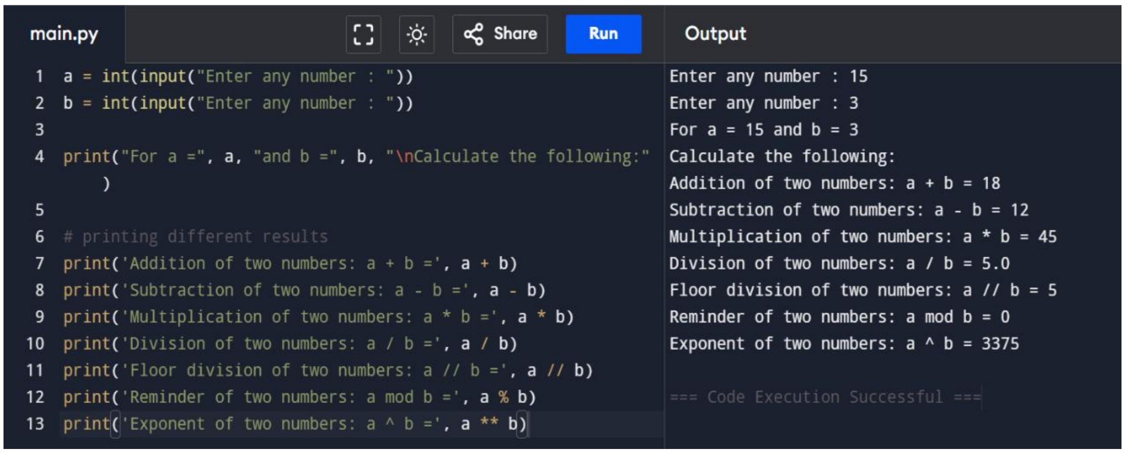
Thumati Pavan Venkata Narendra Kumar

EMP-ID-289219

PYTHON-BASICS-ASSIGNMENTS-1

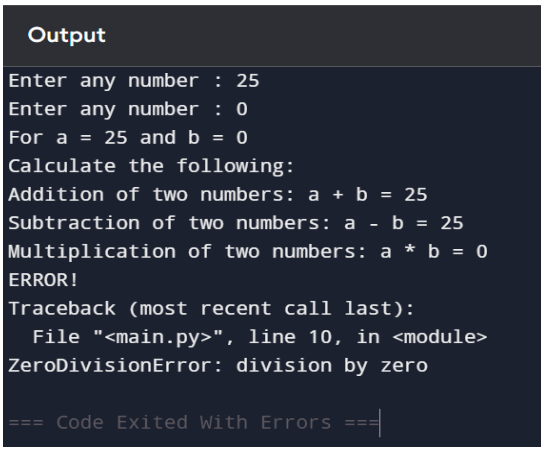
1.Operators in python.

1. Arithmetic Operators.

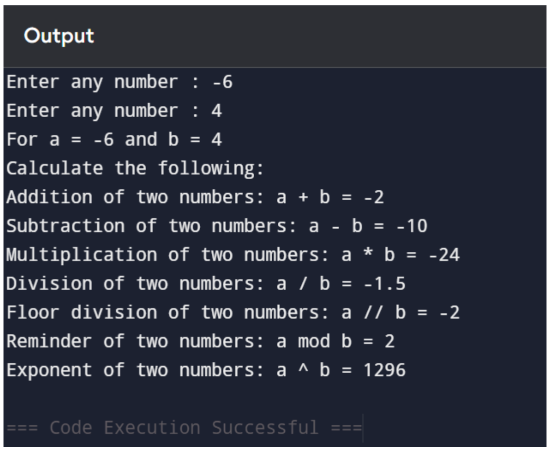


```python
a = int(input("Enter any number : "))
b = int(input("Enter any number : "))

print("For a =", a, "and b =", b, "\nCalculate the following:")

# printing different results
print('Addition of two numbers: a + b =', a + b)
print('Subtraction of two numbers: a - b =', a - b)
print('Multiplication of two numbers: a * b =', a * b)
print('Division of two numbers: a / b =', a / b)
print('Floor division of two numbers: a // b =', a // b)
print('Reminder of two numbers: a mod b =', a % b)
print('Exponent of two numbers: a ^ b =', a ** b)
```

```
Enter any number : 15
Enter any number : 3
For a = 15 and b = 3
Calculate the following:
Addition of two numbers: a + b = 18
Subtraction of two numbers: a - b = 12
Multiplication of two numbers: a * b = 45
Division of two numbers: a / b = 5.0
Floor division of two numbers: a // b = 5
Reminder of two numbers: a mod b = 0
Exponent of two numbers: a ^ b = 3375

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

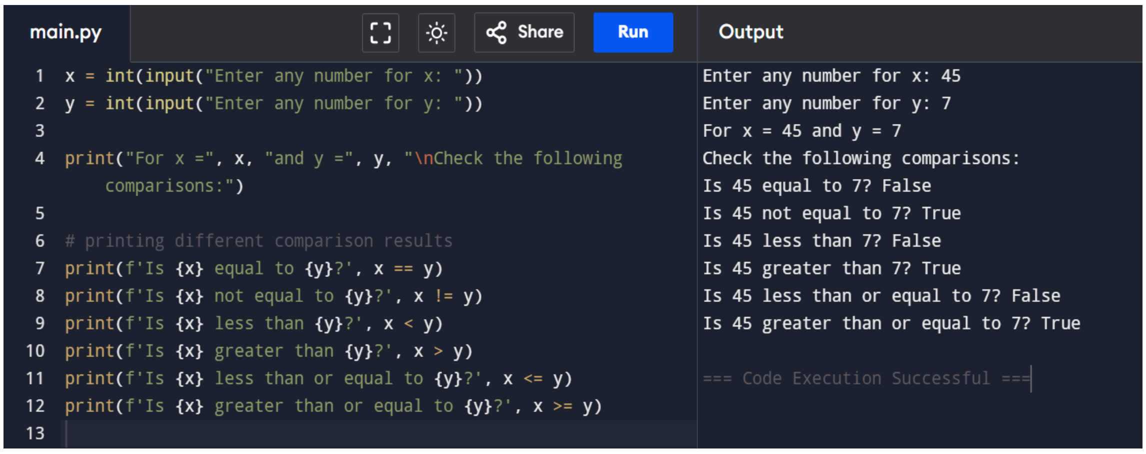


```
Enter any number : 25
Enter any number : 0
For a = 25 and b = 0
Calculate the following:
Addition of two numbers: a + b = 25
Subtraction of two numbers: a - b = 25
Multiplication of two numbers: a * b = 0
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 10, in <module>
ZeroDivisionError: division by zero

=== Code Exited With Errors ===
```



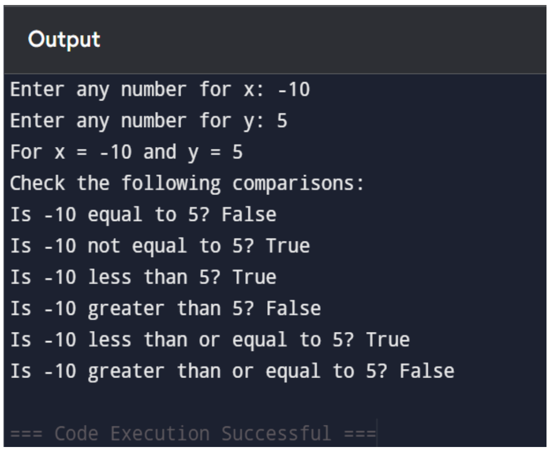
```
Enter any number : -6
Enter any number : 4
For a = -6 and b = 4
Calculate the following:
Addition of two numbers: a + b = -2
Subtraction of two numbers: a - b = -10
Multiplication of two numbers: a * b = -24
Division of two numbers: a / b = -1.5
Floor division of two numbers: a // b = -2
Reminder of two numbers: a mod b = 2
Exponent of two numbers: a ^ b = 1296

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

2. Comparison Operators.

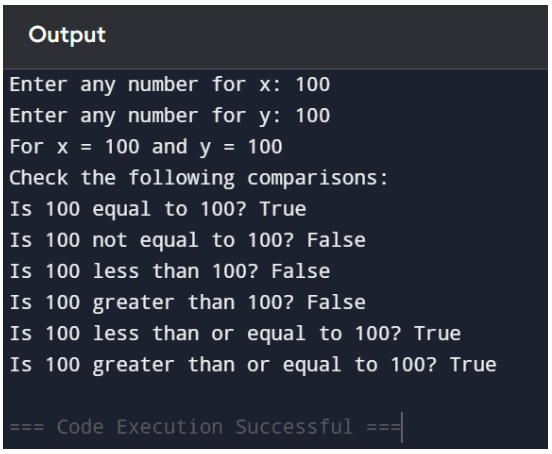


```python
x = int(input("Enter any number for x: "))
y = int(input("Enter any number for y: "))

print("For x =", x, "and y =", y, "\nCheck the following comparisons:")

# printing different comparison results
print(f'Is {x} equal to {y}?', x == y)
print(f'Is {x} not equal to {y}?', x != y)
print(f'Is {x} less than {y}?', x < y)
print(f'Is {x} greater than {y}?', x > y)
print(f'Is {x} less than or equal to {y}?', x <= y)
print(f'Is {x} greater than or equal to {y}?', x >= y)
```

```
Enter any number for x: 45
Enter any number for y: 7
For x = 45 and y = 7
Check the following comparisons:
Is 45 equal to 7? False
Is 45 not equal to 7? True
Is 45 less than 7? False
Is 45 greater than 7? True
Is 45 less than or equal to 7? False
Is 45 greater than or equal to 7? True

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

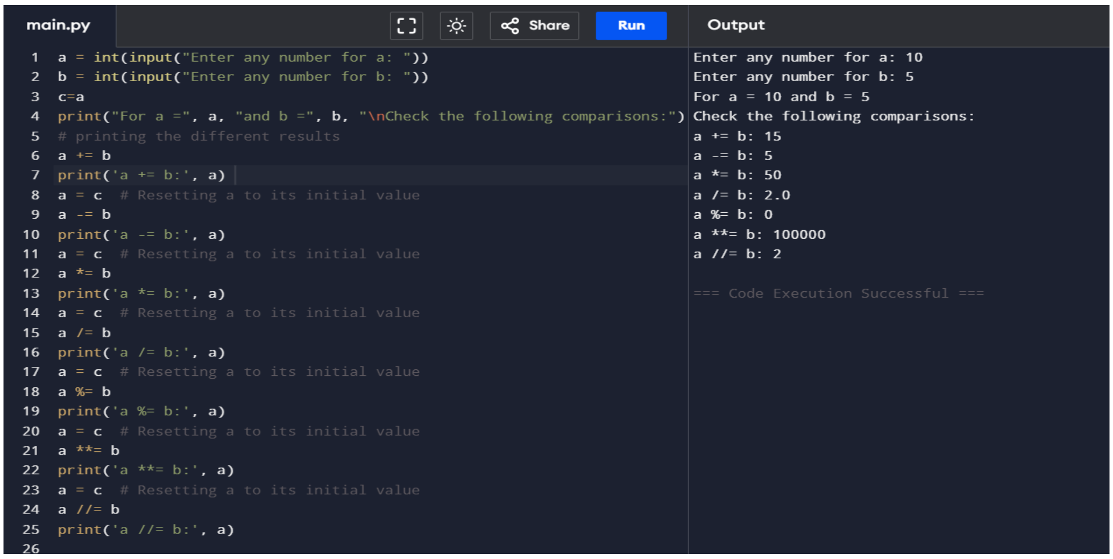


```
Enter any number for x: -10
Enter any number for y: 5
For x = -10 and y = 5
Check the following comparisons:
Is -10 equal to 5? False
Is -10 not equal to 5? True
Is -10 less than 5? True
Is -10 greater than 5? False
Is -10 less than or equal to 5? True
Is -10 greater than or equal to 5? False

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



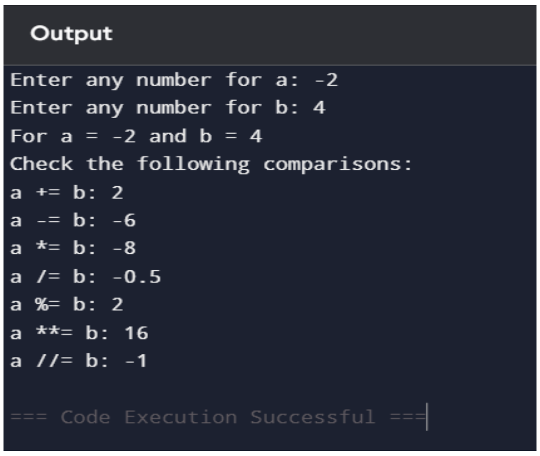
```
Enter any number for x: 100
Enter any number for y: 100
For x = 100 and y = 100
Check the following comparisons:
Is 100 equal to 100? True
Is 100 not equal to 100? False
Is 100 less than 100? False
Is 100 greater than 100? False
Is 100 less than or equal to 100? True
Is 100 greater than or equal to 100? True

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

3. Assignment Operators.

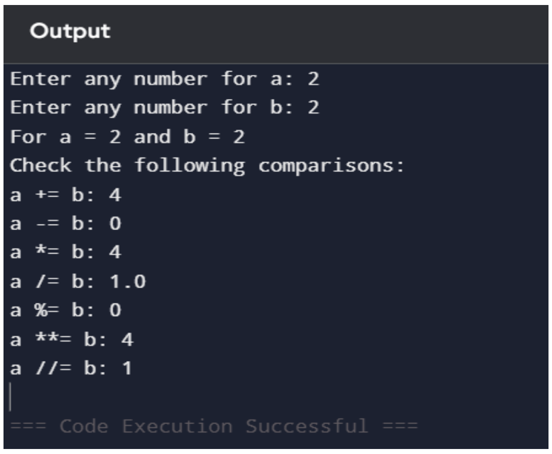


```python
a = int(input("Enter any number for a: "))
b = int(input("Enter any number for b: "))
c=a
print("For a =", a, "and b =", b, "\nCheck the following comparisons:")
# printing the different results
a += b
print('a += b:', a)
a = c  # Resetting a to its initial value
a -= b
print('a -= b:', a)
a = c  # Resetting a to its initial value
a *= b
print('a *= b:', a)
a = c  # Resetting a to its initial value
a /= b
print('a /= b:', a)
a = c  # Resetting a to its initial value
a %= b
print('a %= b:', a)
a = c  # Resetting a to its initial value
a **= b
print('a **= b:', a)
a = c  # Resetting a to its initial value
a //= b
print('a //= b:', a)
```

```
Enter any number for a: 10
Enter any number for b: 5
For a = 10 and b = 5
Check the following comparisons:
a += b: 15
a -= b: 5
a *= b: 50
a /= b: 2.0
a %= b: 0
a **= b: 100000
a //= b: 2

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

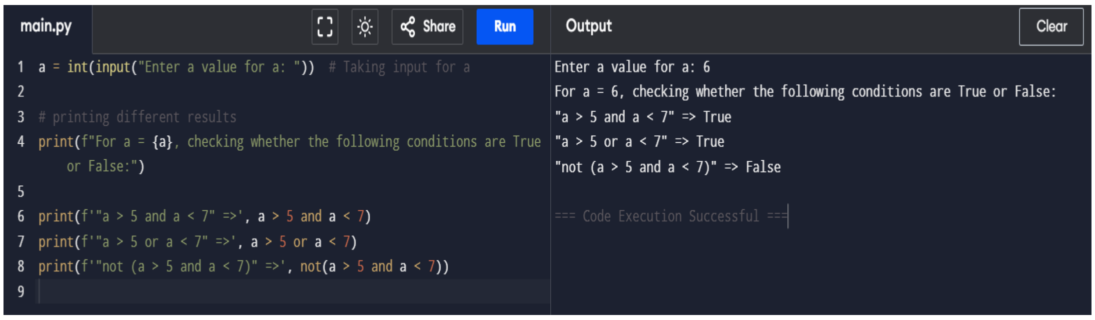


```
Enter any number for a: -2
Enter any number for b: 4
For a = -2 and b = 4
Check the following comparisons:
a += b: 2
a -= b: -6
a *= b: -8
a /= b: -0.5
a %= b: 2
a **= b: 16
a //= b: -1

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



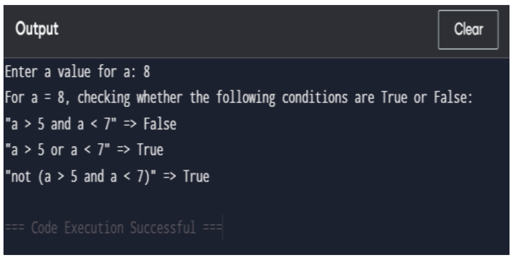
```
Enter any number for a: 2
Enter any number for b: 2
For a = 2 and b = 2
Check the following comparisons:
a += b: 4
a -= b: 0
a *= b: 4
a /= b: 1.0
a %= b: 0
a **= b: 4
a //= b: 1

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

4. Logical Operators.

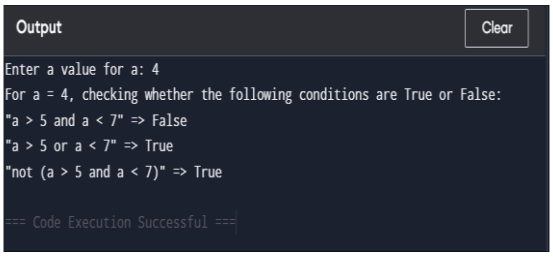


```python
a = int(input("Enter a value for a: "))  # Taking input for a

# printing different results
print(f"For a = {a}, checking whether the following conditions are True or False:")

print(f'"a > 5 and a < 7" =>', a > 5 and a < 7)
print(f'"a > 5 or a < 7" =>', a > 5 or a < 7)
print(f'"not (a > 5 and a < 7)" =>', not(a > 5 and a < 7))
```

```
Enter a value for a: 6
For a = 6, checking whether the following conditions are True or False:
"a > 5 and a < 7" => True
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => False

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

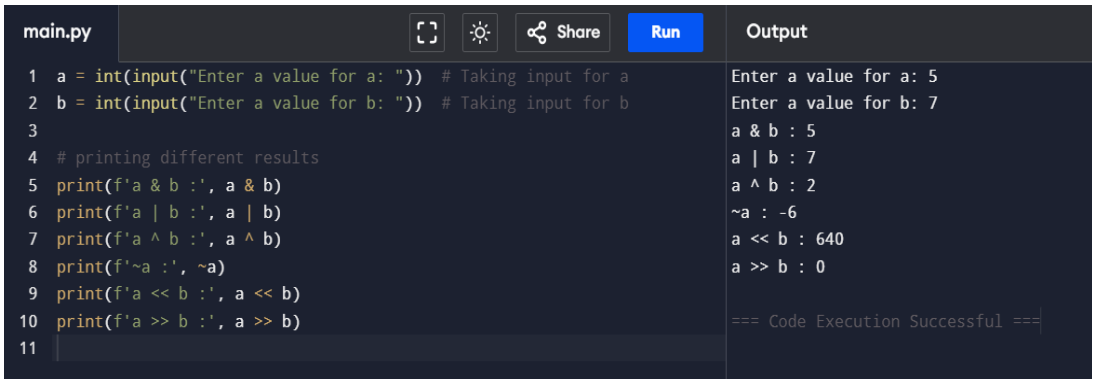


```
Enter a value for a: 8
For a = 8, checking whether the following conditions are True or False:
"a > 5 and a < 7" => False
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => True

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



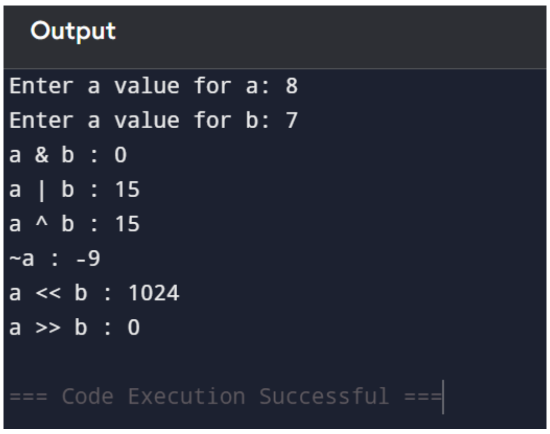
```
Enter a value for a: 4
For a = 4, checking whether the following conditions are True or False:
"a > 5 and a < 7" => False
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => True

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

5. Bitwise Operators.

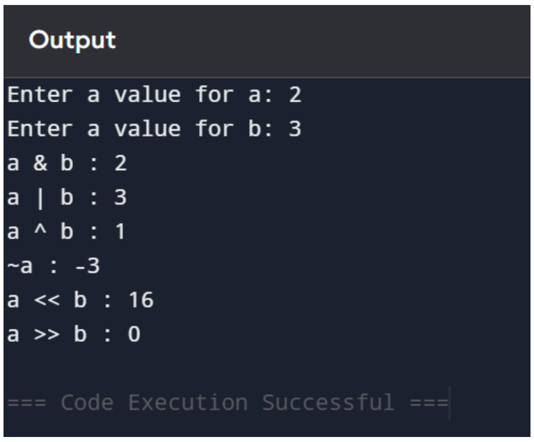


```python
a = int(input("Enter a value for a: "))  # Taking input for a
b = int(input("Enter a value for b: "))  # Taking input for b

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

```
Enter a value for a: 5
Enter a value for b: 7
a & b : 5
a | b : 7
a ^ b : 2
~a : -6
a << b : 640
a >> b : 0

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

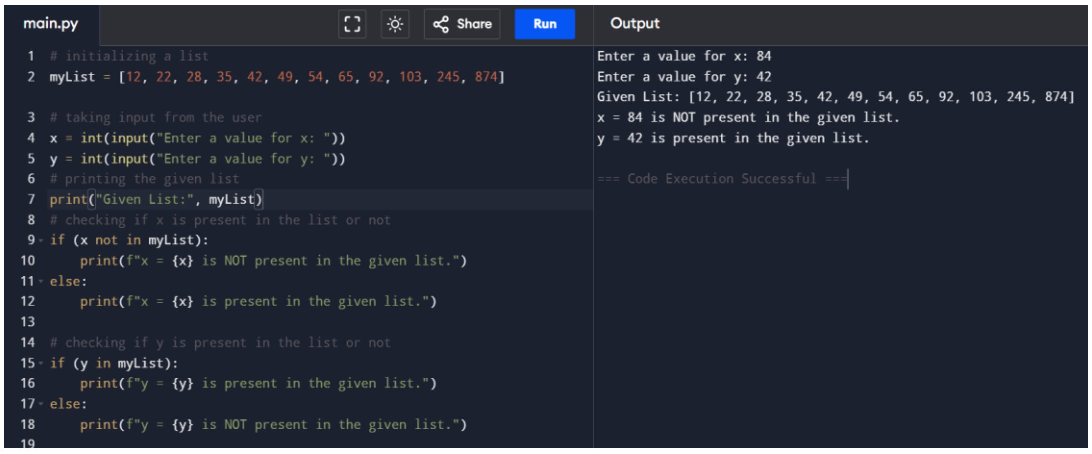


```
Enter a value for a: 8
Enter a value for b: 7
a & b : 0
a | b : 15
a ^ b : 15
~a : -9
a << b : 1024
a >> b : 0

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



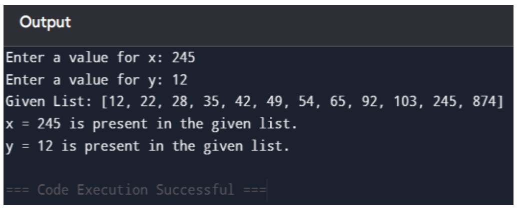
```
Enter a value for a: 2
Enter a value for b: 3
a & b : 2
a | b : 3
a ^ b : 1
~a : -3
a << b : 16
a >> b : 0

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

6. Membership Operators.

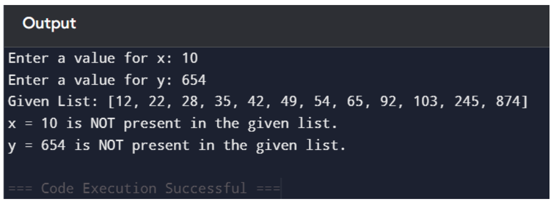


```python
# initializing a list
myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]

# taking input from the user
x = int(input("Enter a value for x: "))
y = int(input("Enter a value for y: "))
# printing the given list
print("Given List:", myList)
# checking if x is present in the list or not
if (x not in myList):
    print(f"x = {x} is NOT present in the given list.")
else:
    print(f"x = {x} is present in the given list.")

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

```
Enter a value for x: 84
Enter a value for y: 42
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 84 is NOT present in the given list.
y = 42 is present in the given list.

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

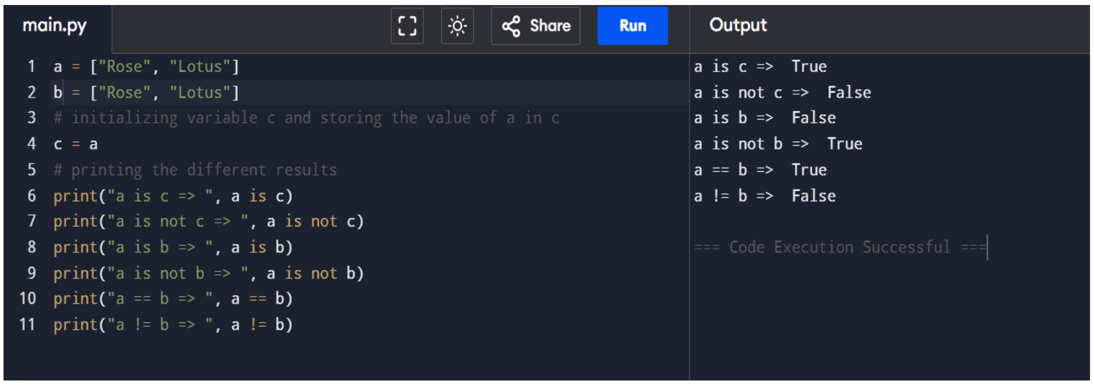


```
Enter a value for x: 245
Enter a value for y: 12
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 245 is present in the given list.
y = 12 is present in the given list.

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



```
Enter a value for x: 10
Enter a value for y: 654
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 10 is NOT present in the given list.
y = 654 is NOT present in the given list.

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

7. Identity Operators.

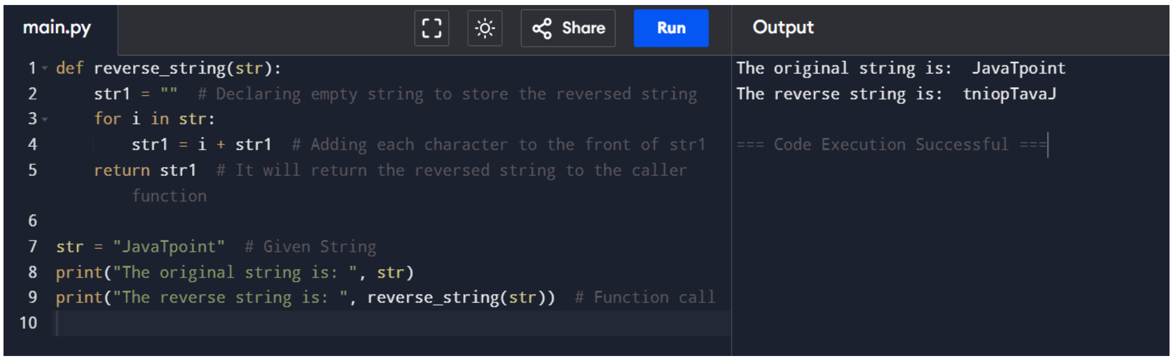


```python
a = ["Rose", "Lotus"]
b = ["Rose", "Lotus"]
# initializing variable c and storing the value of a in c
c = a
# printing the different results
print("a is c => ", a is c)
print("a is not c => ", a is not c)
print("a is b => ", a is b)
print("a is not b => ", a is not b)
print("a == b => ", a == b)
print("a != b => ", a != b)
```

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

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

2. Reversing a string in Python

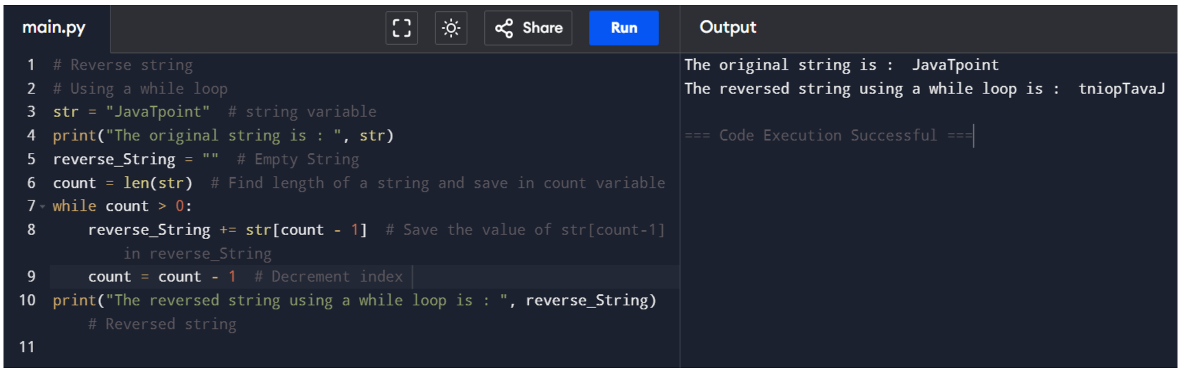
1. Using for loop.



```python
def reverse_string(str):
    str1 = ""   # Declaring empty string to store the reversed string
    for i in str:
        str1 = i + str1  # Adding each character to the front of str1
    return str1  # It will return the reversed 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

```

```
The original string is:  JavaTpoint
The reverse string is:  tniopTavaJ

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

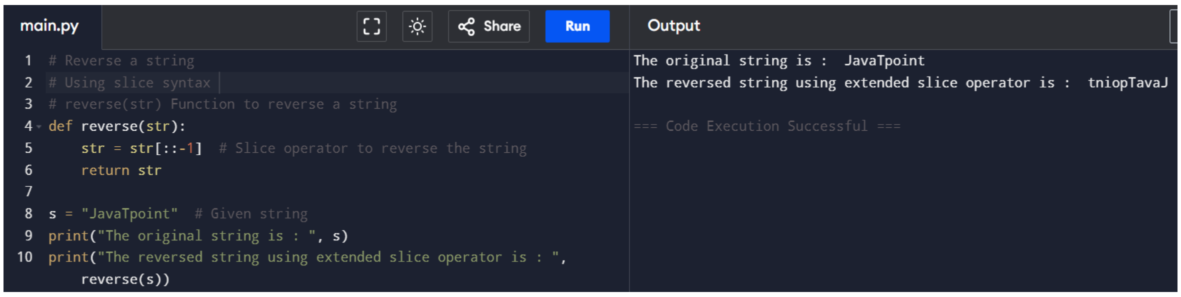
2. Using while loop.



```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 reverse_String
    count = count - 1  # Decrement index
print("The reversed string using a while loop is : ", reverse_String)
    # Reversed string

```

```
The original string is :  JavaTpoint
The reversed string using a while loop is :  tniopTavaJ

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

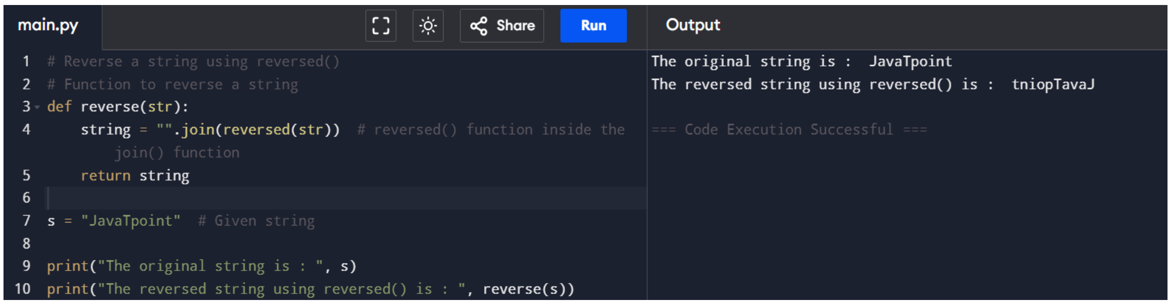
3. Using the slice operator.



```python
# Reverse a string
# Using slice syntax
# reverse(str) Function to reverse a string
def reverse(str):
    str = str[::-1]  # Slice operator to reverse the string
    return str

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

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

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

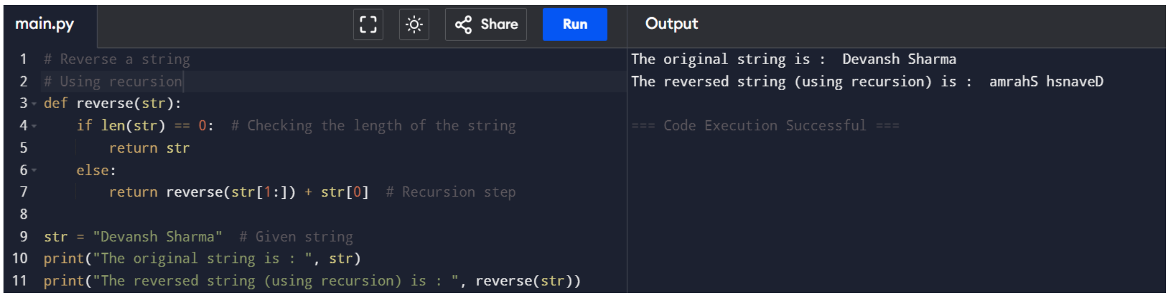
4. Using the reversed() function.



```python
# Reverse a string using reversed()
# Function to reverse a string
def reverse(str):
    string = "".join(reversed(str))  # reversed() function inside the
        join() function
    return string

s = "JavaTpoint"  # Given string

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

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

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

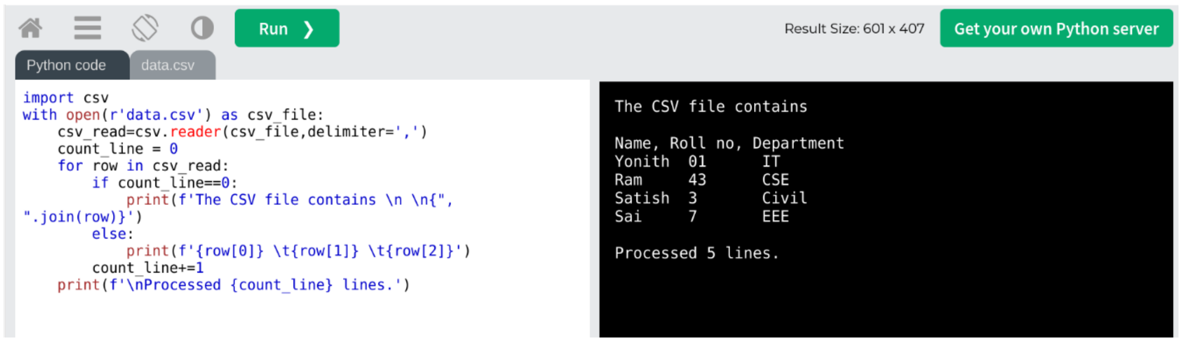
5. Using the recursion.



```python
# Reverse a string
# Using recursion
def reverse(str):
    if len(str) == 0:  # Checking the length of the string
        return str
    else:
        return reverse(str[1:]) + str[0]  # Recursion step

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

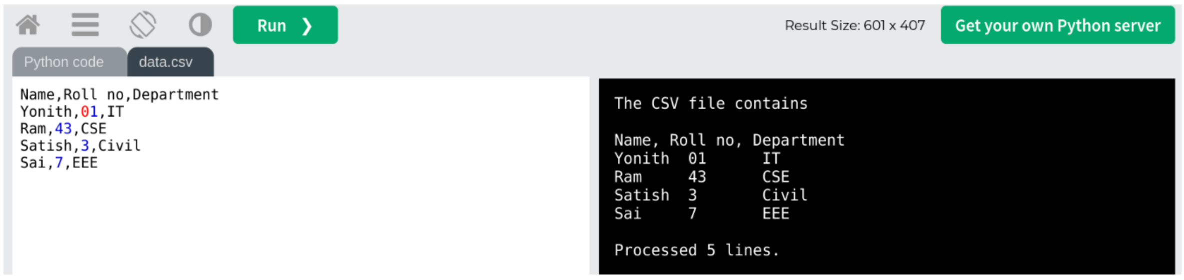
```
The original string is :  Devansh Sharma
The reversed string (using recursion) is :  amrahS hsnaveD

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

3. Reading CSV file in Python:



```python
import csv
with open(r'data.csv') as csv_file:
    csv_read=csv.reader(csv_file,delimiter=',')
    count_line = 0
    for row in csv_read:
        if count_line==0:
            print(f'The CSV file contains \n \n{",
".join(row)}')
        else:
            print(f'{row[0]} \t{row[1]} \t{row[2]}')
        count_line+=1
    print(f'\nProcessed {count_line} lines.')
```

```
The CSV file contains

Name, Roll no, Department
Yonith  01      IT
Ram     43      CSE
Satish  3       Civil
Sai     7       EEE

Processed 5 lines.
```

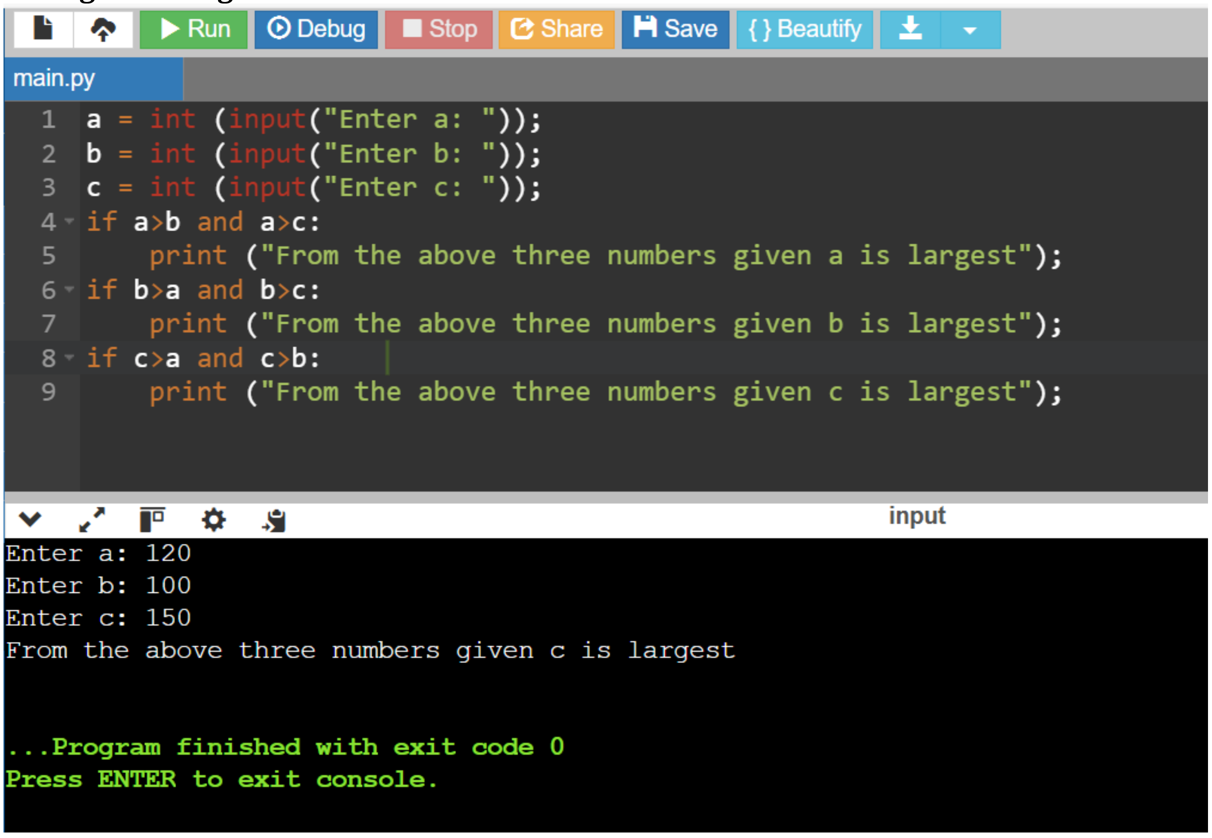


```
Name,Roll no,Department
Yonith,01,IT
Ram,43,CSE
Satish,3,Civil
Sai,7,EEE
```

```
The CSV file contains

Name, Roll no, Department
Yonith  01      IT
Ram     43      CSE
Satish  3       Civil
Sai     7       EEE

Processed 5 lines.
```

4.Python If Statement:

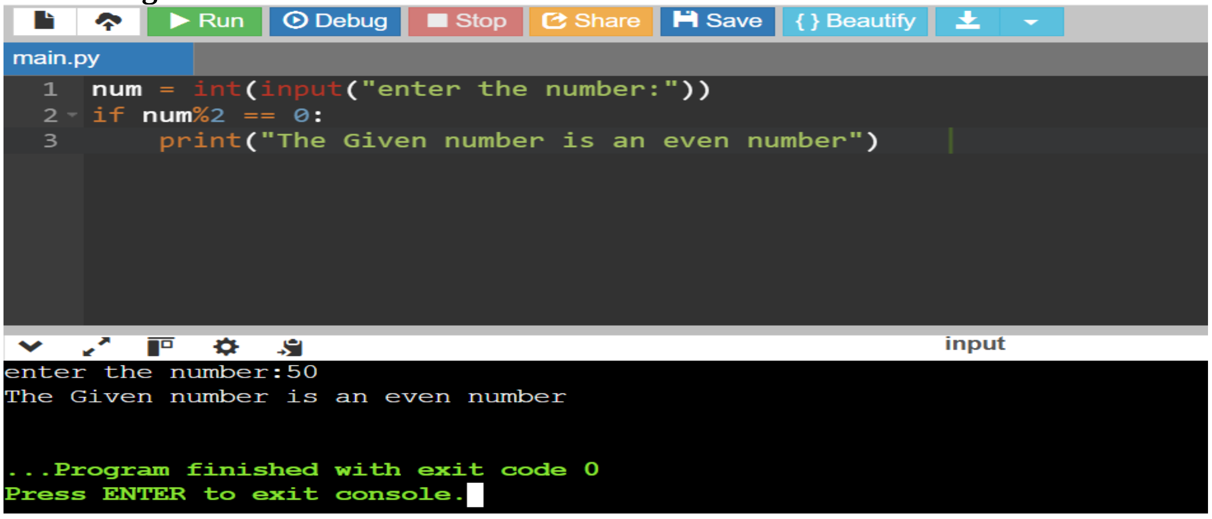
1.Largest among Three numbers`



```
a = int (input("Enter a: "));
b = int (input("Enter b: "));
c = int (input("Enter c: "));
if a>b and a>c:
    print ("From the above three numbers given a is largest");
if b>a and b>c:
    print ("From the above three numbers given b is largest");
if c>a and c>b:
    print ("From the above three numbers given c is largest");
```

```
Enter a: 120
Enter b: 100
Enter c: 150
From the above three numbers given c is largest


...Program finished with exit code 0
Press ENTER to exit console.
```

2.Checking either a number EVEN or not.

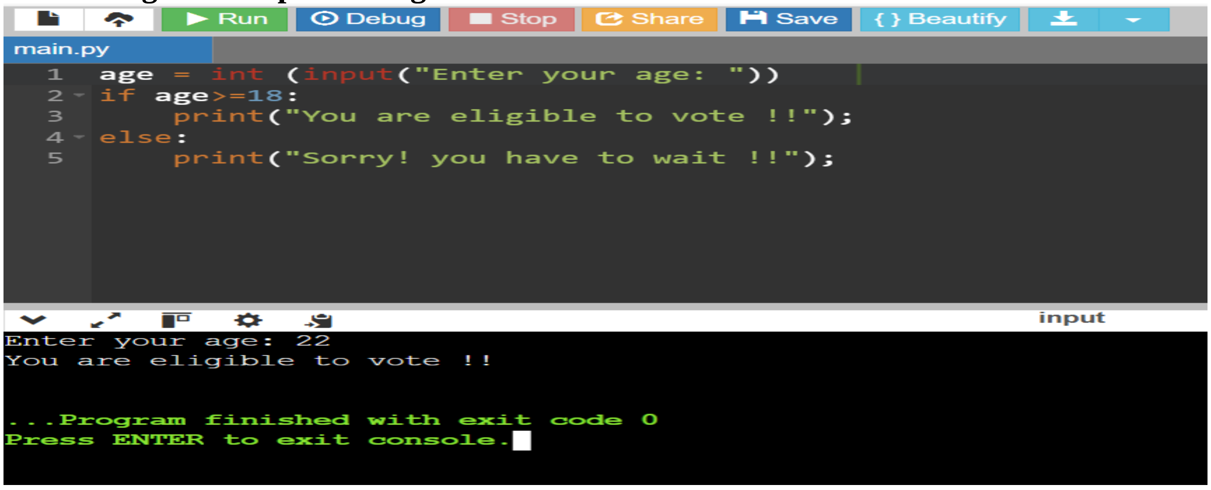


```
num = int(input("enter the number:"))
if num%2 == 0:
    print("The Given number is an even number")
```

```
enter the number:50
The Given number is an even number


...Program finished with exit code 0
Press ENTER to exit console.
```

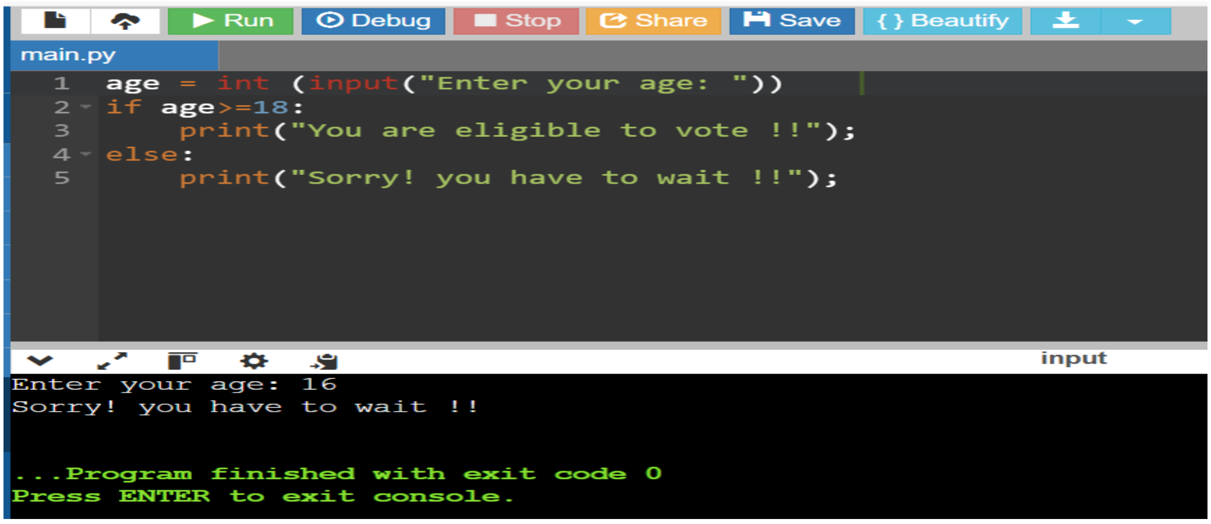
Python If-Else Statement:

1.Checking either a person eligible to vote or not.



```python
age = int (input("Enter your age: "))
if age>=18:
     print("You are eligible to vote !!");
else:
     print("Sorry! you have to wait !!");
```

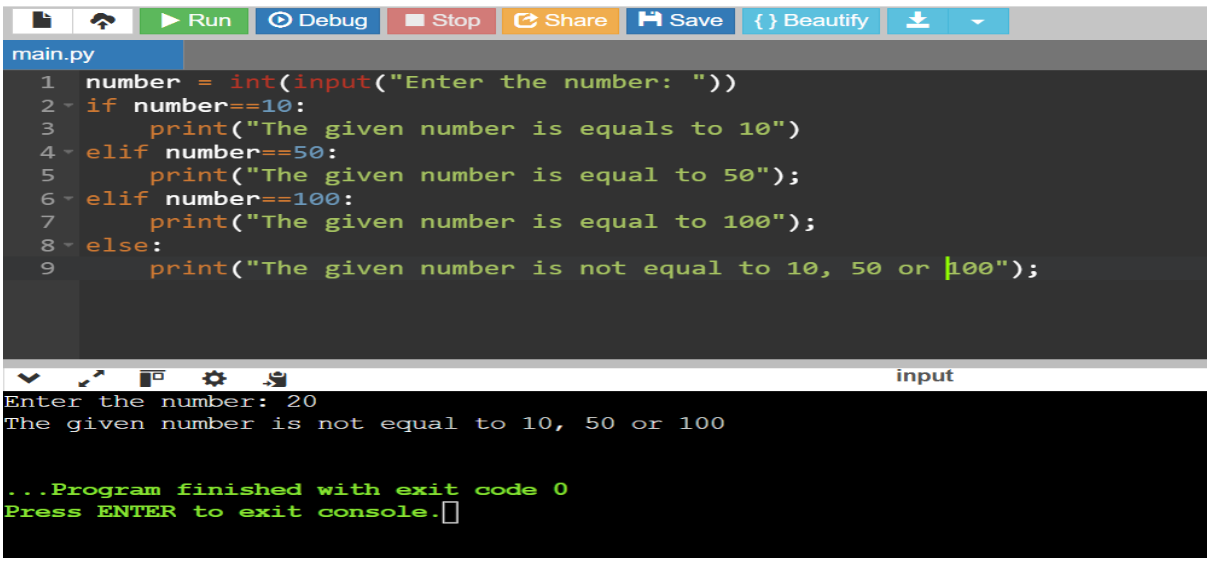
```
Enter your age: 22
You are eligible to vote !!

...Program finished with exit code 0
Press ENTER to exit console.
```



```python
age = int (input("Enter your age: "))
if age>=18:
     print("You are eligible to vote !!");
else:
     print("Sorry! you have to wait !!");
```

```
Enter your age: 16
Sorry! you have to wait !!

...Program finished with exit code 0
Press ENTER to exit console.
```

Python Elif Statement:



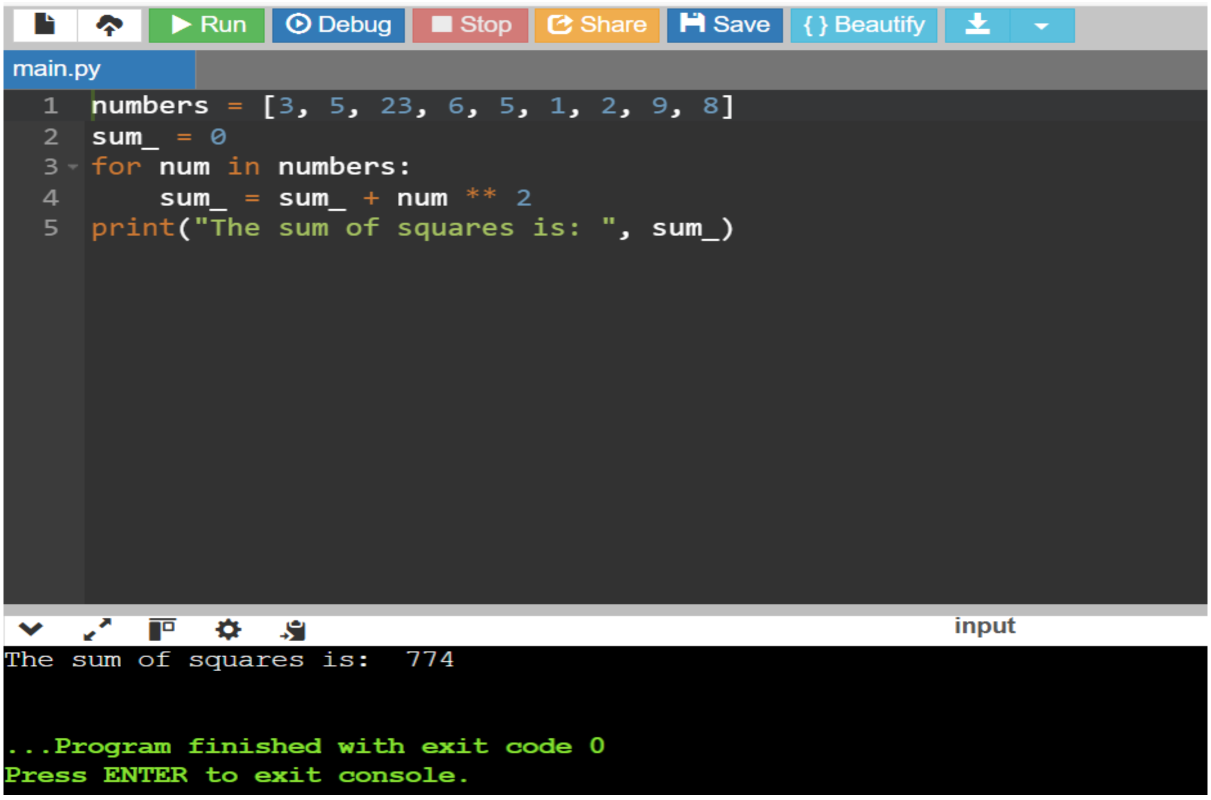
```python
number = int(input("Enter the number: "))
if number==10:
     print("The given number is equals to 10")
elif number==50:
     print("The given number is equal to 50");
elif number==100:
     print("The given number is equal to 100");
else:
     print("The given number is not equal to 10, 50 or 100");
```

```
Enter the number: 20
The given number is not equal to 10, 50 or 100

...Program finished with exit code 0
Press ENTER to exit console.
```

5.Python Loops:

5.1 For loop:

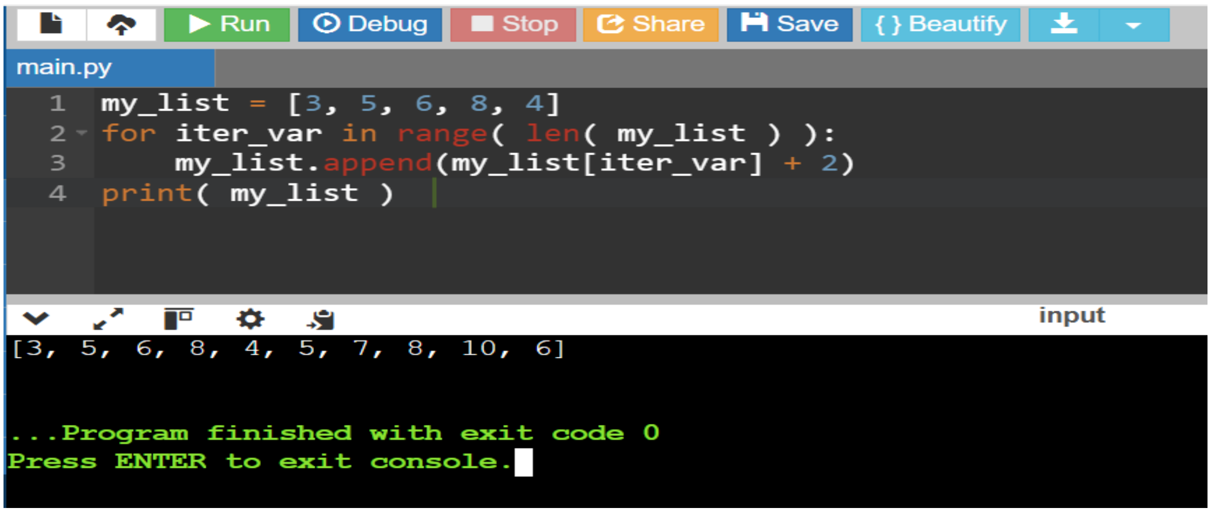
1. **Iterating by using index of sequence**



```python
numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8]
sum_ = 0
for num in numbers:
    sum_ = sum_ + num ** 2
print("The sum of squares is: ", sum_)
```

```
The sum of squares is:  774


...Program finished with exit code 0
Press ENTER to exit console.
```

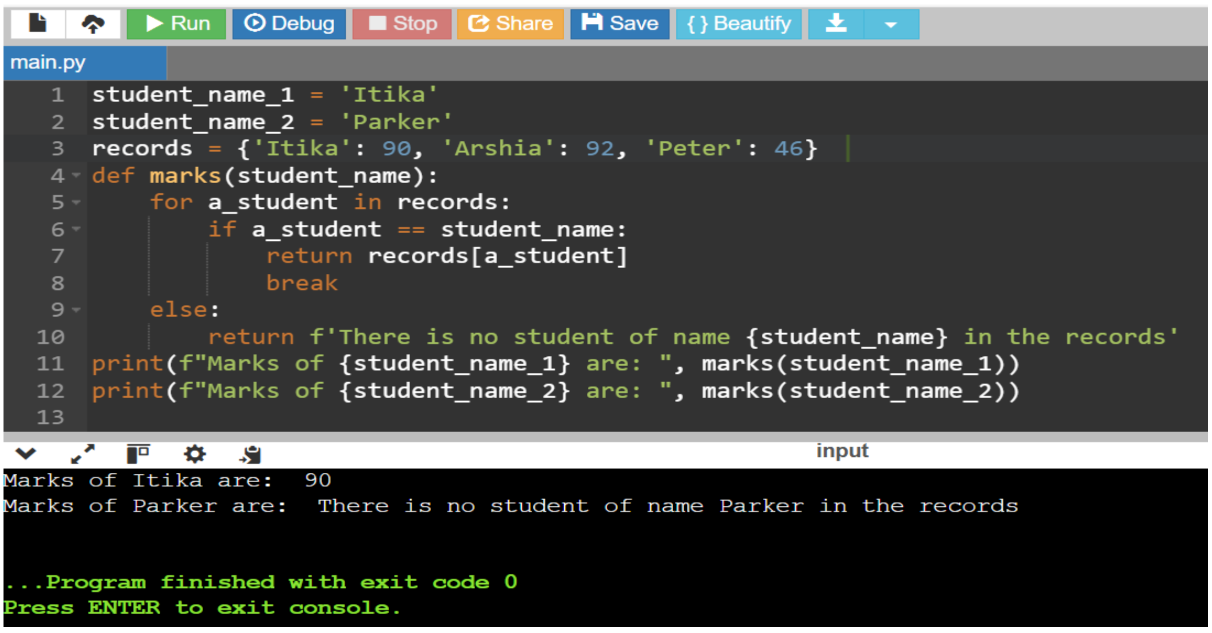
2. **Using Range ()**



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

```
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]


...Program finished with exit code 0
Press ENTER to exit console.
```

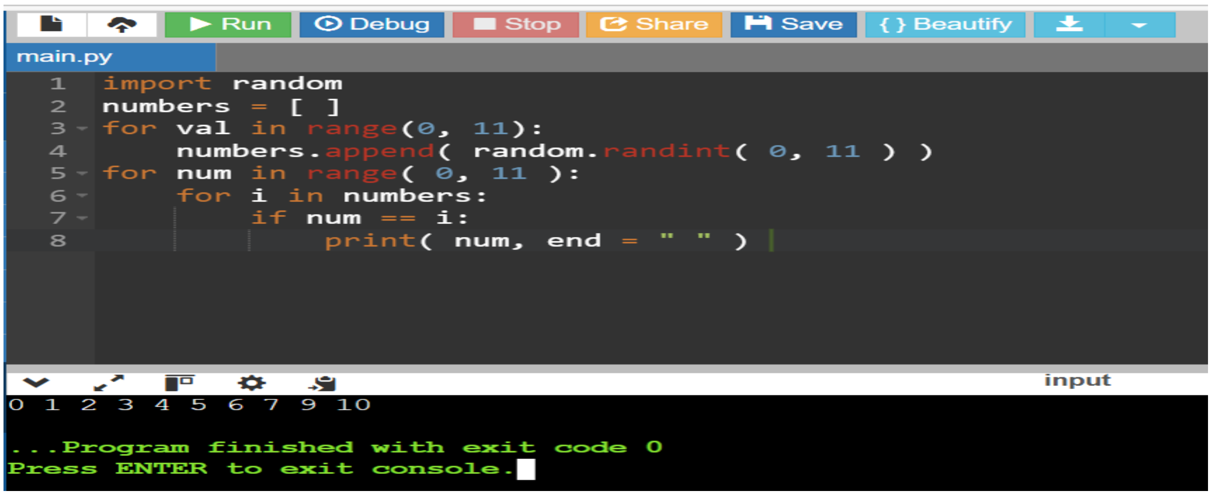
3. **Using else statement with For loop**



```python
student_name_1 = 'Itika'
student_name_2 = 'Parker'
records = {'Itika': 90, 'Arshia': 92, 'Peter': 46}
def marks(student_name):
    for a_student in records:
        if a_student == student_name:
            return records[a_student]
            break
    else:
        return f'There is no student of name {student_name} in the records'
print(f"Marks of {student_name_1} are: ", marks(student_name_1))
print(f"Marks of {student_name_2} are: ", marks(student_name_2))
```

```
Marks of Itika are:  90
Marks of Parker are:  There is no student of name Parker in the records


...Program finished with exit code 0
Press ENTER to exit console.
```

4. **Nested For loop**

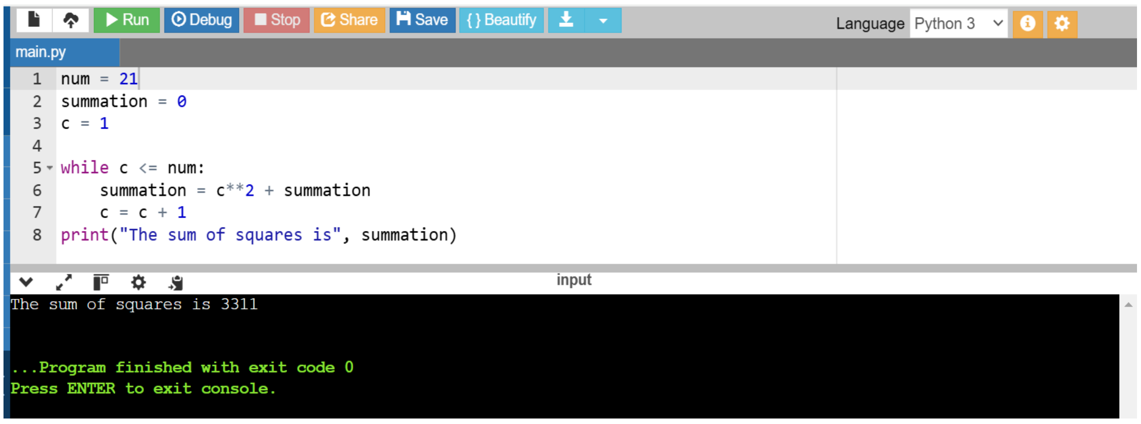


```python
import random
numbers = [ ]
for val in range(0, 11):
    numbers.append( random.randint( 0, 11 ) )
for num in range( 0, 11 ):
    for i in numbers:
        if num == i:
            print( num, end = " " )
```

```
0 1 2 3 4 5 6 7 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

5.2 While loop:

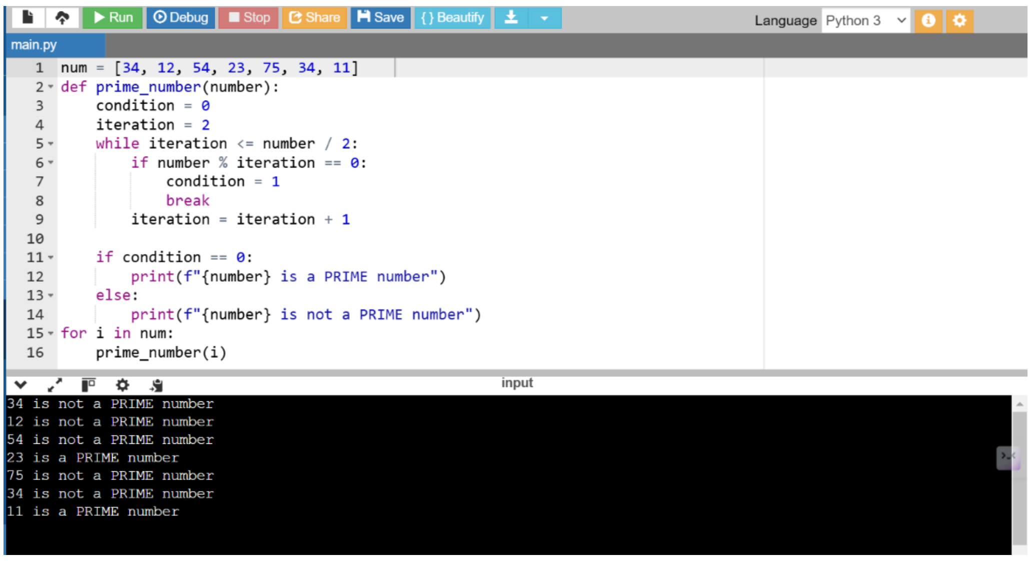
1. Sum of squares



```python
num = 21
summation = 0
c = 1

while c <= num:
    summation = c**2 + summation
    c = c + 1
print("The sum of squares is", summation)
```

```
The sum of squares is 3311


...Program finished with exit code 0
Press ENTER to exit console.
```

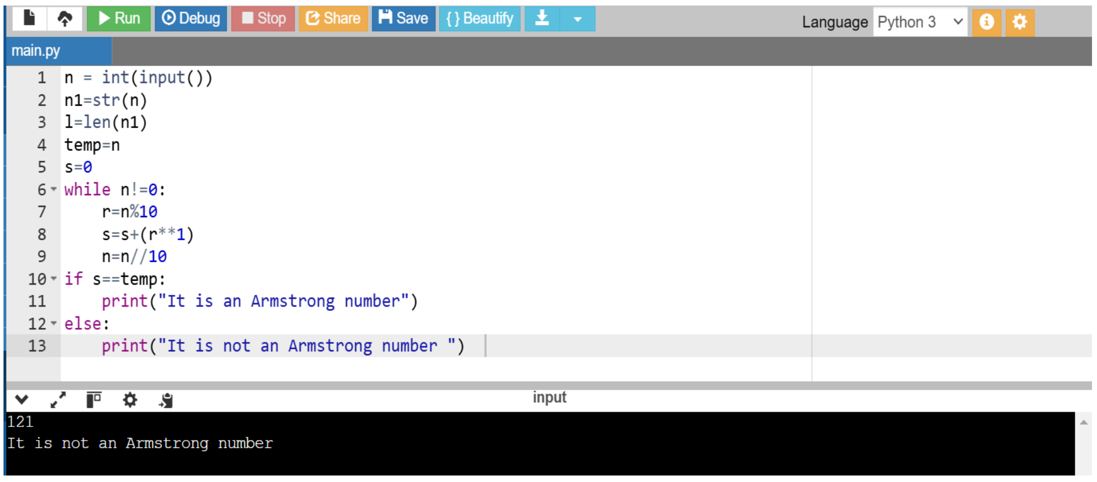
2. To check whether given number is Prime or not



```python
num = [34, 12, 54, 23, 75, 34, 11]
def prime_number(number):
    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)
```

```
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
```

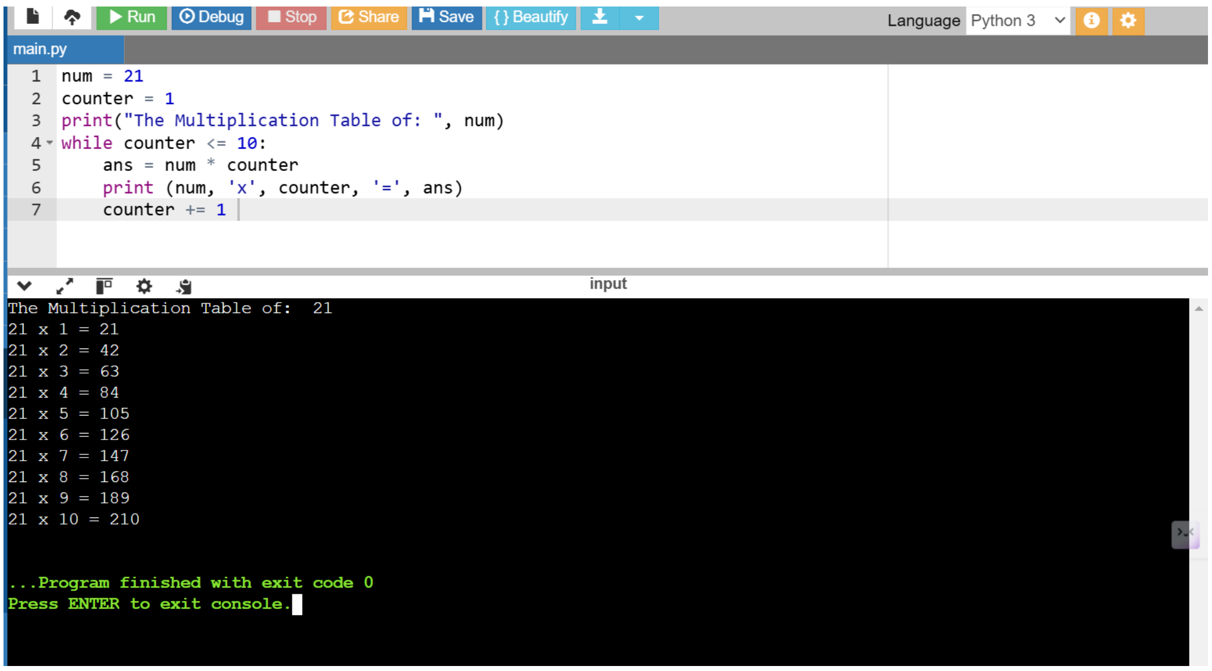
3. Armstrong number



```python
n = int(input())
n1=str(n)
l=len(n1)
temp=n
s=0
while n!=0:
    r=n%10
    s=s+(r**1)
    n=n//10
if s==temp:
    print("It is an Armstrong number")
else:
    print("It is not an Armstrong number ")
```

```
121
It is not an Armstrong number
```

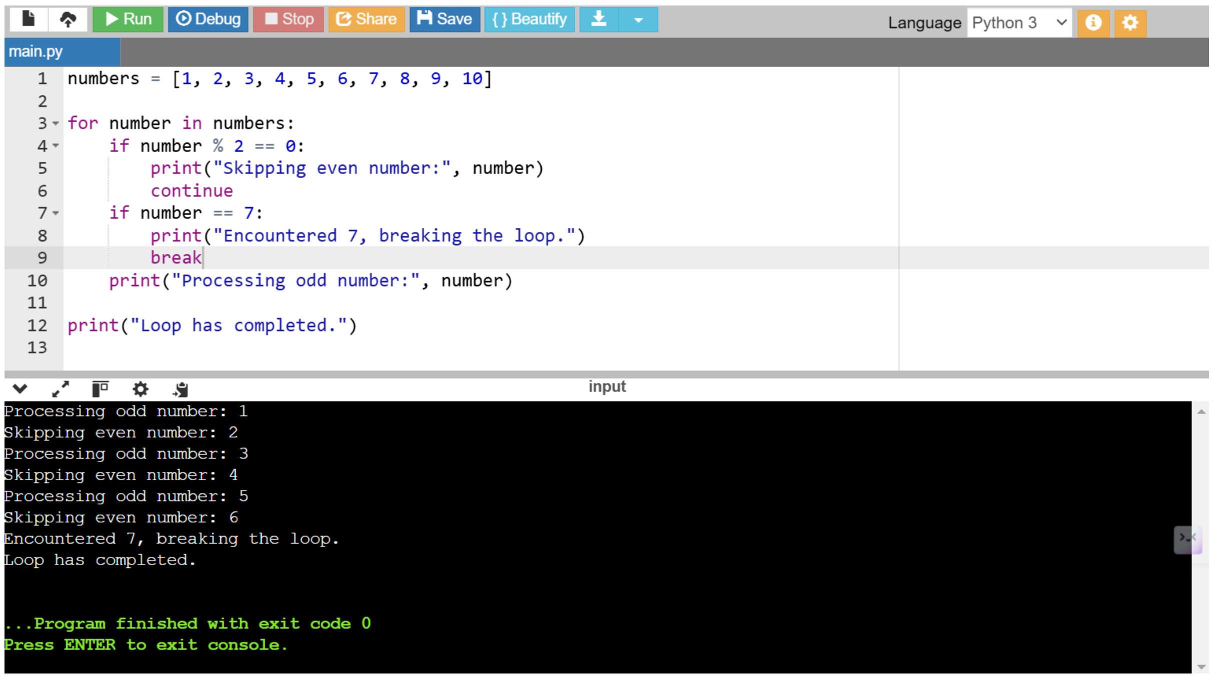
4. Multiplication Table



```python
num = 21
counter = 1
print("The Multiplication Table of: ", num)
while counter <= 10:
    ans = num * counter
    print (num, 'x', counter, '=', ans)
    counter += 1
```

```
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


...Program finished with exit code 0
Press ENTER to exit console.
```

5. BREAK Statement

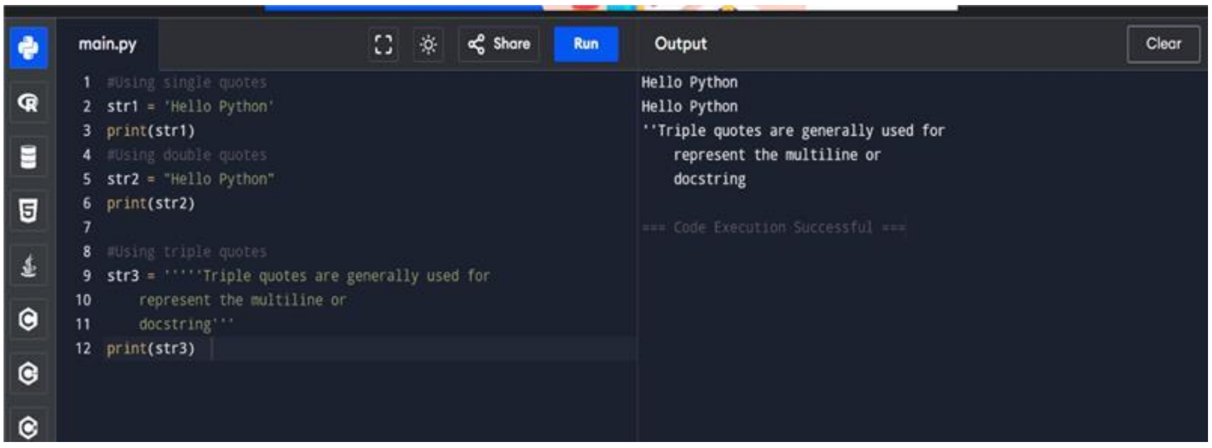


```python
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for number in numbers:
    if number % 2 == 0:
        print("Skipping even number:", number)
        continue
    if number == 7:
        print("Encountered 7, breaking the loop.")
        break
    print("Processing odd number:", number)

print("Loop has completed.")
```

```
Processing odd number: 1
Skipping even number: 2
Processing odd number: 3
Skipping even number: 4
Processing odd number: 5
Skipping even number: 6
Encountered 7, breaking the loop.
Loop has completed.


...Program finished with exit code 0
Press ENTER to exit console.
```

6. PYTHON STRINGS

1. Creating String in Python

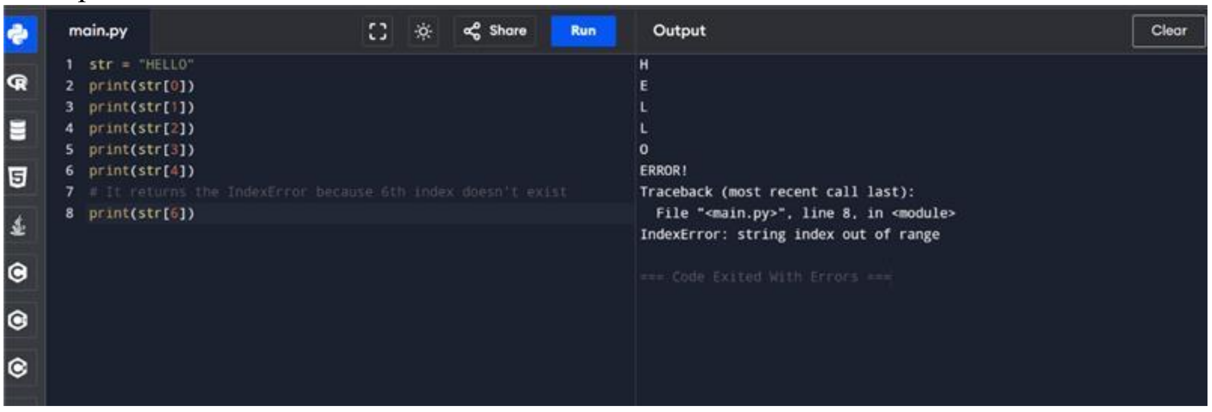


```python
#Using single quotes
str1 = 'Hello Python'
print(str1)
#Using double quotes
str2 = "Hello Python"
print(str2)

#Using triple quotes
str3 = '''''Triple quotes are generally used for
    represent the multiline or
    docstring'''
print(str3)
```

```
Hello Python
Hello Python
''Triple quotes are generally used for
    represent the multiline or
    docstring

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

2. Strings indexing and splitting

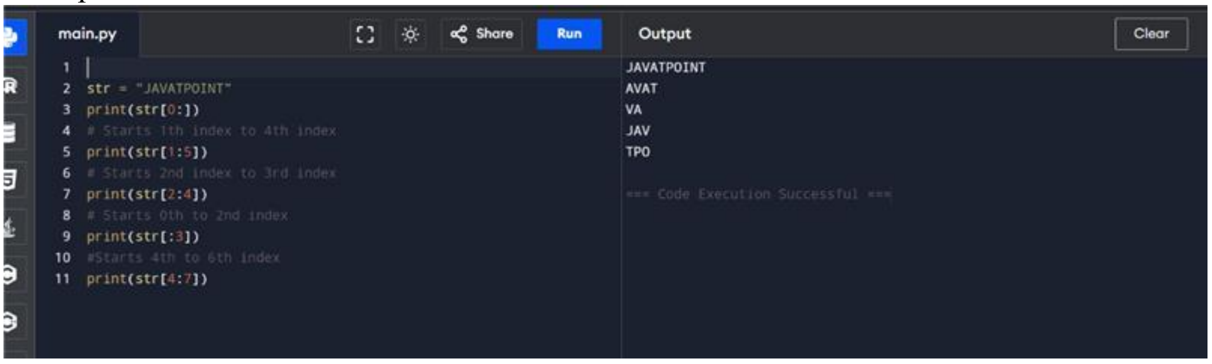
Example 1:



```python
str = "HELLO"
print(str[0])
print(str[1])
print(str[2])
print(str[3])
print(str[4])
# It returns the IndexError because 6th index doesn't exist
print(str[6])
```

```
H
E
L
L
O
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 8, in <module>
IndexError: string index out of range

=== Code Exited With Errors ===
```

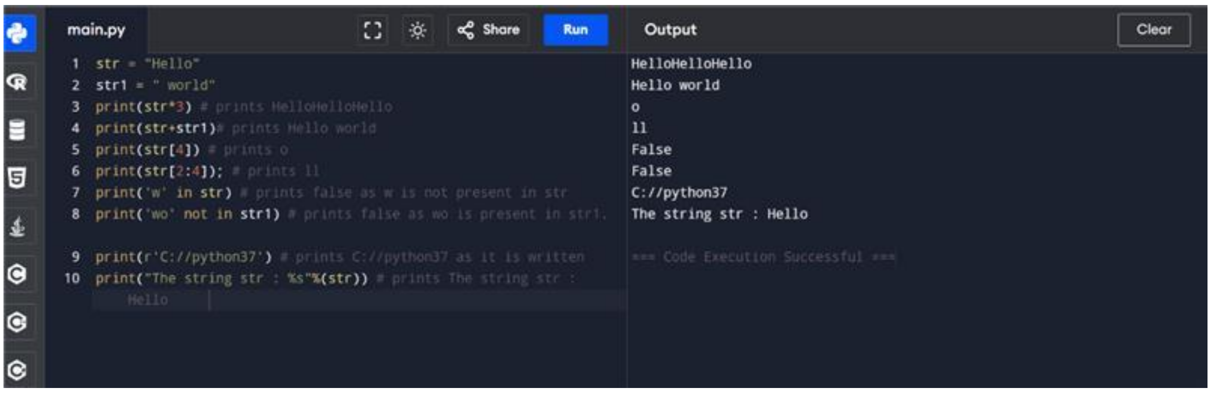
Example 2



```python

str = "JAVATPOINT"
print(str[0:])
# Starts 1th index to 4th index
print(str[1:5])
# Starts 2nd index to 3rd index
print(str[2:4])
# Starts 0th to 2nd index
print(str[:3])
#Starts 4th to 6th index
print(str[4:7])
```

```
JAVATPOINT
AVAT
VA
JAV
TPO

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

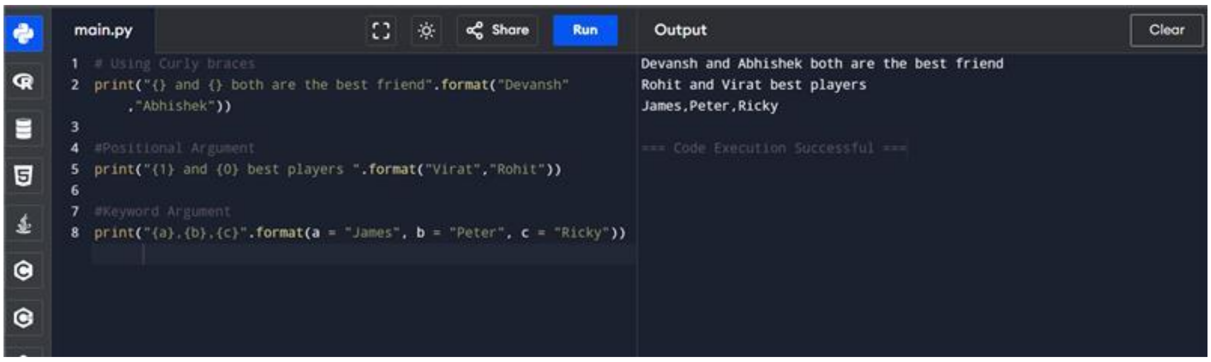
3. Strings Operators



```
str = "Hello"
str1 = " world"
print(str*3) # prints HelloHelloHello
print(str+str1)# prints Hello world
print(str[4]) # prints o
print(str[2:4]); # prints ll
print('w' in str) # prints false as w is not present in str
print('wo' not in str1) # prints false as wo is present in str1.

print(r'C://python37') # prints C://python37 as it is written
print("The string str : %s"%(str)) # prints The string str :
    Hello
```

```
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello

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

4. String formatting

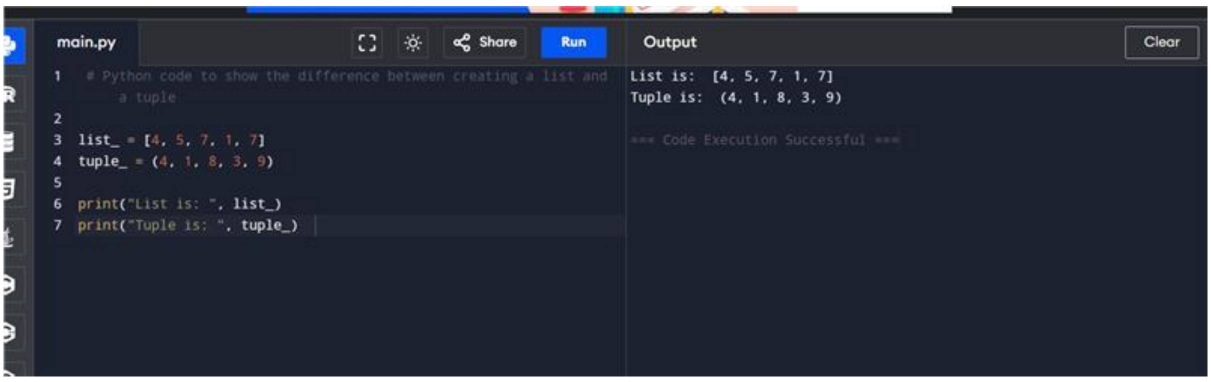


```
# Using Curly braces
print("{} and {} both are the best friend".format("Devansh"
    ,"Abhishek"))

#Positional Argument
print("{1} and {0} best players ".format("Virat","Rohit"))

#Keyword Argument
print("{a},{b},{c}".format(a = "James", b = "Peter", c = "Ricky"))
```

```
Devansh and Abhishek both are the best friend
Rohit and Virat best players
James,Peter,Ricky

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

7.Lists and Tuples

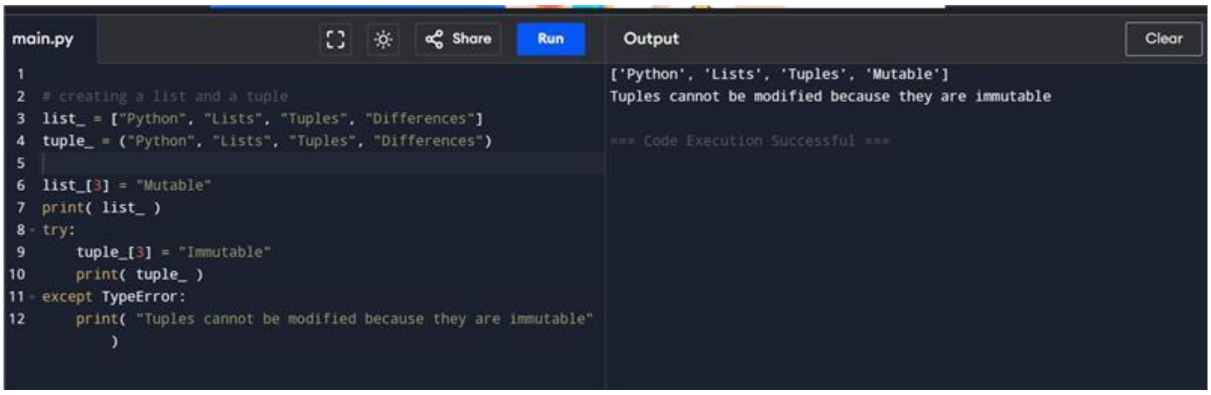
1. **List and Tuple Syntax Differences**



```python
# Python code to show the difference between creating a list and a tuple

list_ = [4, 5, 7, 1, 7]
tuple_ = (4, 1, 8, 3, 9)

print("List is: ", list_)
print("Tuple is: ", tuple_)
```

```
List is:  [4, 5, 7, 1, 7]
Tuple is:  (4, 1, 8, 3, 9)

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

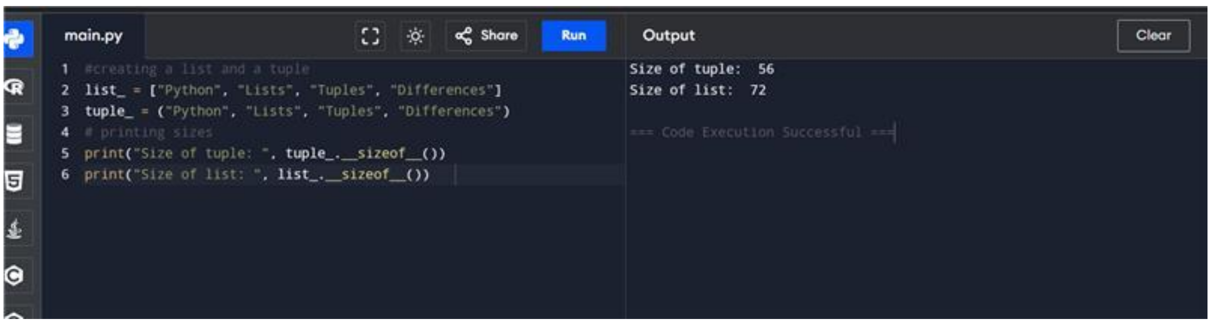
2. **Updating the element of list and tuple at a particular index**



```python

# creating a list and a tuple
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")

list_[3] = "Mutable"
print( list_ )
try:
    tuple_[3] = "Immutable"
    print( tuple_ )
except TypeError:
    print( "Tuples cannot be modified because they are immutable" )
```

```
['Python', 'Lists', 'Tuples', 'Mutable']
Tuples cannot be modified because they are immutable

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

3. **Code to show the difference in the size of a list and a tuple**



```python
#creating a list and a tuple
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")
# printing sizes
print("Size of tuple: ", tuple_.__sizeof__())
print("Size of list: ", list_.__sizeof__())
```

```
Size of tuple:  56
Size of list:  72

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