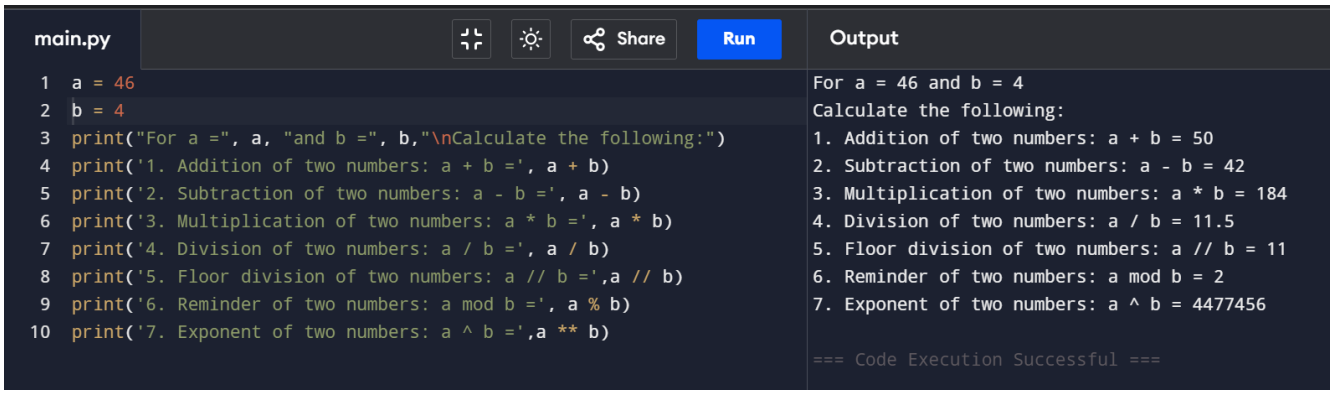
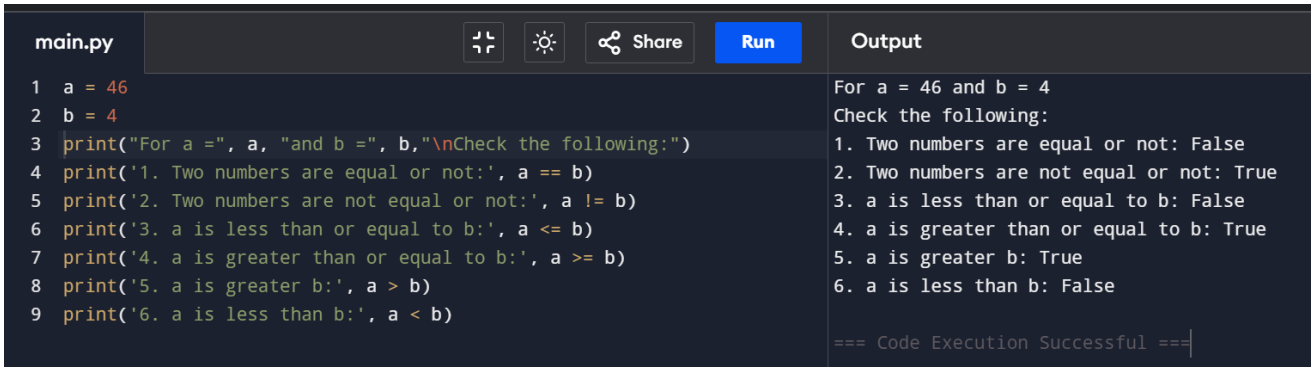
Python Assignment

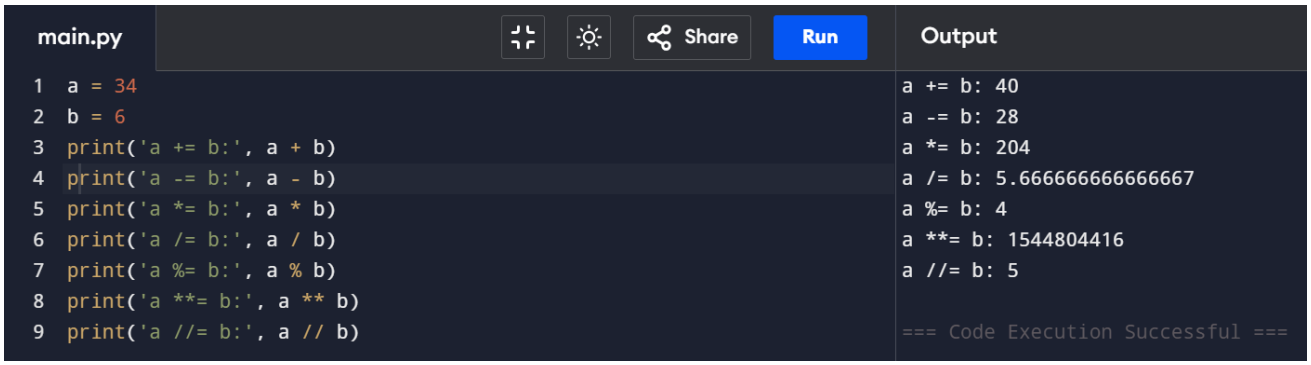
Arithmetic operations between 2 numbers



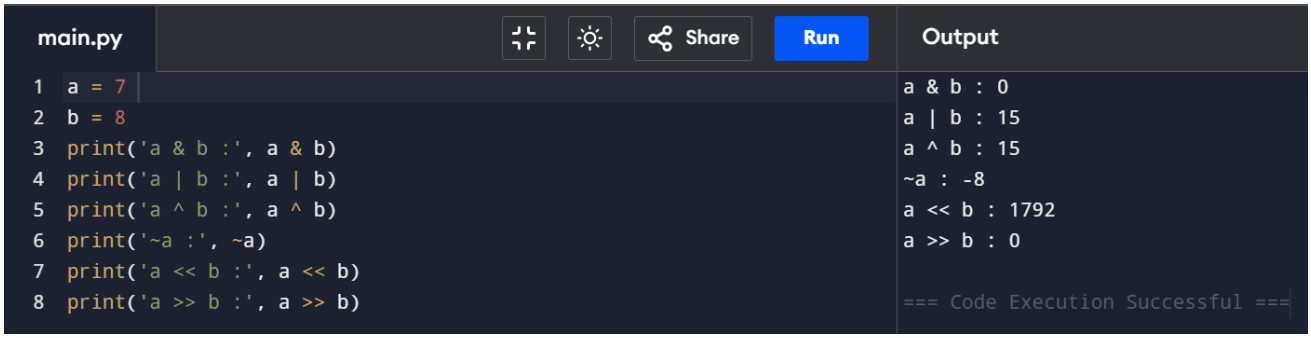
Comparison operations between 2 numbers



Assignment operators

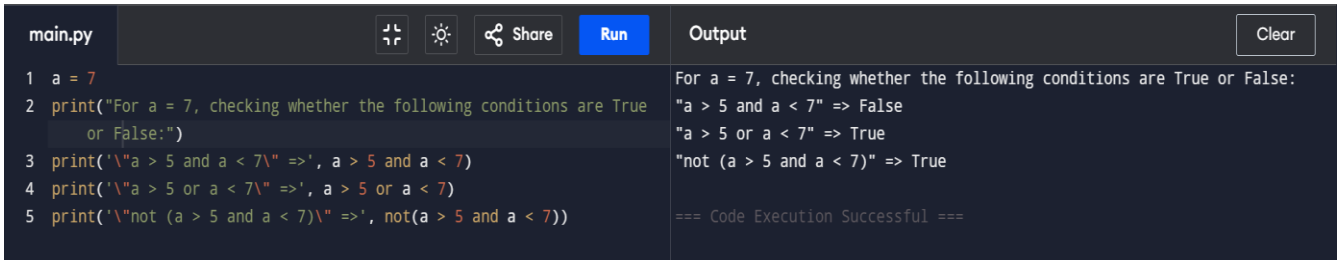


Bitwise operation between 2 numbers



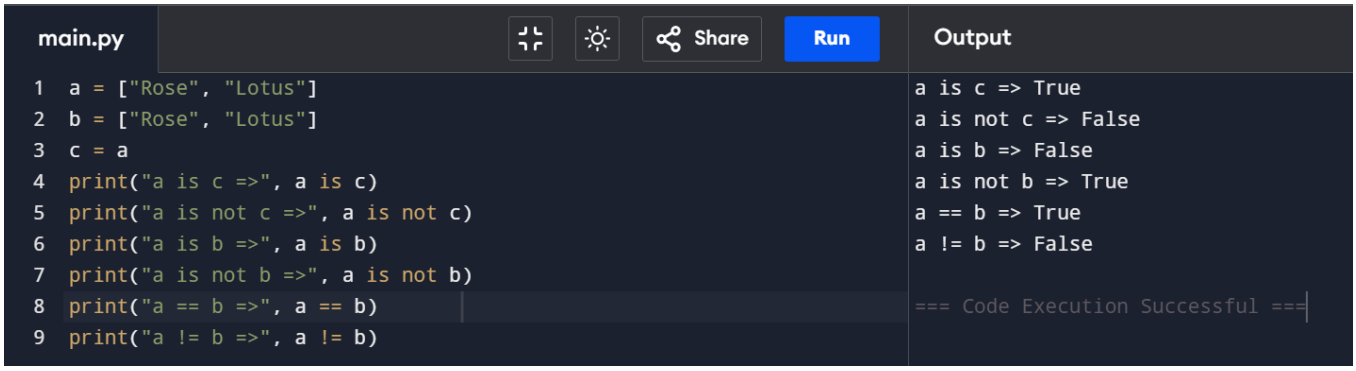
Python Assignment

Logical operators

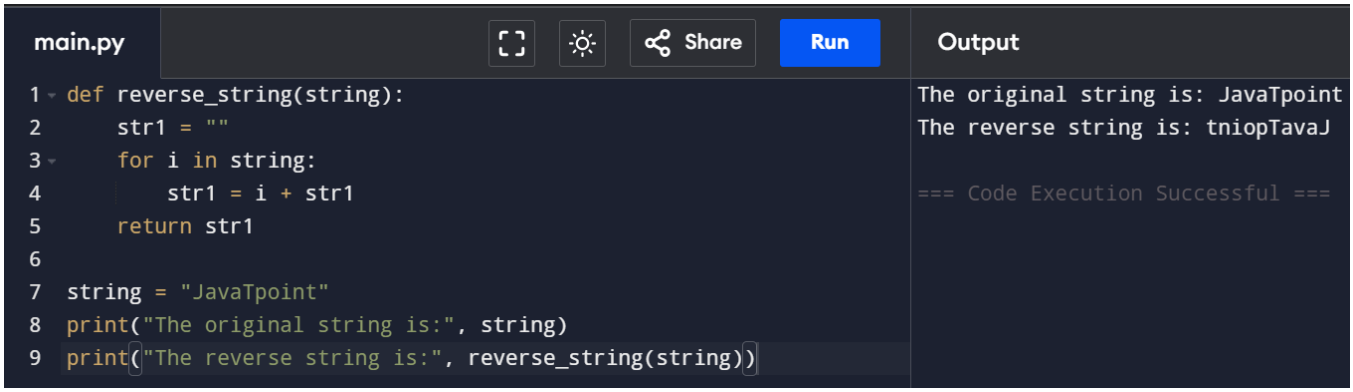


Membership operators

Identity operators



Reverse a string using for loop



Python Assignment

Reverse a string using while loop

main.py

```
str = "JavaTpoint"
print("The original string is:", str)

reverse_String = ""
count = len(str)

while count > 0:
    reverse_String += str[count - 1]
    count = count - 1

print("The reversed string using a while loop is:",
    reverse_String)
```

Output

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

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

Reverse a string using slice operator

main.py

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

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

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

Reverse function using join

main.py

```
def reverse(s):
    string = "".join(reversed(s))
    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

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

Reverse using recursion

main.py

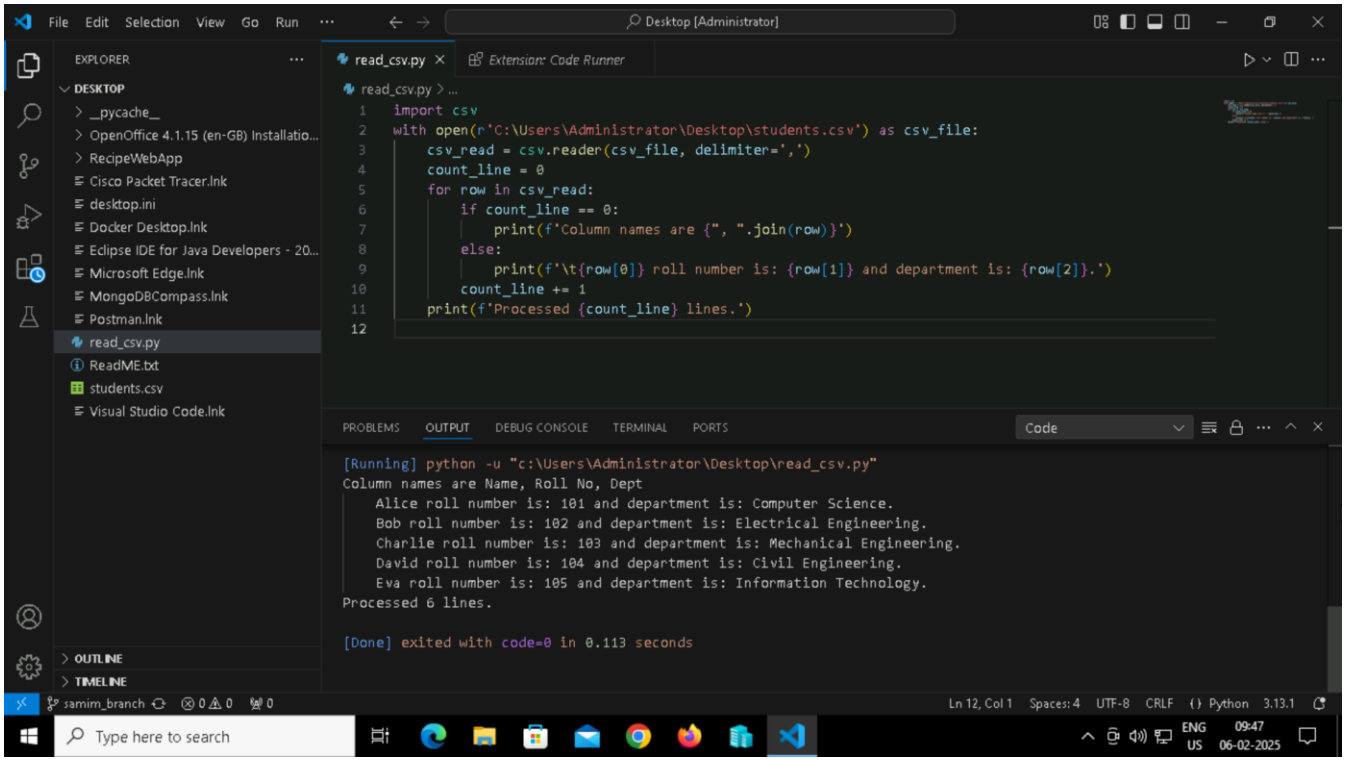
```
def reverse(s):
    string = "".join(reversed(s))
    return string

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

Output

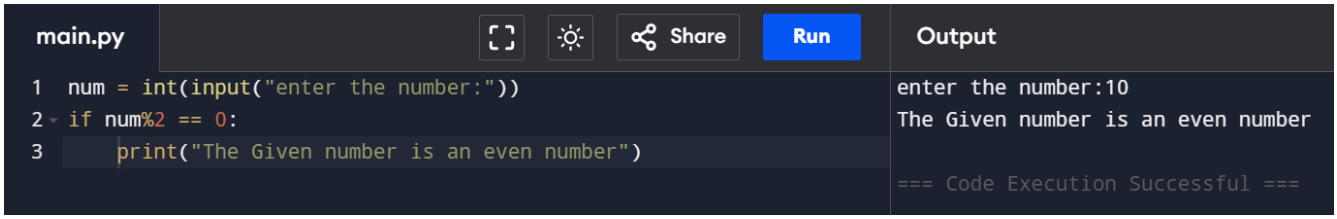
```
The original string is: JavaTpoint
The reversed string using reversed() is: tniopTavaJ

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

Python Assignment

Read a csv file in python

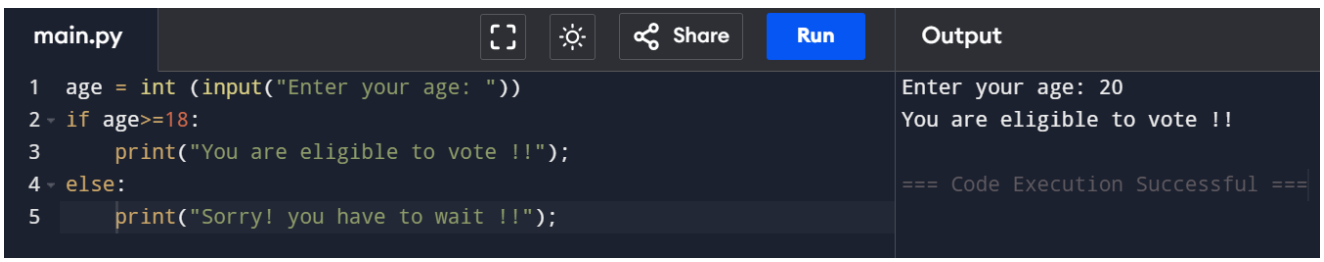


Simple python program to understand if statement



Program to print largest of 3 numbers

Program to check whether a person is eligible to vote or not



Python Assignment

Program to check if the given number is even or odd

main.py

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

Output

```
enter the number:9
The Given Number is an odd number

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

Python program 1 elif statement

main.py

```
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")
```

Output

```
Enter the number:10
The given number is equals to 10

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

Python program 2 elif statement

main.py

```
marks = int(input("Enter the marks:"))
if marks > 85 and marks <= 100:
    print("Congrats ! you scored grade A ...")
elif marks > 60 and marks <= 85:
    print("You scored grade B + ...")
elif marks > 40 and marks <= 60:
    print("You scored grade B ...")
elif marks > 30 and marks <= 40:
    print("You scored grade C ...")
else:
    print("Sorry you failed")
```

Output

```
Enter the marks:60
You scored grade B ...

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

List of squares using for loop

main.py

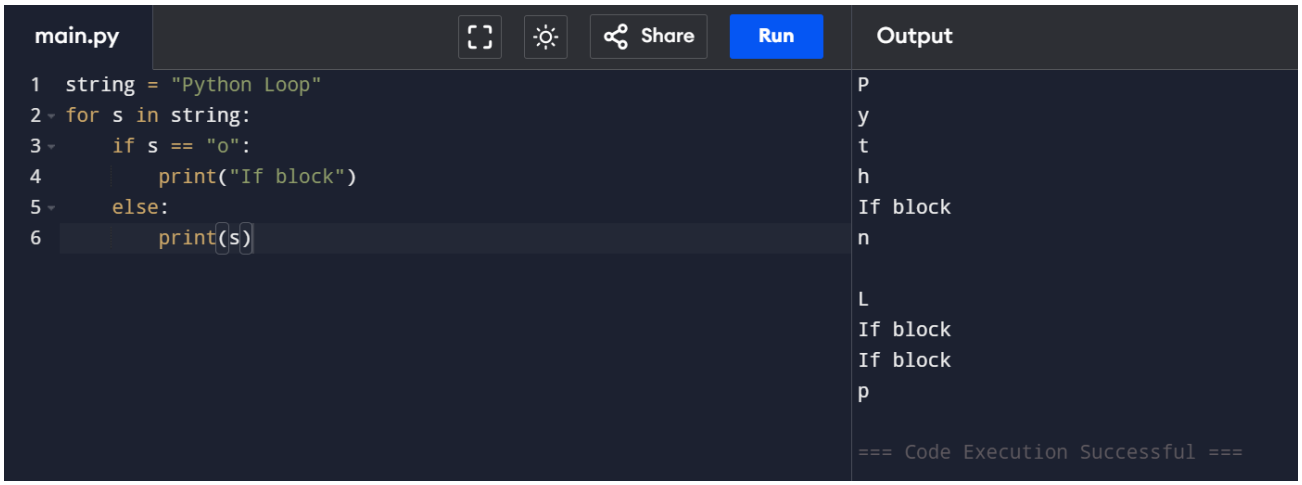
```
numbers = [4, 2, 6, 7, 3, 5, 8, 10, 6, 1, 9, 2]
square = 0
squares = []
for value in numbers:
    square = value ** 2
    squares.append(square)
print("The list of squares is", squares)
```

Output

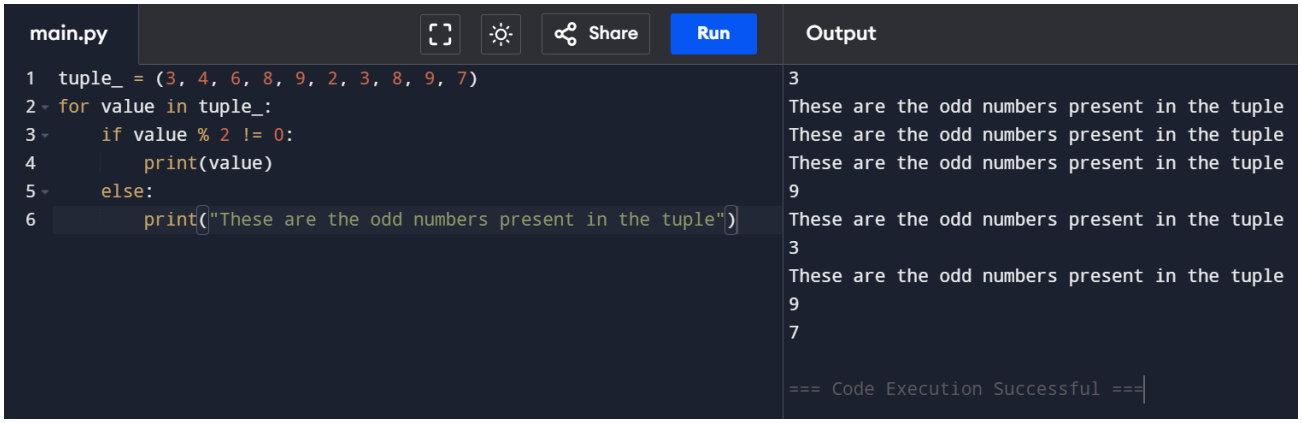
```
The list of squares is [16, 4, 36, 49, 9, 25, 64, 100, 36, 1, 81, 4]

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

Python Assignment

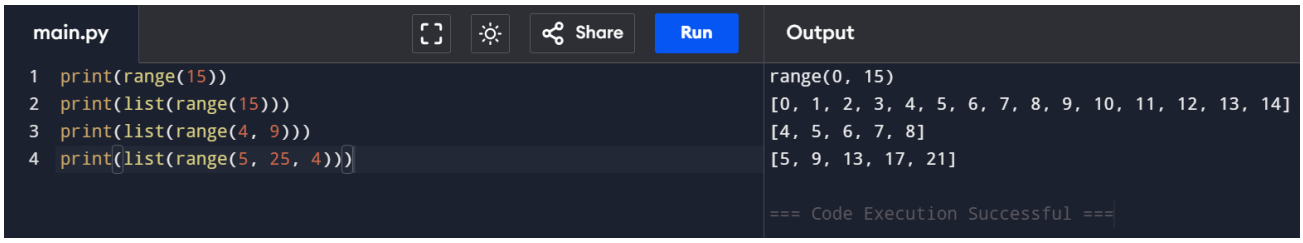
Using loop in string manipulation



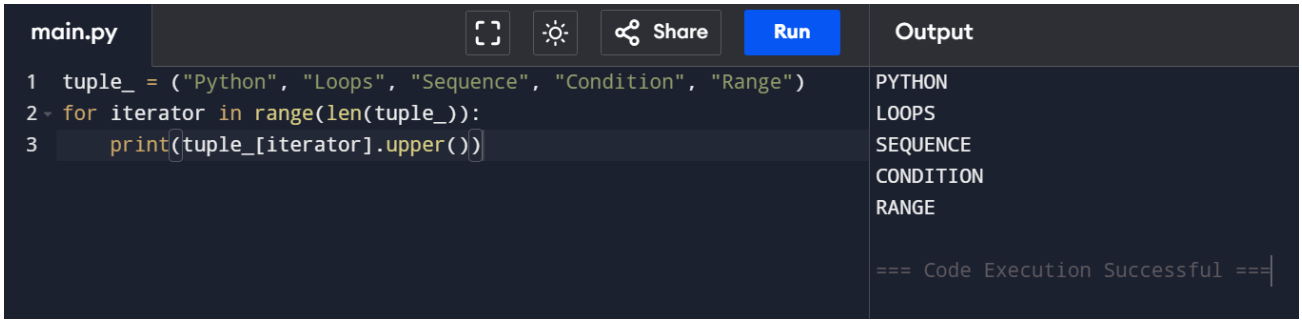
Using else statement with for loop



Range function

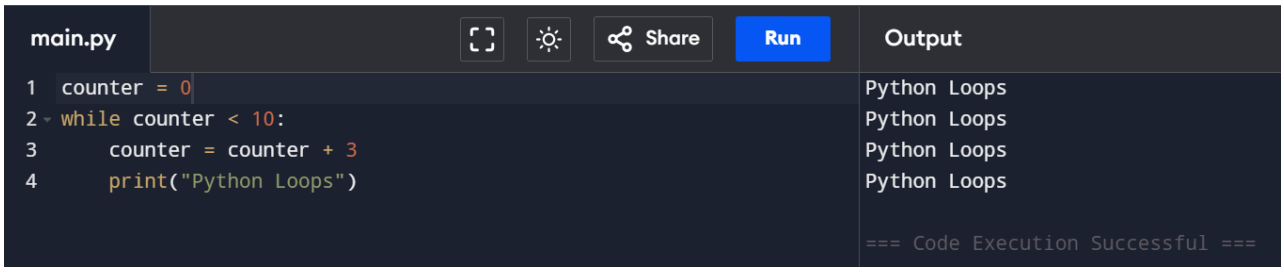


Python program to iterate over a sequence using with the help of indexing



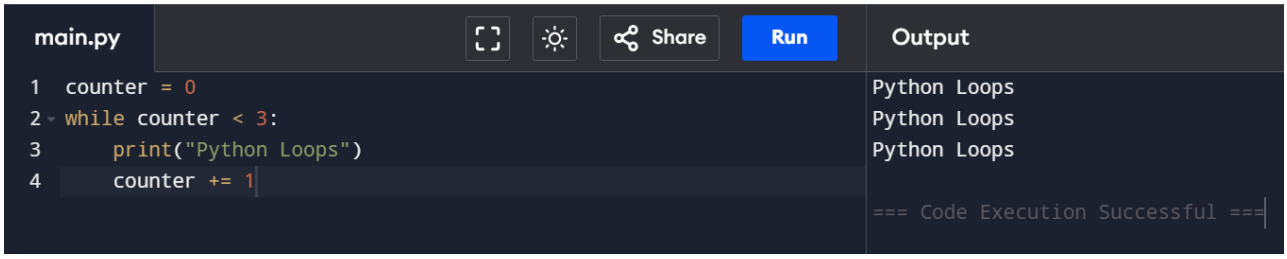
Python Assignment

While loop

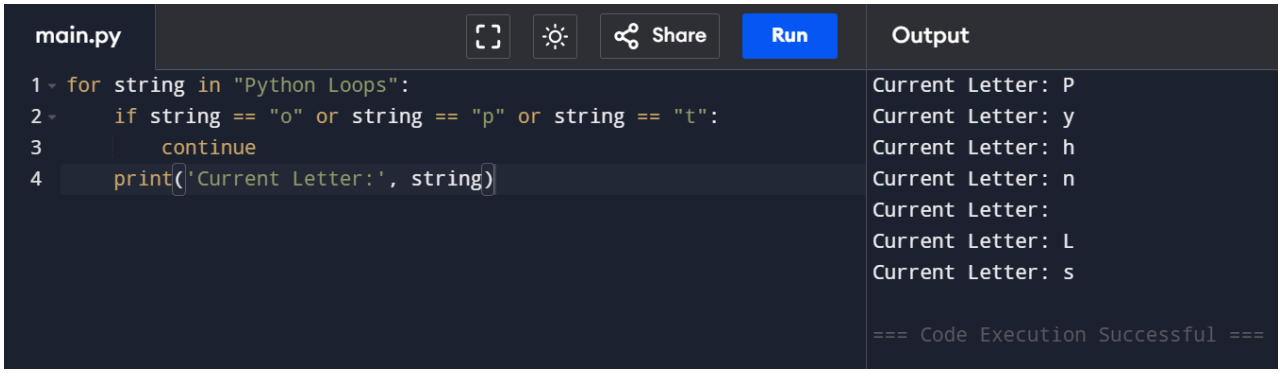


Else statement inside while loop

Single statement while block



Continue statement



Python Assignment

Break statement

main.py

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

Output

```
Current Letter:  P
Current Letter:  y
Current Letter:  t
Current Letter:  h
Current Letter:  o
Current Letter:  n
Current Letter:

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

Pass statement

main.py

```
for string in "Python Loops":
    pass
print('Last Letter:', string)
```

Output

```
Last Letter: s

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

Code to find the sum of squares of each element of the list using for loop

main.py

```
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_)
```

Output

```
The sum of squares is:  774

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

Code to find the sum of squares of each element of the list using for loop

main.py

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

Output

```
The sum of squares is:  774

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

code to print marks of a student from the record

main.py

```
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]
    return f'There is no student of name {student_name} in the
        record'
print(f"Marks of {student_name_1} are: ", marks(student_name_1))
print(f"Marks of {student_name_2} are: ", marks(student_name_2))
```

Output

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

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

Python Assignment

Nested loops

main.py

```
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=" ")
```

Output

```
0 2 4 4 5 8 8 8 9 9
=== Code Execution Successful ===
```

while loops in Python for printing numbers from 1 to 10

main.py

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

Output

```
1 2 3 4 5 6 7 8 9 10
=== Code Execution Successful ===
```

while loops in Python for Printing those numbers divisible by either 5 or 7 within 1 to 50

main.py

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

Output

```
5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50
=== Code Execution Successful ===
```

the sum of squares of the first 15 natural numbers using a while loop.

main.py

```
num = 15
summation = 0
c = 1
while c <= num:
    summation = c**2 + summation
    c = c + 1
print("The sum of squares is", summation)
```

Output

```
The sum of squares is 1240

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

Python Assignment

while loops in Python for a number is Prime number or not.

main.py

```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)
```

Output

```
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

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

while loops in Python for a number is Armstrong number or not.

main.py

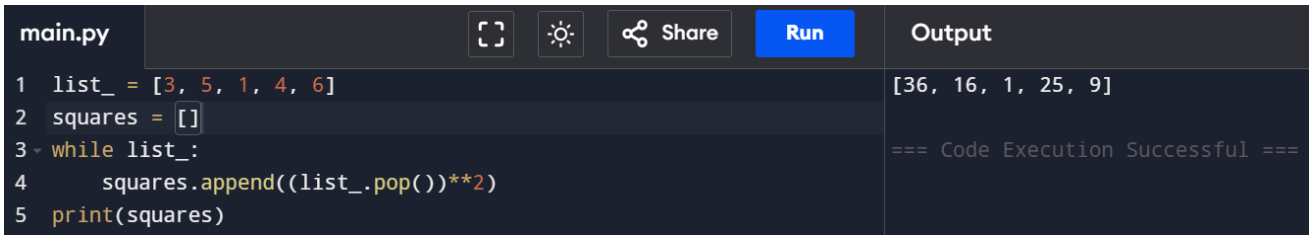
```python
n = int(input())
n1 = str(n)
l = len(n1)
temp = n
s = 0
while n != 0:
    r = n % 10
    s = s + (r ** l)
    n = n // 10
if s == temp:
    print("It is an Armstrong number")
else:
    print("It is not an Armstrong number")
```

```
153
It is an Armstrong number


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

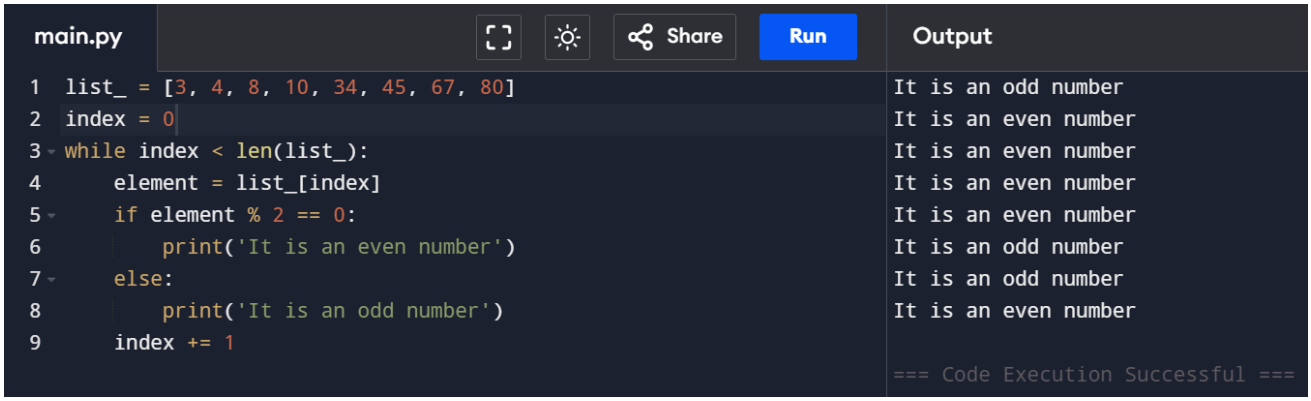
Python Assignment

while loop for printing the multiplication table of a given number.

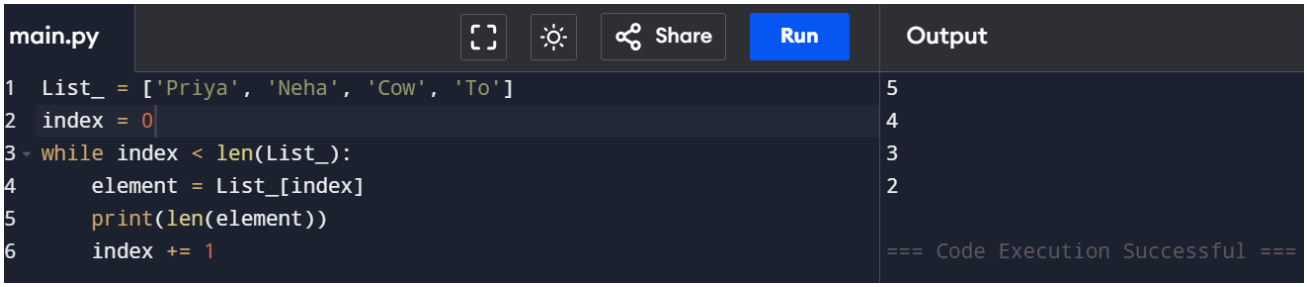
while loops in Python for square every number of a list



while loops in Python for determine odd and even number from every number of a list

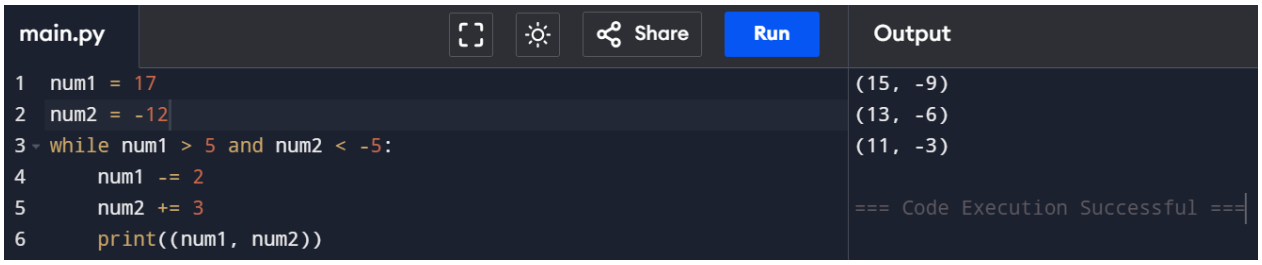


while loops in Python for determine the number letters of every word from the given list.

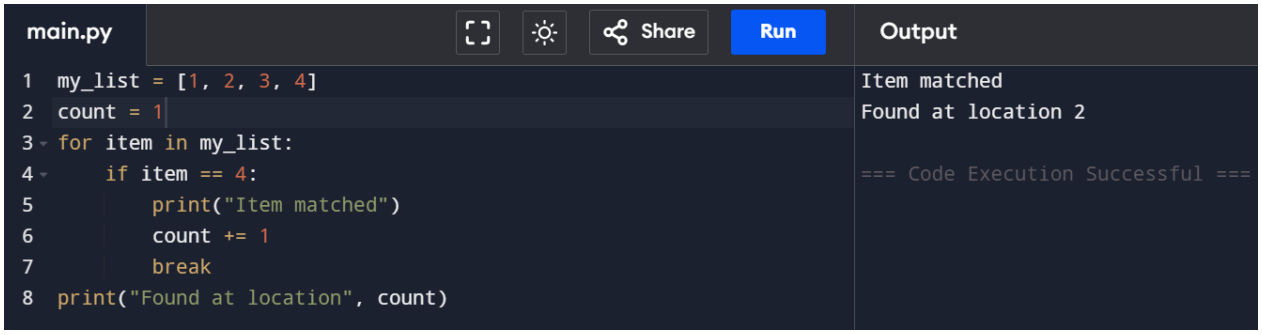


Python Assignment

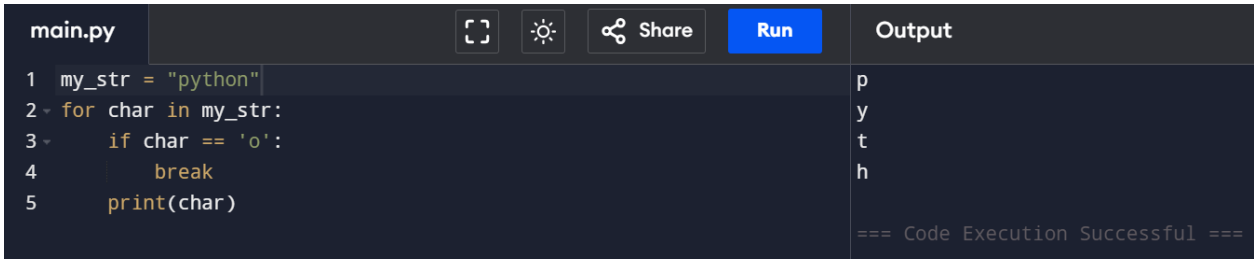
while loops in Python for multiple condition.



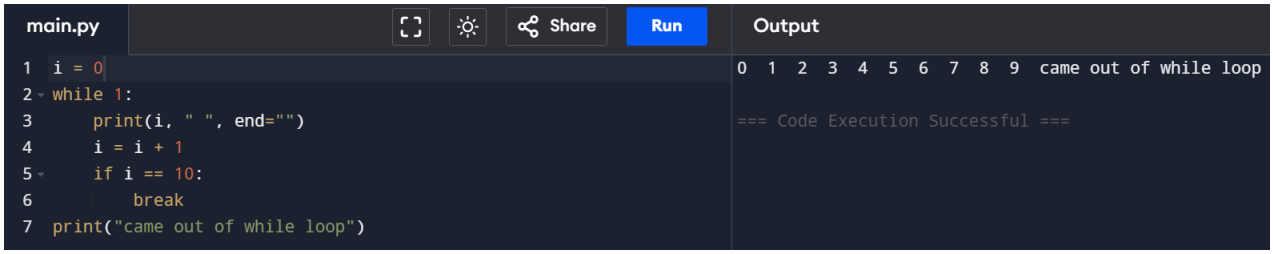
break statement with for loop



Breaking out of a loop early

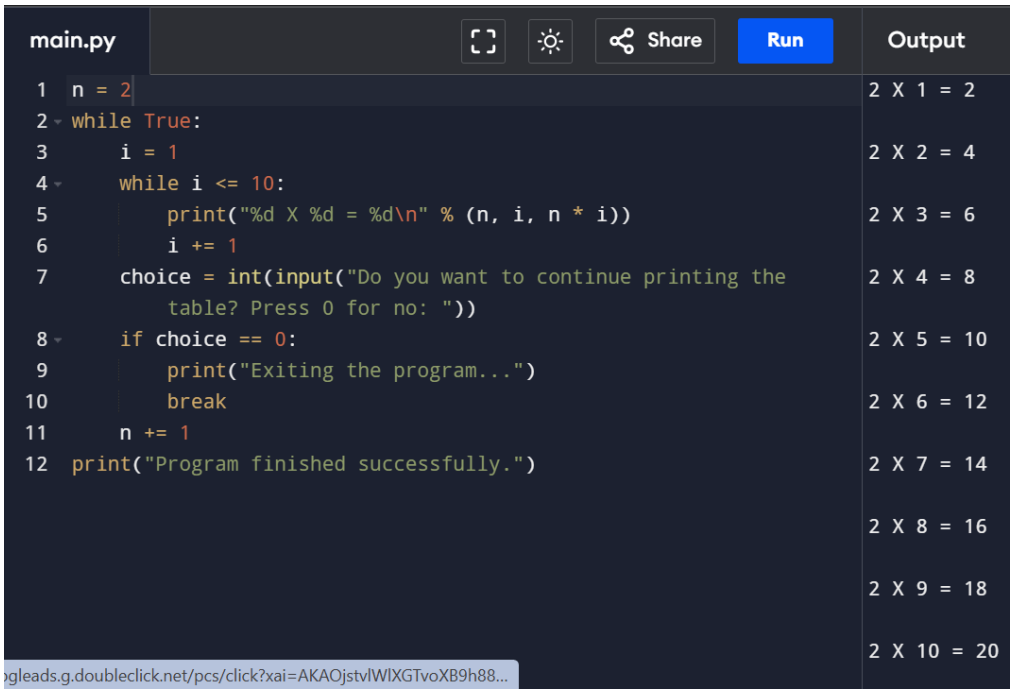


break statement with while loop



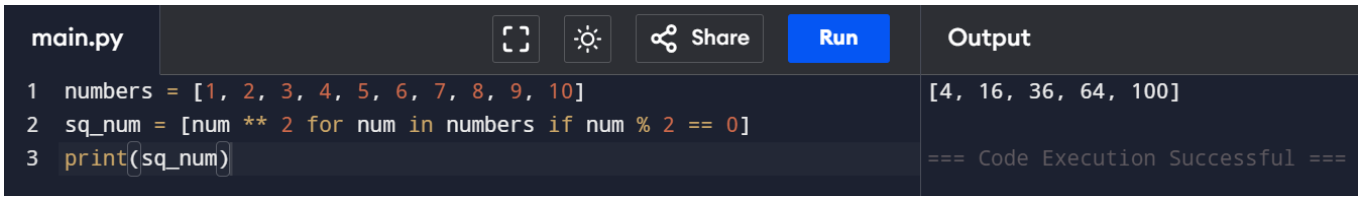
Python Assignment

break statement with nested loops

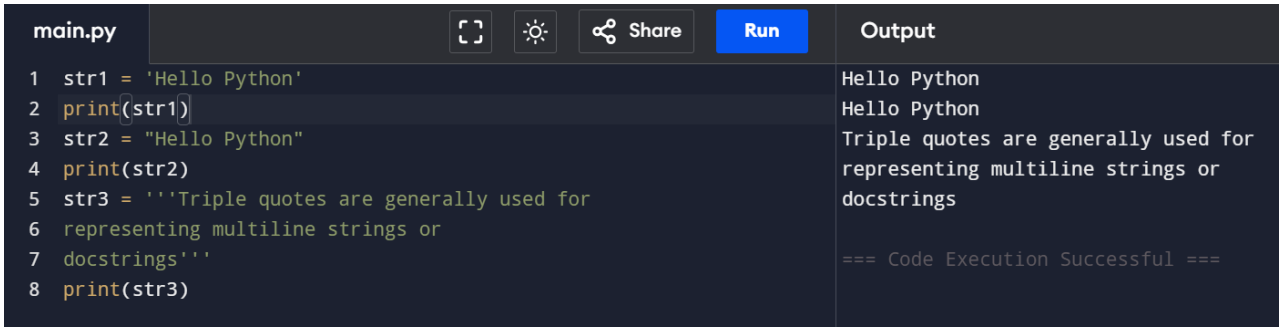


Python Continue Statements in while Loop

Python Continue statement in list comprehension



Creating String in Python



Python Assignment

Strings indexing and splitting

Example 1

main.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

```
H
E
L
L
O
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 7, in <module>
IndexError: string index out of range

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

Example 2

main.py

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

Output

```
JAVATPOINT
AVAT
VA
JAV
TPO

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

Reverse

main.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  print(str[::-1])
8  print(str[-12])
9
```

Output

```
T
I
NT
OIN
ATPOI
TNIOPTAVAJ
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 8, in <module>
IndexError: string index out of range

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

Reassigning Strings

main.py

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

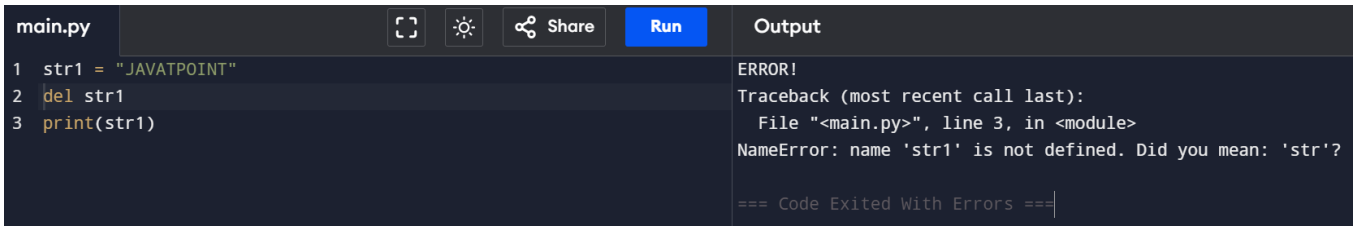
Output

```
HELLO
hello

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

Deleting the String

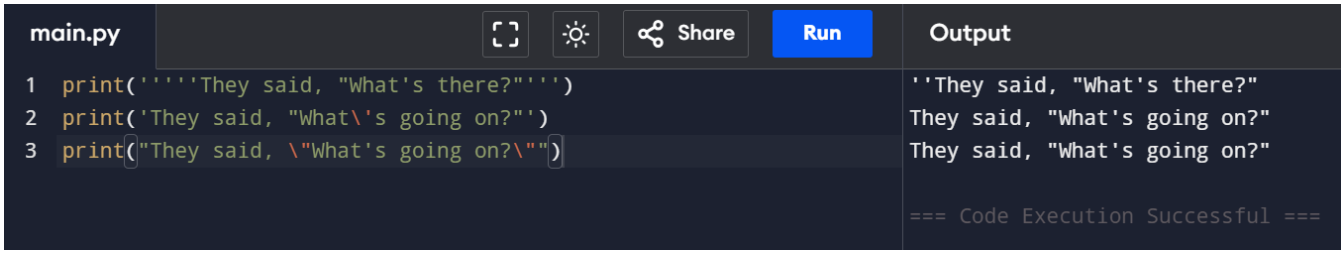
Python Assignment

Deleting a string



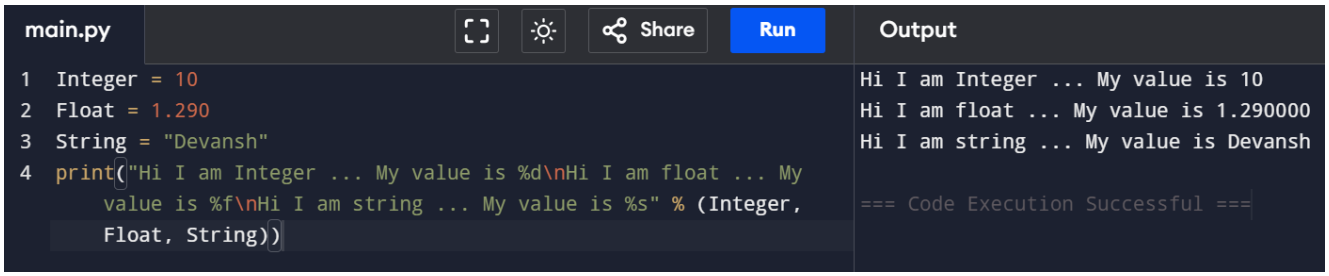
use of Python operators

Escape Sequence



The format() method

Python String Formatting Using % Operator



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

Python Assignment

main.py

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

Output

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

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

Updating the element of list and tuple at a particular index

main.py

```
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")
```

Output

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

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

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

main.py

```
list_ = ["Python", "Lists", "Tuples", "Differences"]
tuple_ = ("Python", "Lists", "Tuples", "Differences")
print("Size of tuple: ", tuple_.__sizeof__())
print("Size of list: ", list_.__sizeof__())
```

Output

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

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