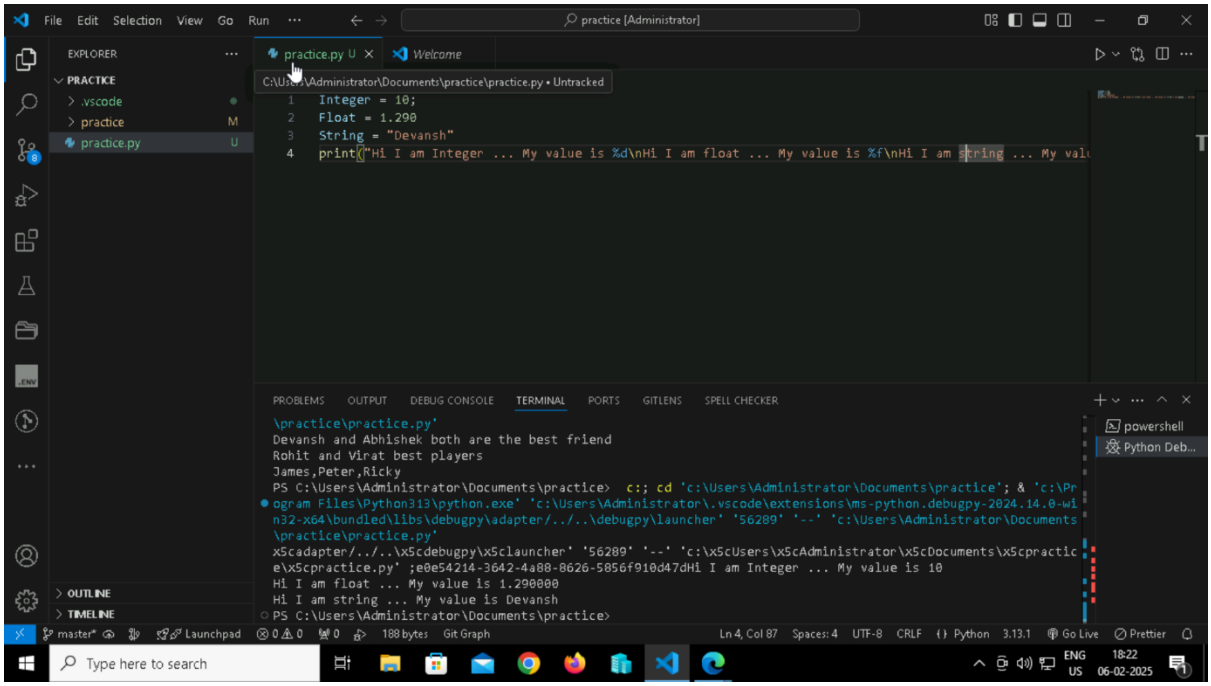

**Output:**



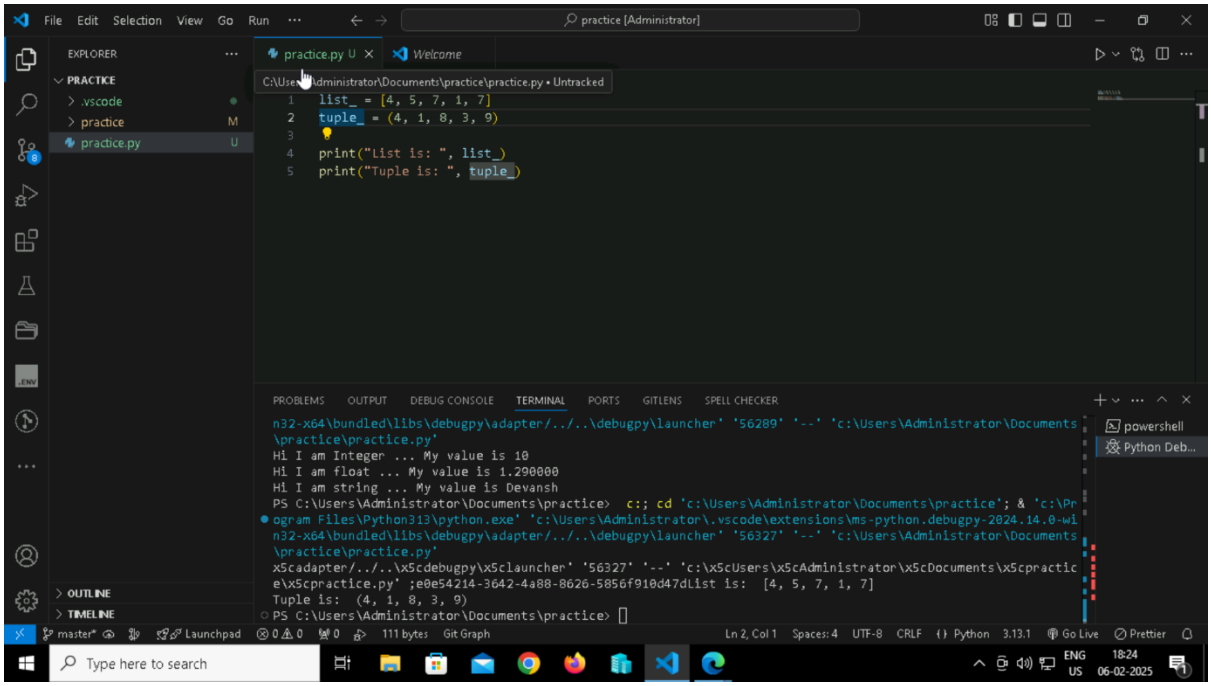

## 72) Formatting Using % Operator



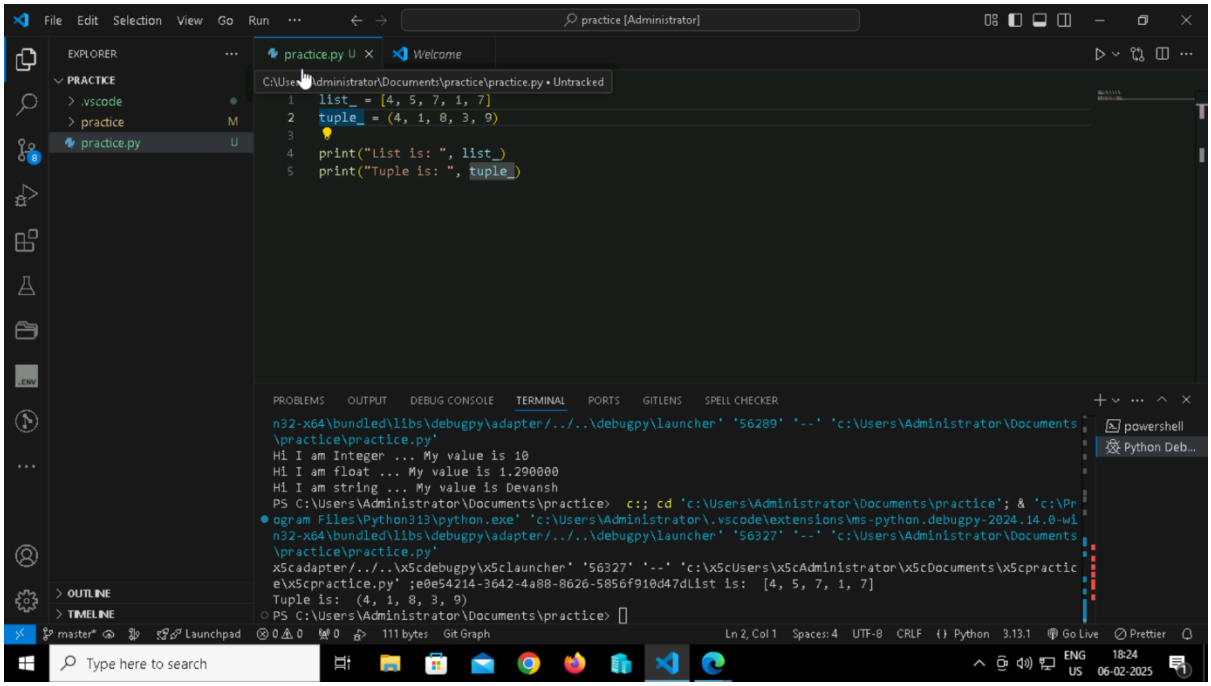

**Output:**



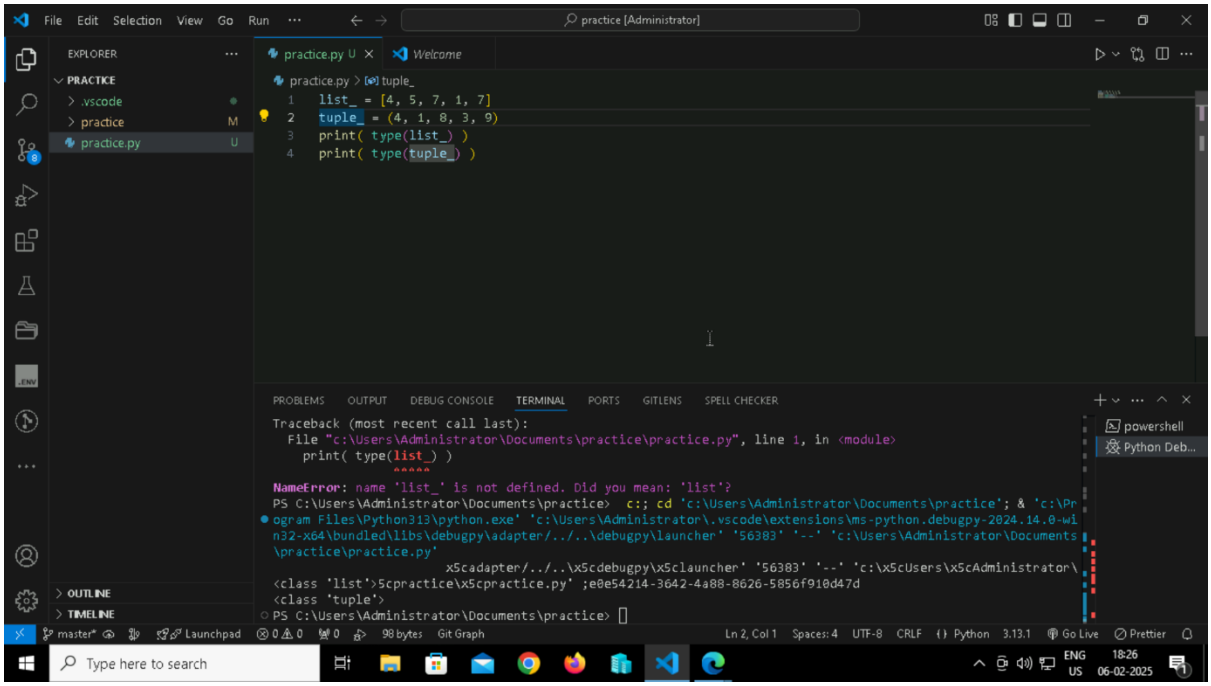

### 73) Difference between creating a list and a tuple



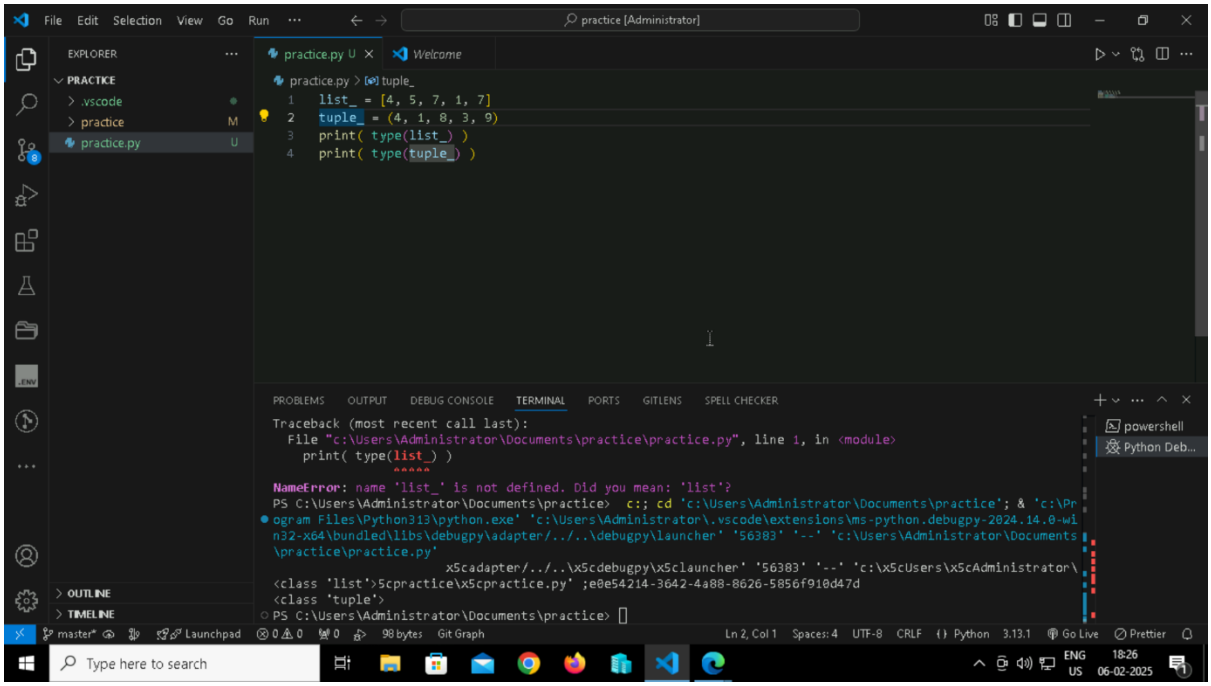

**Output:**



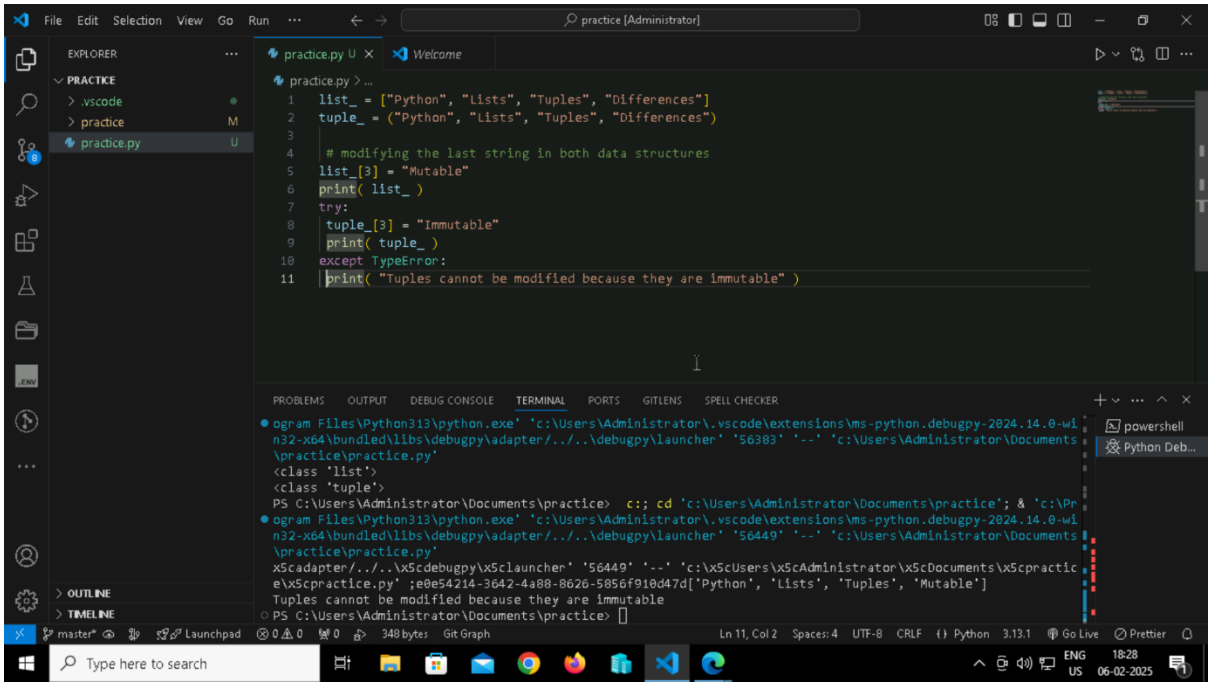

## 74) type() function



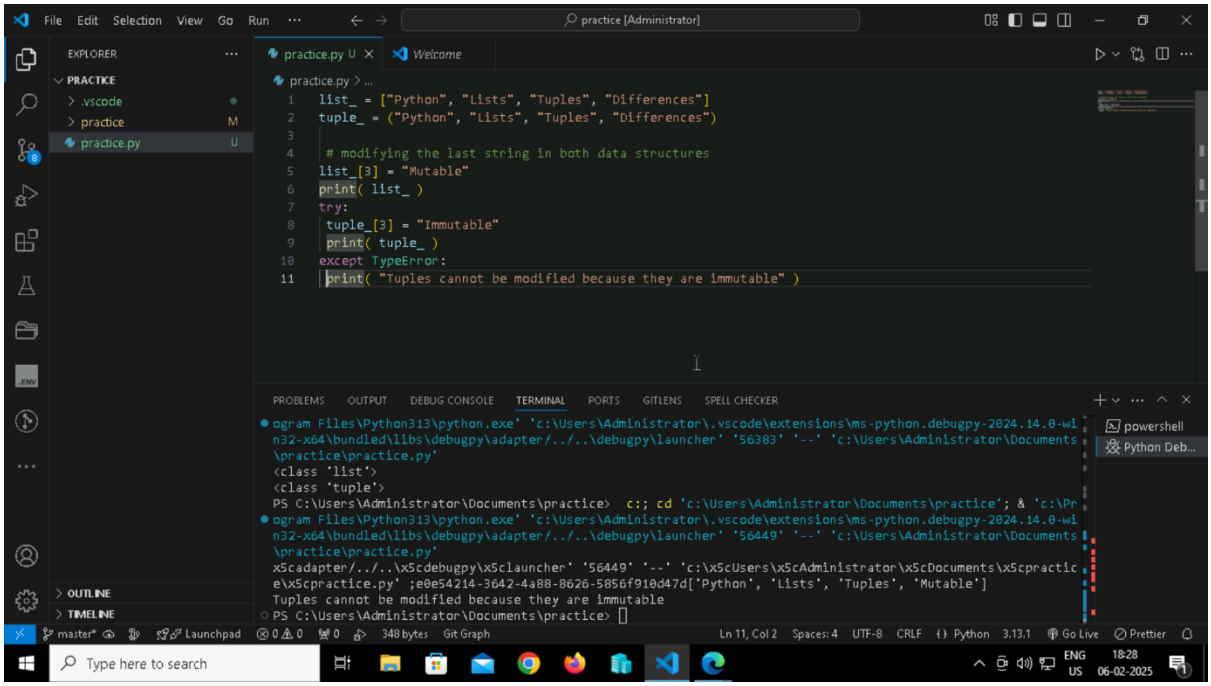

**Output:**



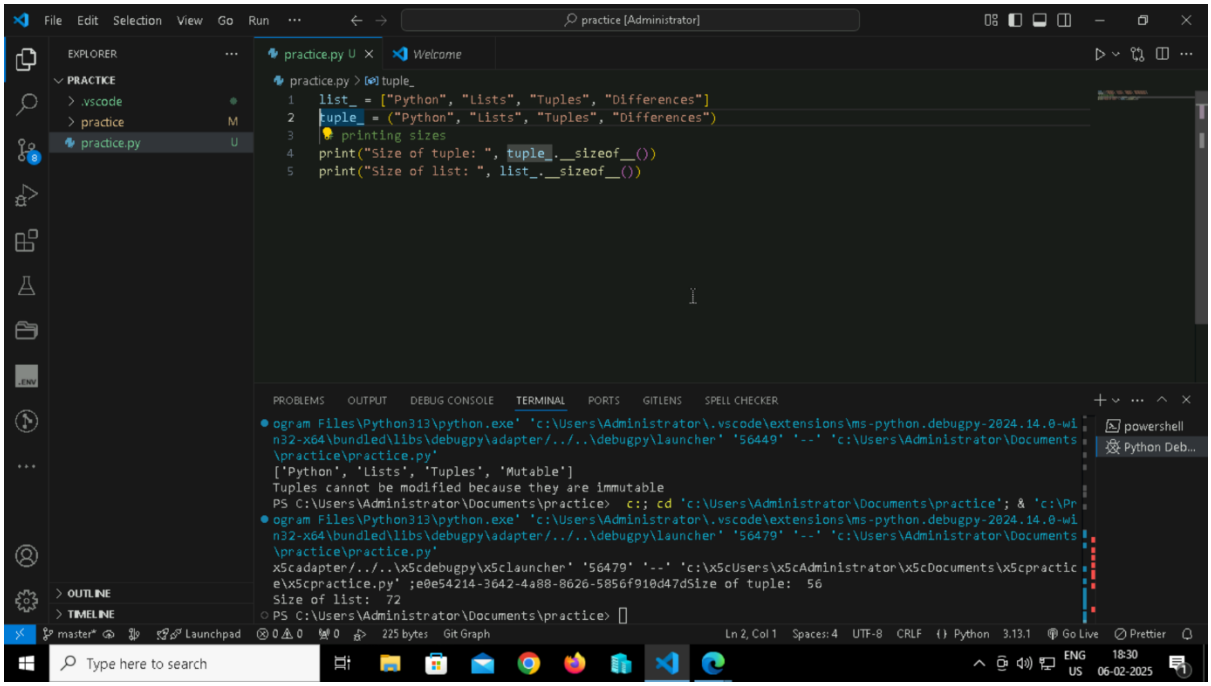

**75) difference between lists and tuples in immutability and mutability.**



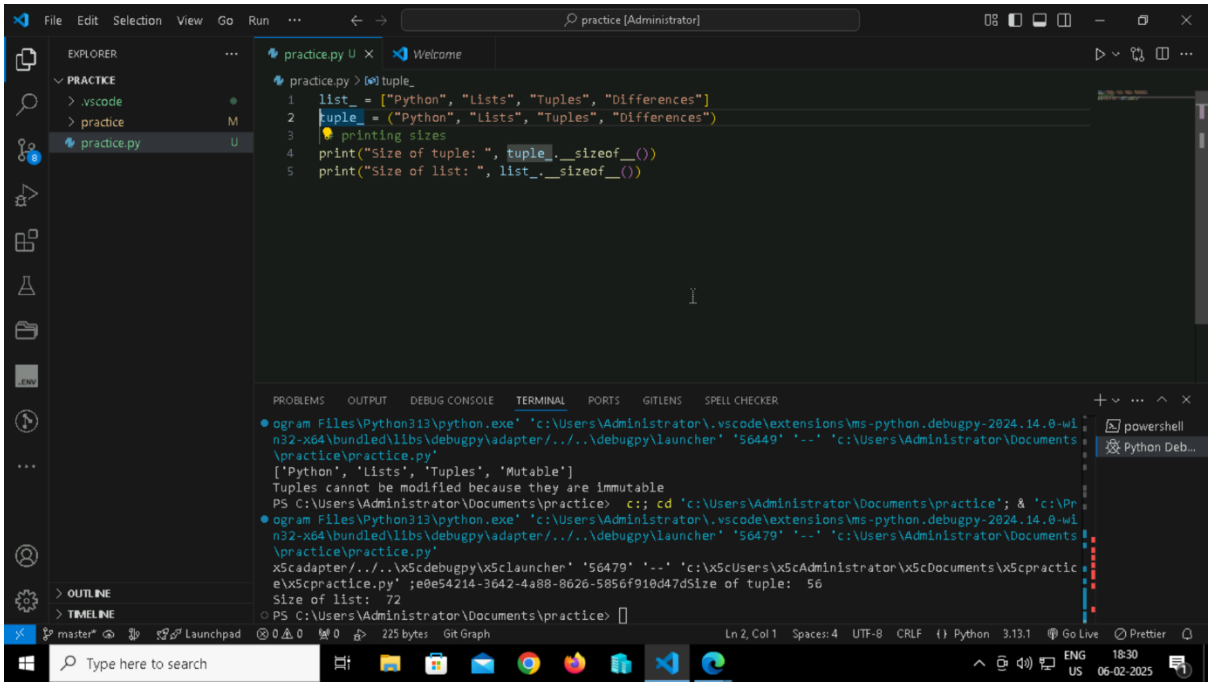

**Output:**




### 76) Size Difference


practice [Administrator]
practice.py U
Welcome
practice.py > tuple_
1 list_ = ["Python", "Lists", "Tuples", "Differences"]
2 tuple_ = ("Python", "Lists", "Tuples", "Differences")
3 printing sizes
4 print("Size of tuple: ", tuple_.__sizeof__())
5 print("Size of list: ", list_.__sizeof__())
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable
Size of tuple: 56
Size of list: 72
PS C:\Users\Administrator\Documents\practice>

## Output:


practice [Administrator]
practice.py U
Welcome
practice.py > tuple_
1 list_ = ["Python", "Lists", "Tuples", "Differences"]
2 tuple_ = ("Python", "Lists", "Tuples", "Differences")
3 printing sizes
4 print("Size of tuple: ", tuple_.__sizeof__())
5 print("Size of list: ", list_.__sizeof__())
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable
Size of tuple: 56
Size of list: 72
PS C:\Users\Administrator\Documents\practice>