Python Flow Control

Python if...else

Python for Loop

Python while Loop

Python break and continue

Python Pass

Python if...else

- What are if statement in Python?
  - Python if Statement Syntax
  - Python if Statement Flowchart
  - Example: Python if Statement
- Python if...else Statement
  - Syntax of if...else
  - Python if..else Flowchart
  - Example of if...else
- Python if...elif...else Statement
  - Syntax of if...elif...else
  - Flowchart of if...elif...else
  - Example of if...elif...else
- Python Nested if statements

Python if Statement Syntax

- Syntax

```
if test expression:
    statement(s)
```

Python if Statement Flowchart

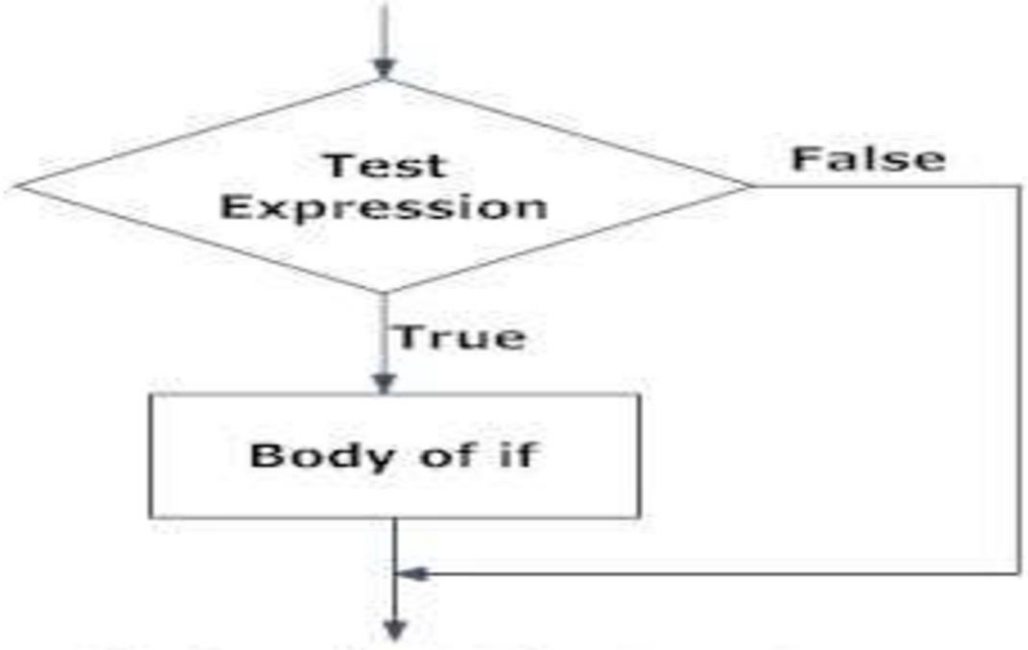
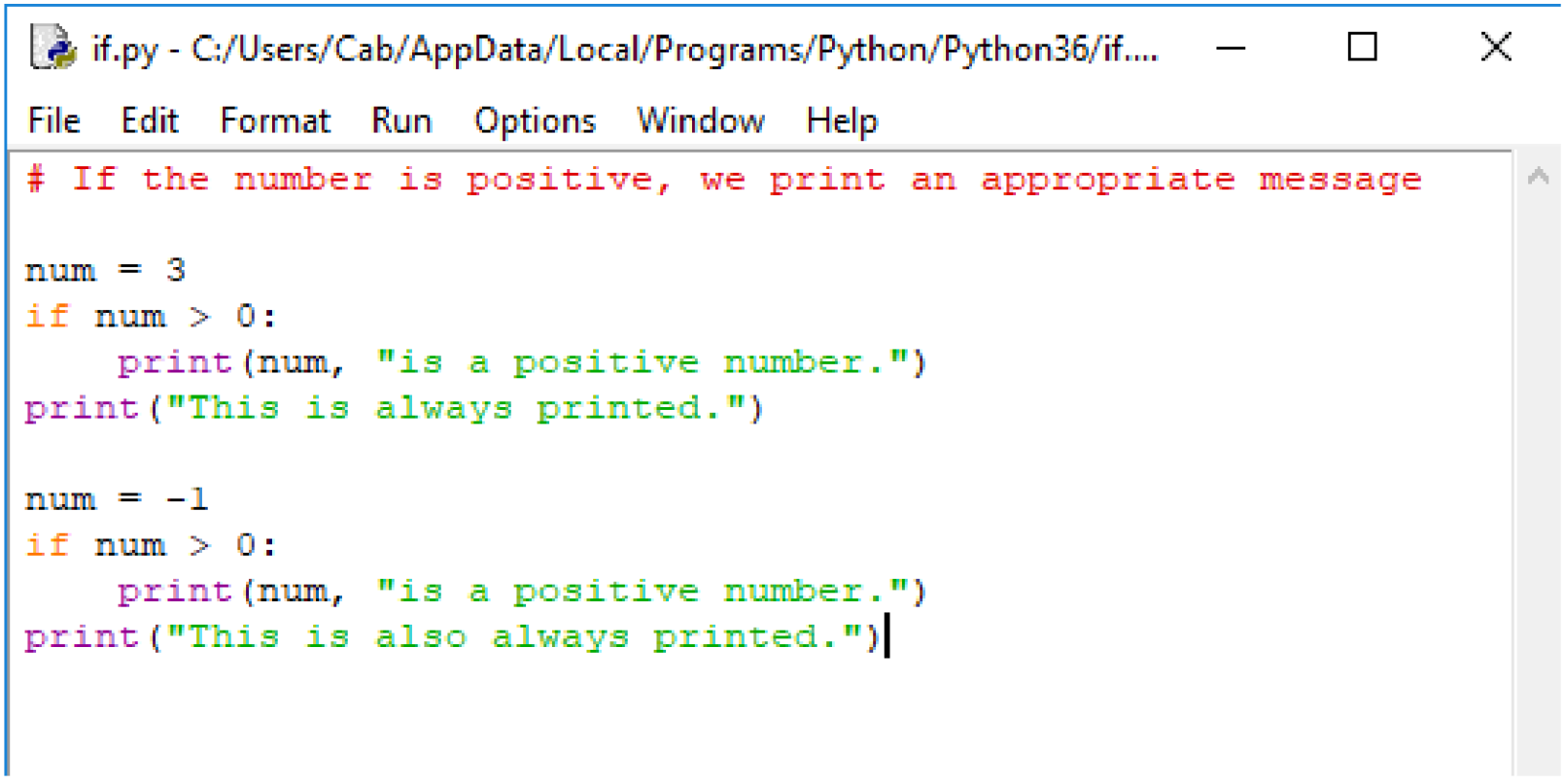
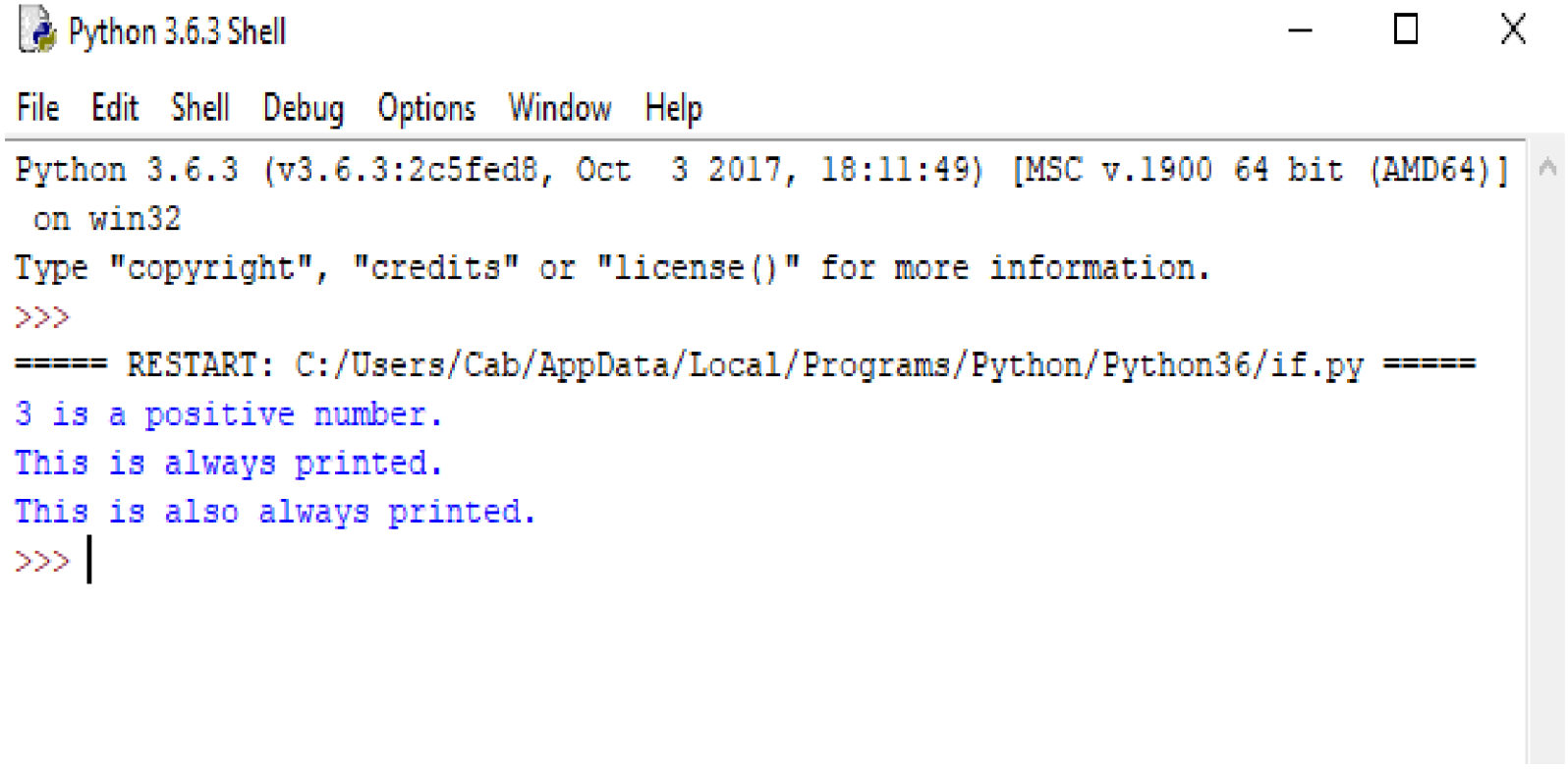


Fig: Operation of if statement

Example:


if.py - C:/Users/Cab/AppData/Local/Programs/Python/Python36/if....
File Edit Format Run Options Window Help
# If the number is positive, we print an appropriate message
num = 3
if num > 0:
print(num, "is a positive number.")
print("This is always printed.")
num = -1
if num > 0:
print(num, "is a positive number.")
print("This is also always printed.")

Output


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Cab/AppData/Local/Programs/Python/Python36/if.py =====
3 is a positive number.
This is always printed.
This is also always printed.
>>>

Syntax of if...else

- Syntax

```
if test expression:
     Body of if
else:
     Body of else
```

Python if..else Flowchart

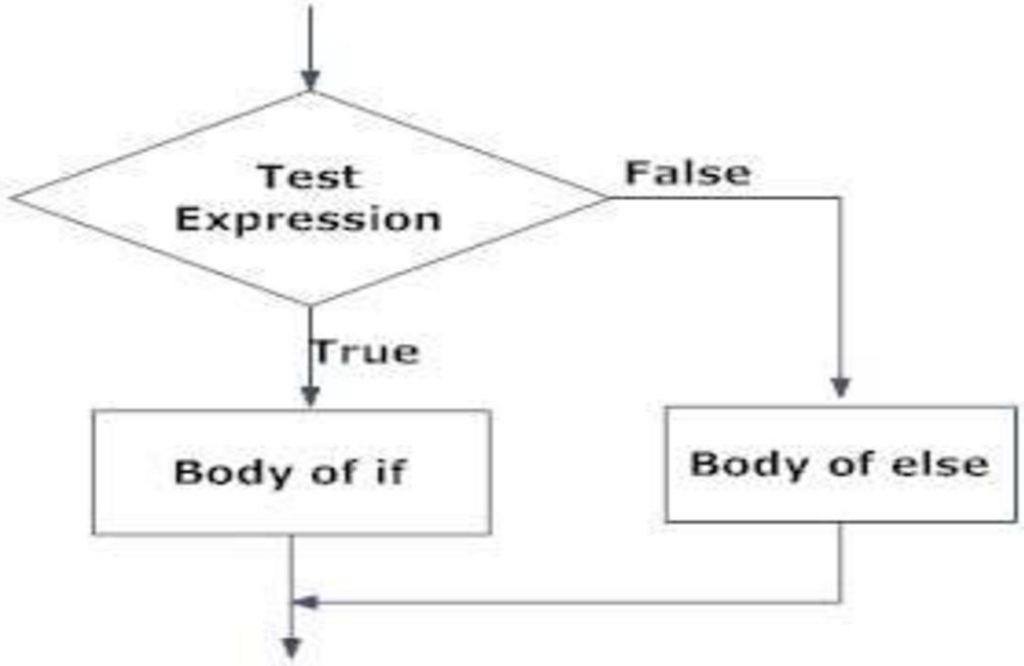


Fig: Operation of if...else statement

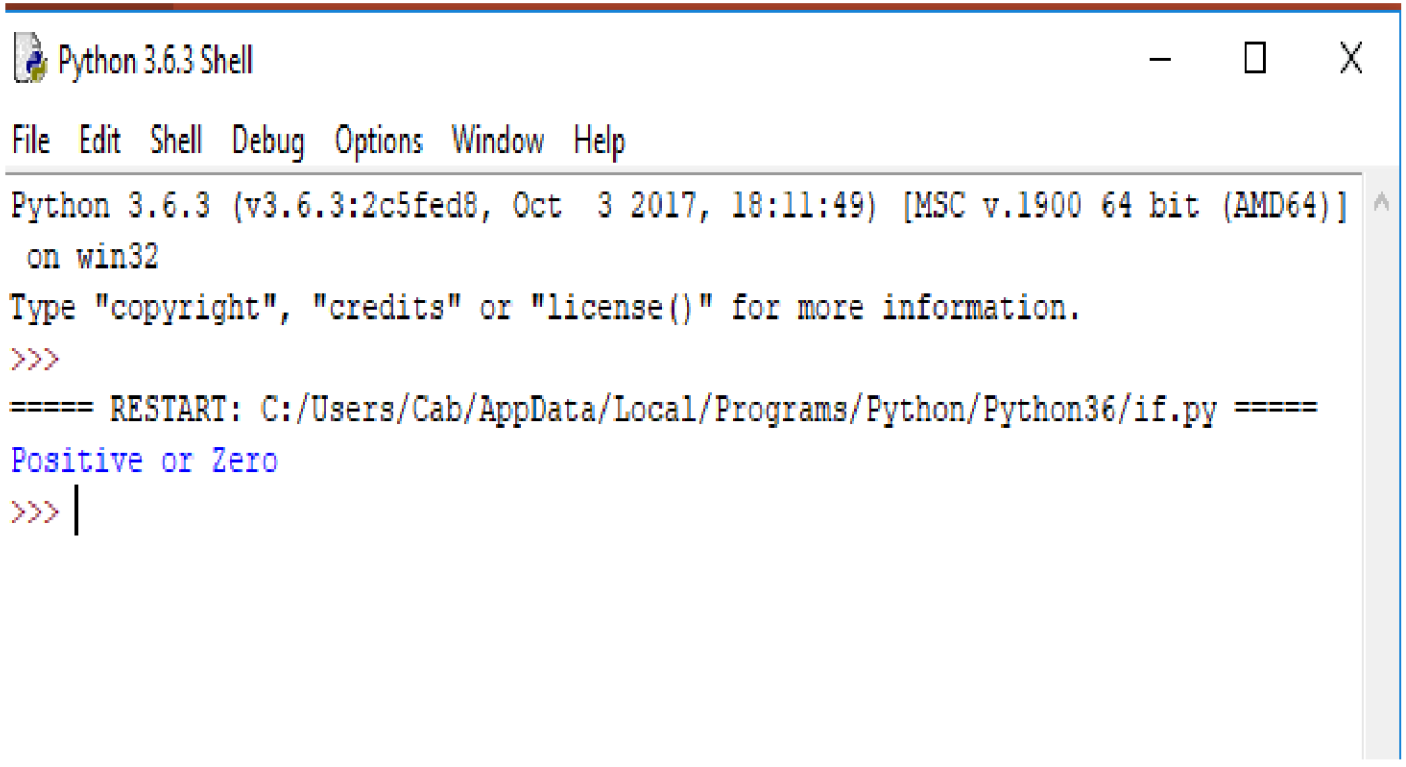
Example of if...else

if.py - C:/Users/Cab/AppData/Local/Programs/Python/Python36/if....

File Edit Format Run Options Window Help

```
# Program checks if the number is positive or negative
# And displays an appropriate message

num = 3

# Try these two variations as well.
# num = -5
# num = 0

if num >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
```

Output


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Cab/AppData/Local/Programs/Python/Python36/if.py =====
Positive or Zero
>>>

Syntax of if...elif...else

- Syntax

```
if test expression:
     Body of if
elif test expression:
     Body of elif
else:
     Body of else
```

Flowchart of if...elif...else

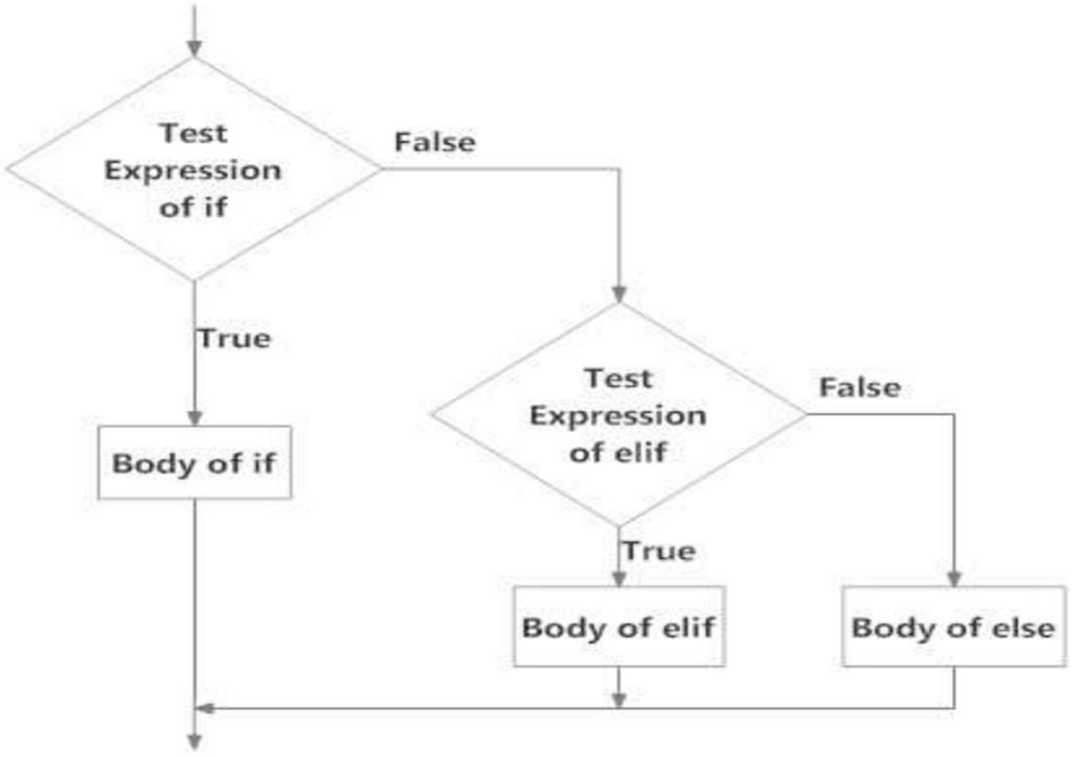
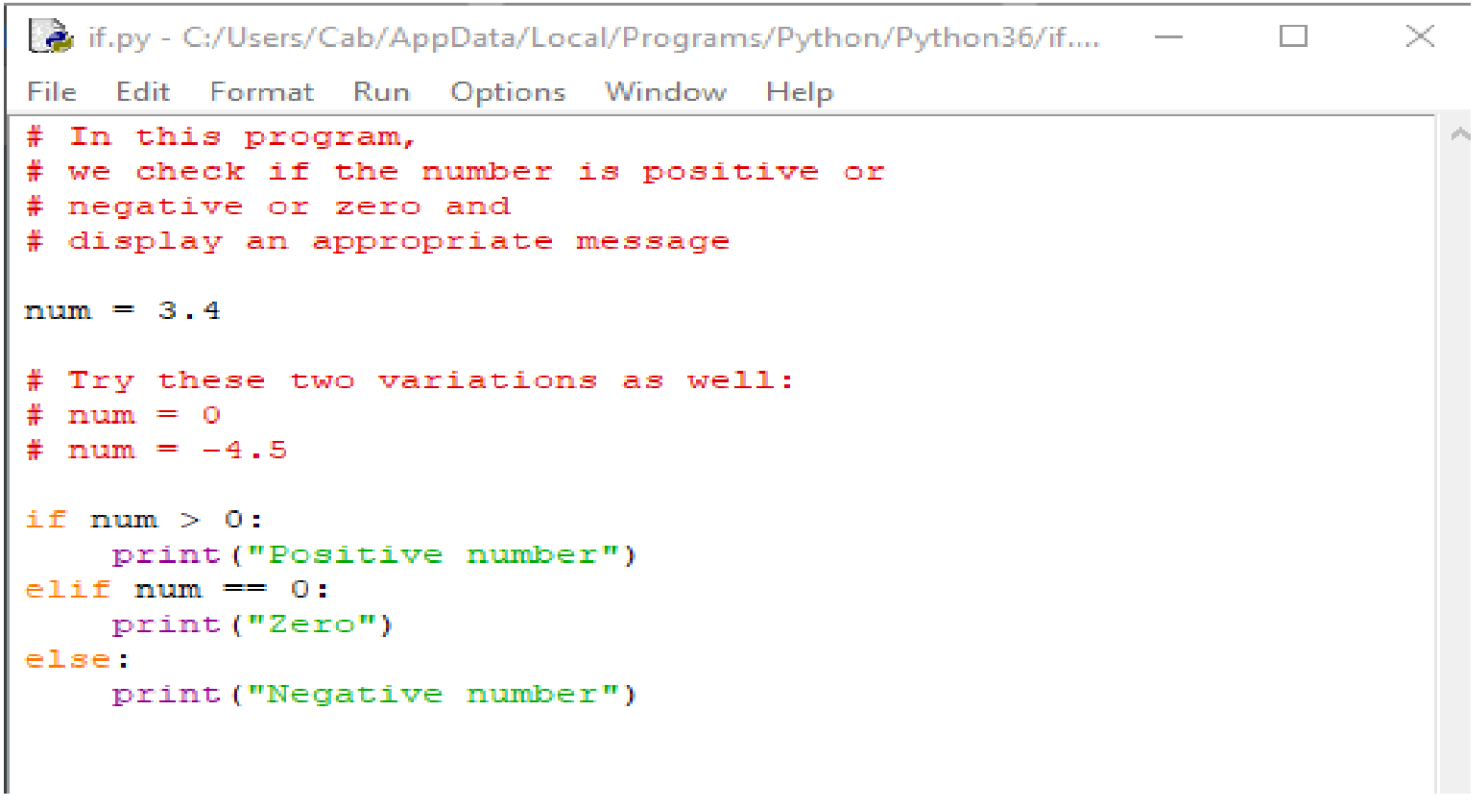
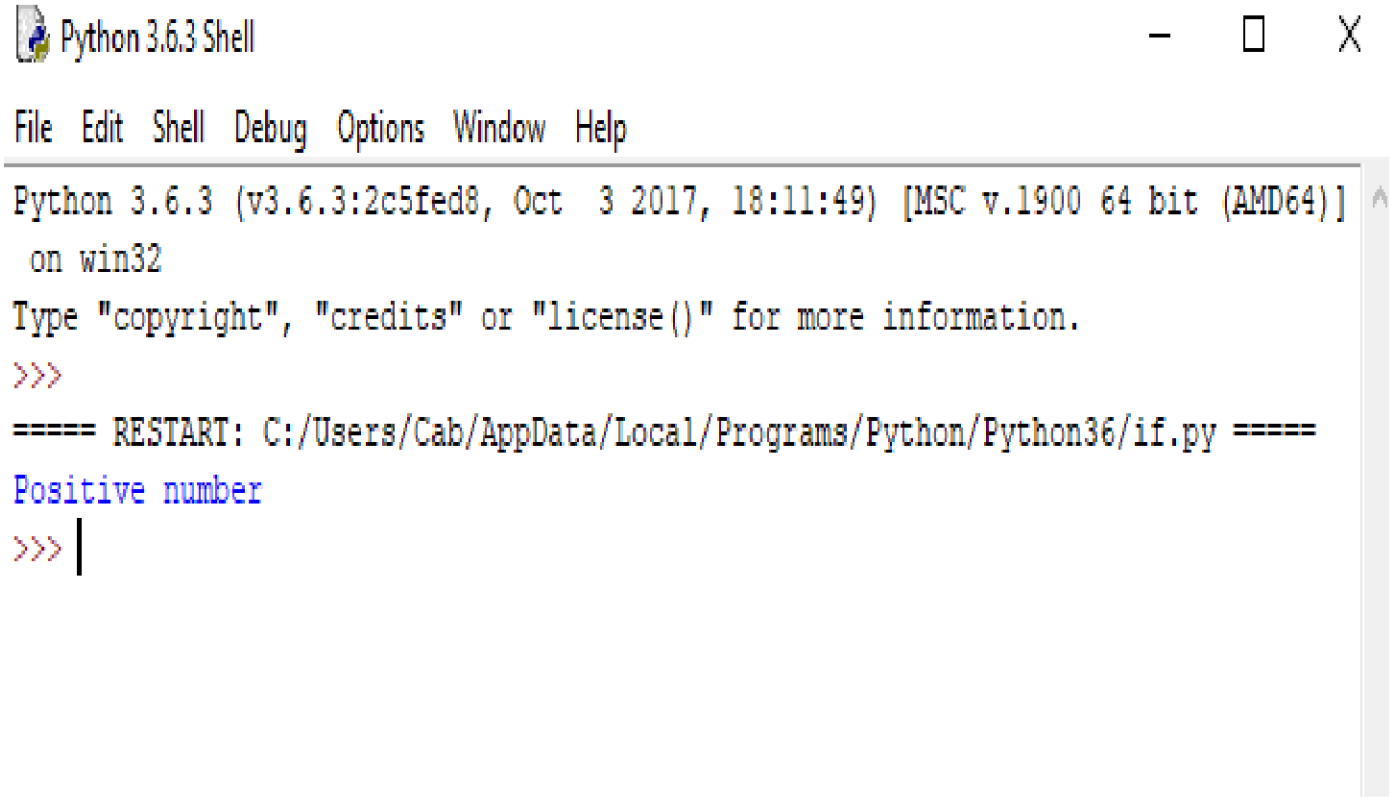


Fig: Operation of if...elif...else statement

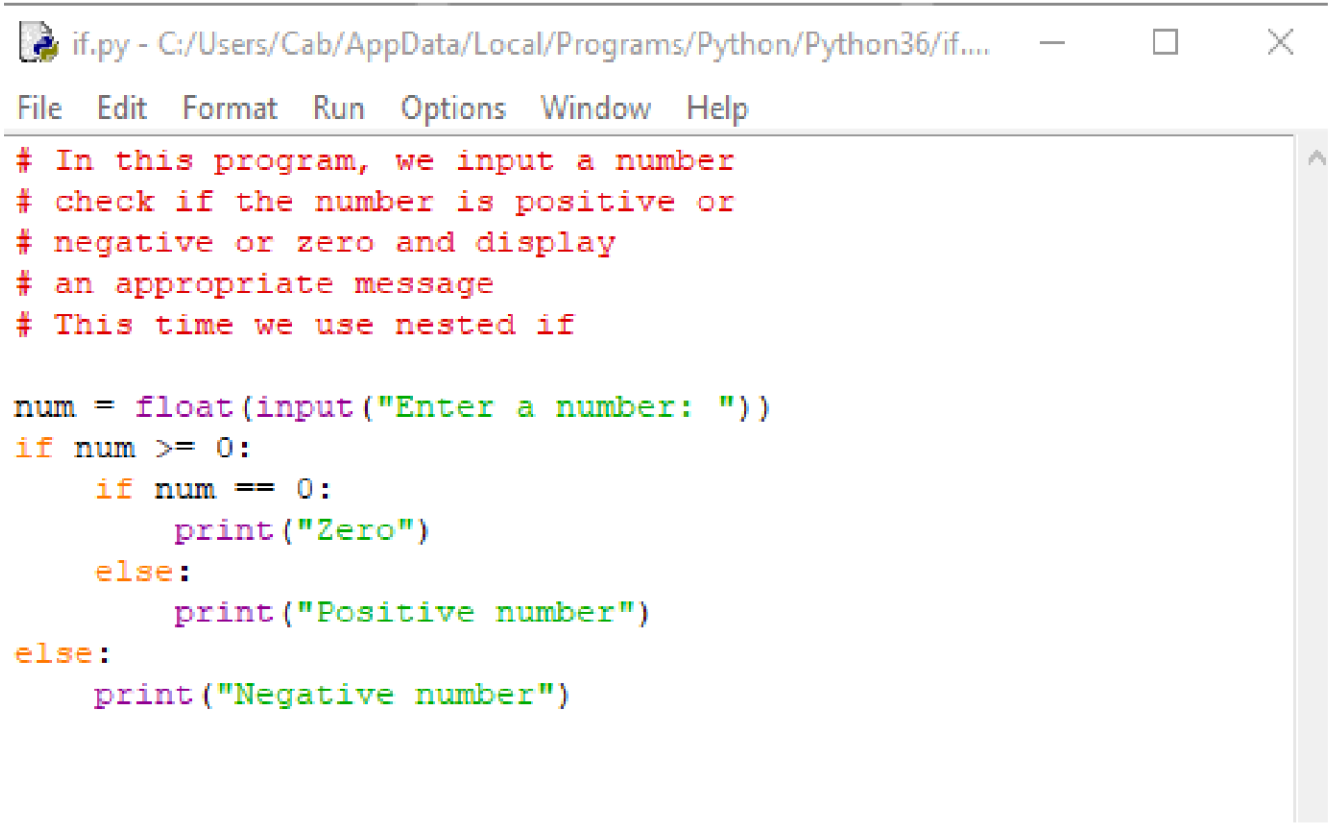
Example of if...elif...else



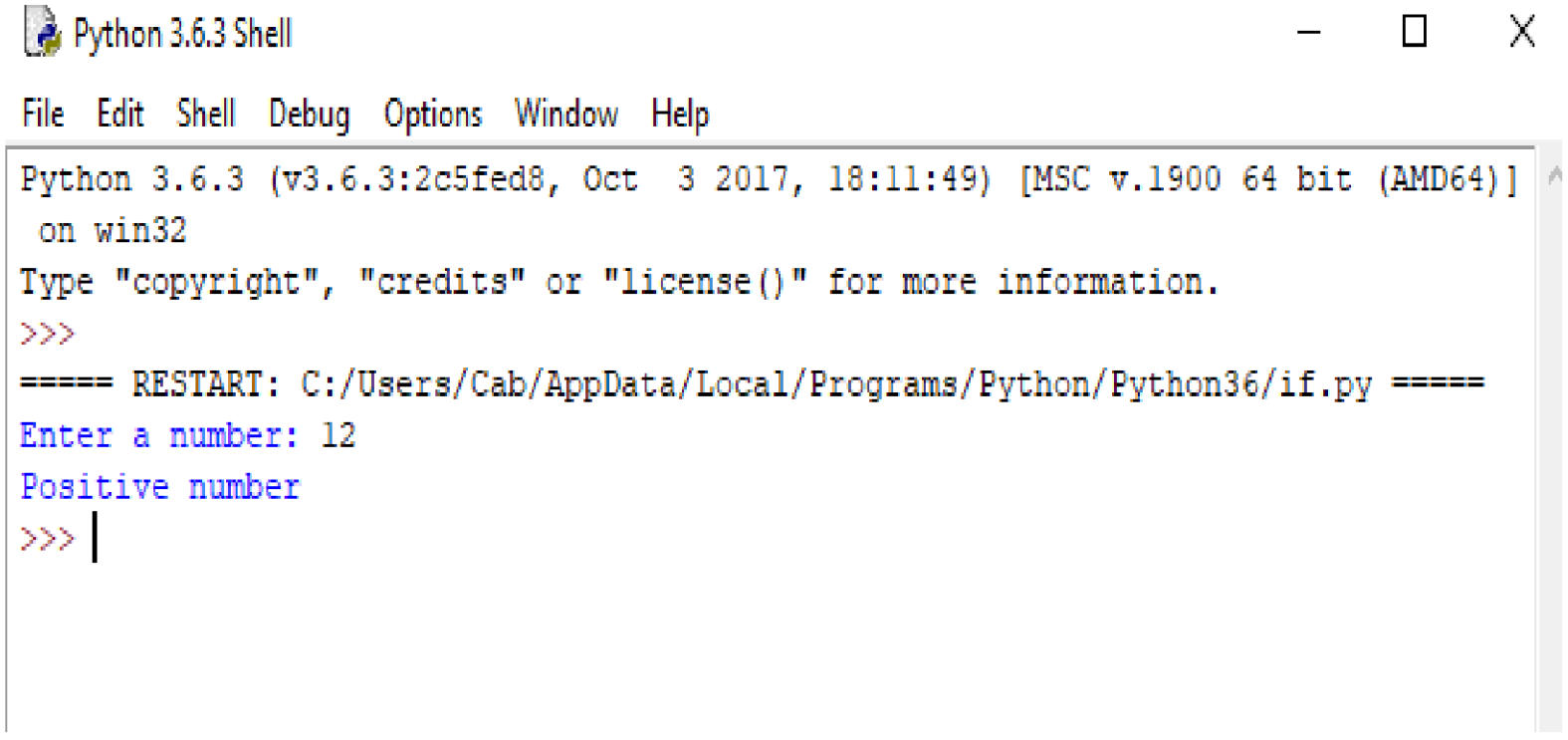
Output


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Cab/AppData/Local/Programs/Python/Python36/if.py =====
Positive number
>>>

Python Nested if statements



Output


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Cab/AppData/Local/Programs/Python/Python36/if.py =====
Enter a number: 12
Positive number
>>>

Python for Loop

- What is for loop in Python?
  - Syntax of for Loop
  - Flowchart of for loop
  - Example: Python for Loop
- The range() function
- for loop with else

What is for loop in Python?

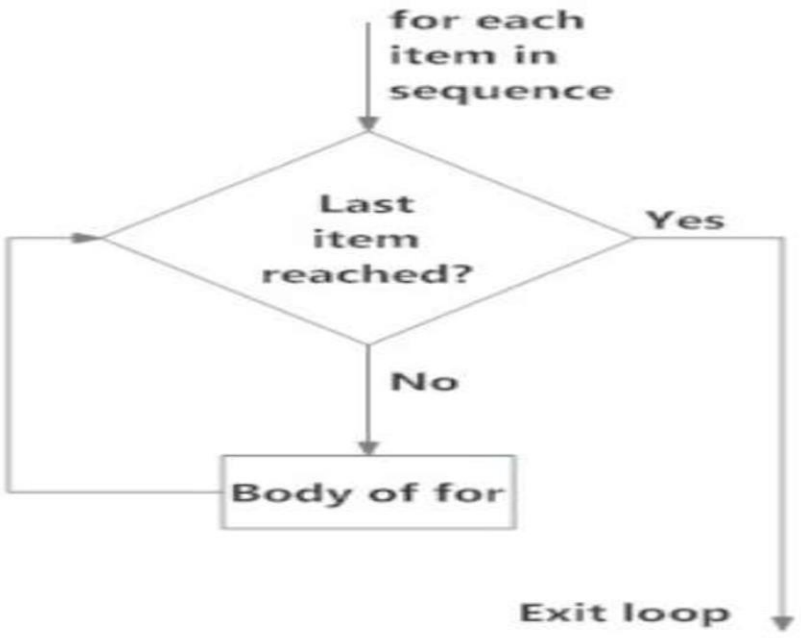
- Is used to iterate over a sequence (list, tuple, string) or other iterable objects
- Iterating over a sequence is called traversal

- Syntax of for Loop:

```
for val in sequence:
        Body of for
```

- Here, val is the variable that takes the value of the item inside the sequence on each iteration
- Loop continues until we reach the last item in the sequence. The body of for loop is separated from the rest of the code using indentation.

Flowchart of for Loop

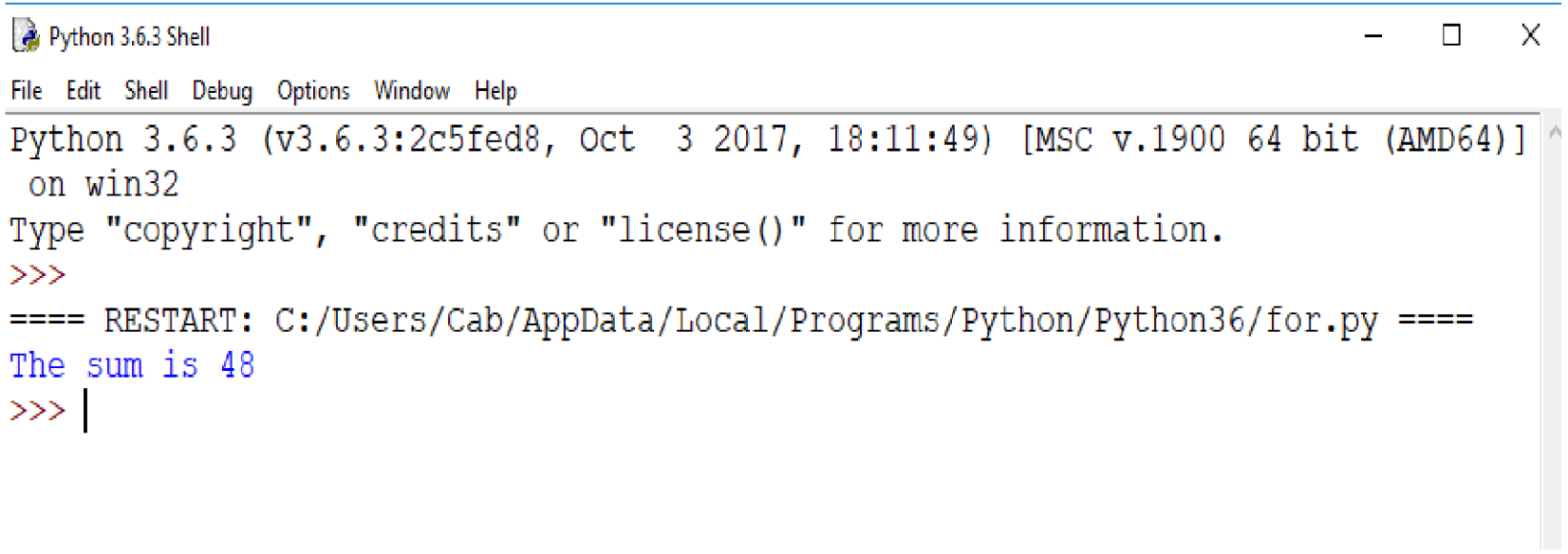


Example

for.py - C:/Users/Cab/AppData/Local/Programs/Python/Python36/for.py (3.6.3)

File Edit Format Run Options Window Help

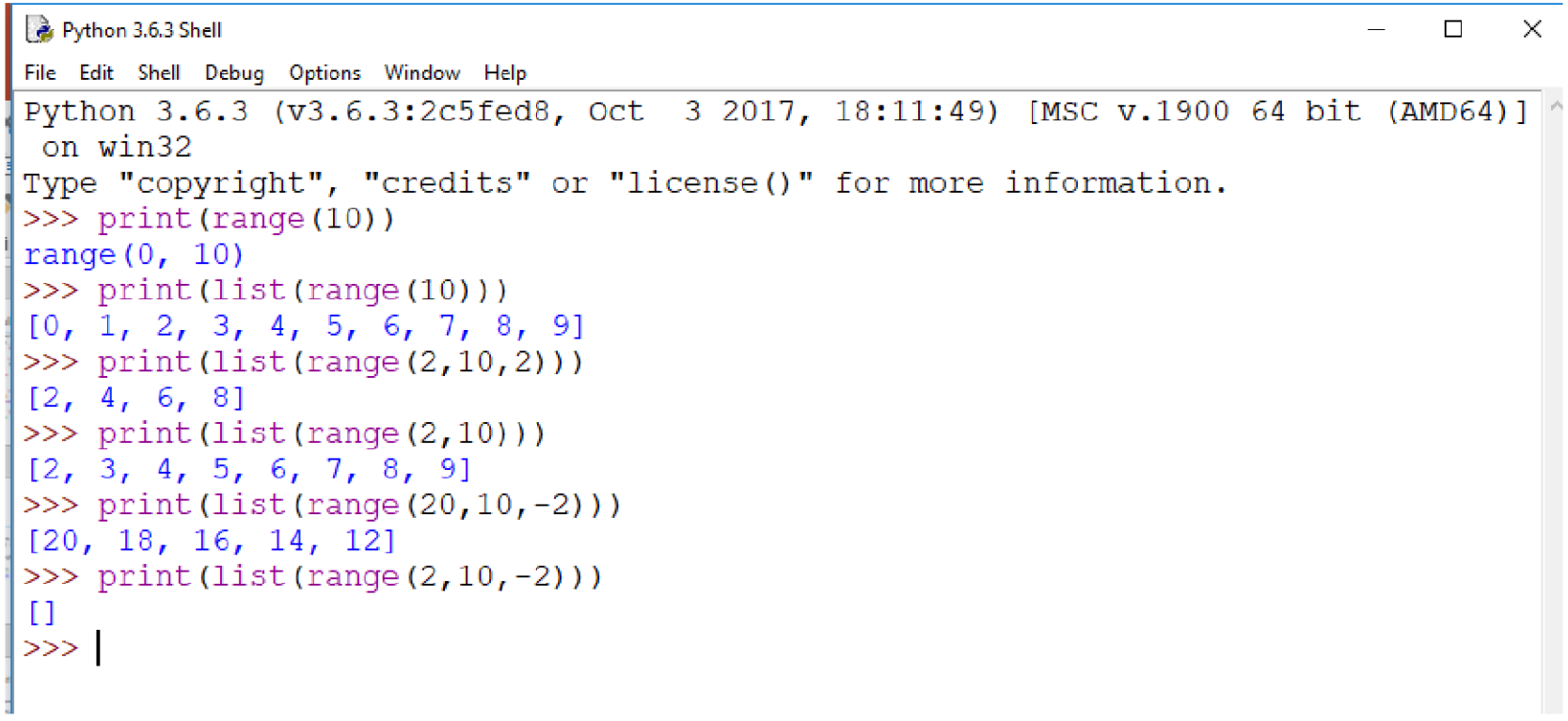
```
# Program to find the sum of all numbers stored in a list

# List of numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]

# variable to store the sum
sum = 0

# iterate over the list
for val in numbers:
        sum = sum+val

# Output: The sum is 48
print("The sum is", sum)
```

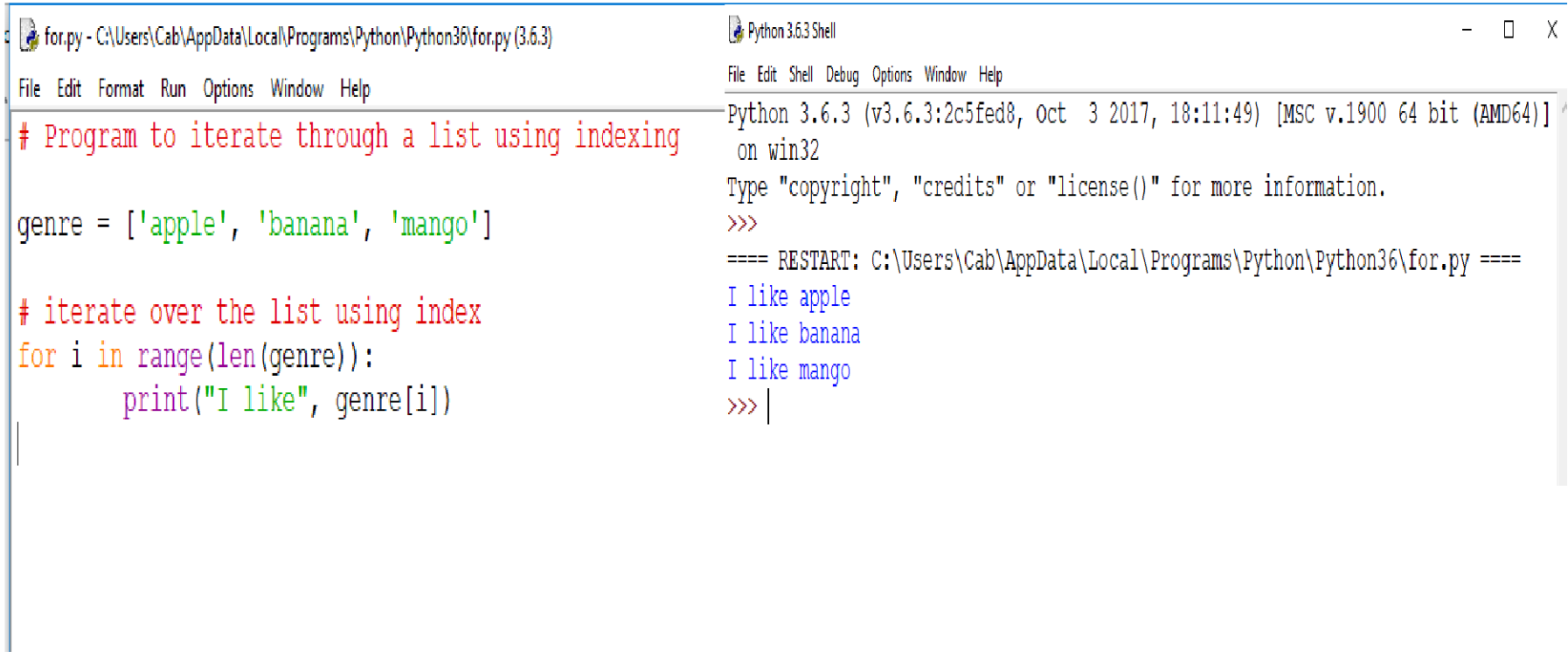

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Cab/AppData/Local/Programs/Python/Python36/for.py ====
The sum is 48
>>>

The range() function

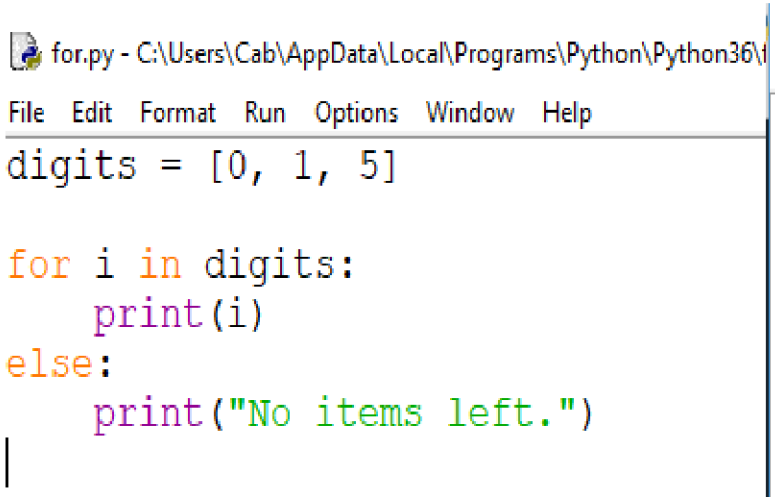
- Can generate a sequence of numbers using range() function
- range(10) will generate numbers from 0 to 9 (10 numbers)
- Can also define the start, stop and step size as range(start,stop,stepsize)
- Step size defaults to 1 if not provided.
- Does not store all the values in memory, it would be inefficient
- So it remembers the start, stop, step size and generates the next number on the go

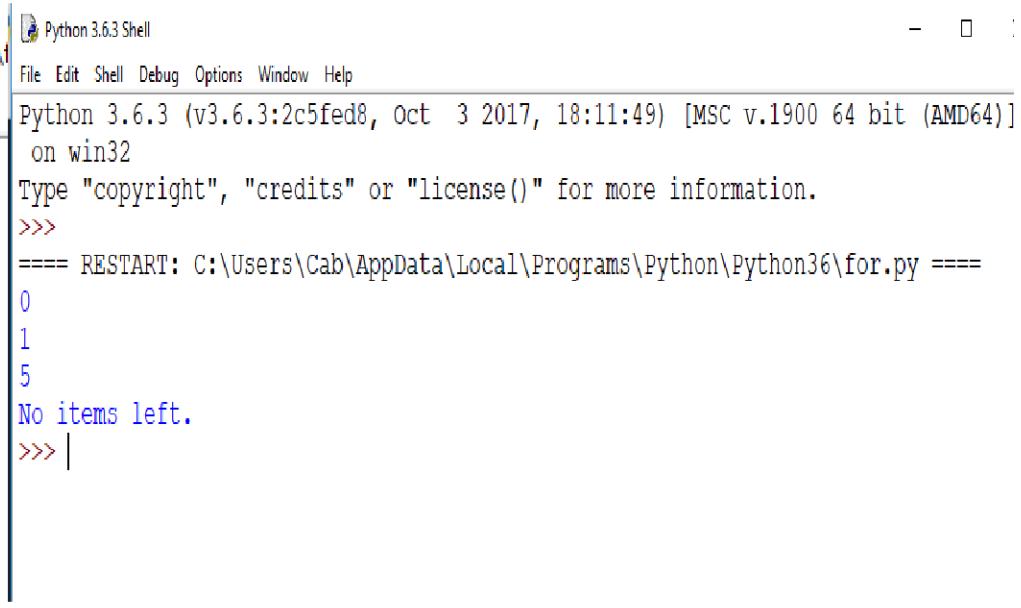
Example


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(range(10))
range(0, 10)
>>> print(list(range(10)))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> print(list(range(2,10,2)))
[2, 4, 6, 8]
>>> print(list(range(2,10)))
[2, 3, 4, 5, 6, 7, 8, 9]
>>> print(list(range(20,10,-2)))
[20, 18, 16, 14, 12]
>>> print(list(range(2,10,-2)))
[]
>>>


for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)
File Edit Format Run Options Window Help
# Program to iterate through a list using indexing
genre = ['apple', 'banana', 'mango']
# iterate over the list using index
for i in range(len(genre)):
print("I like", genre[i])
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
I like apple
I like banana
I like mango
>>>

for loop with else





Python while Loop

- What is while loop in Python?
  - Syntax of while Loop in Python
  - Flowchart of while loop
  - Example: Python while Loop
- while loop with else

What is while loop in Python?

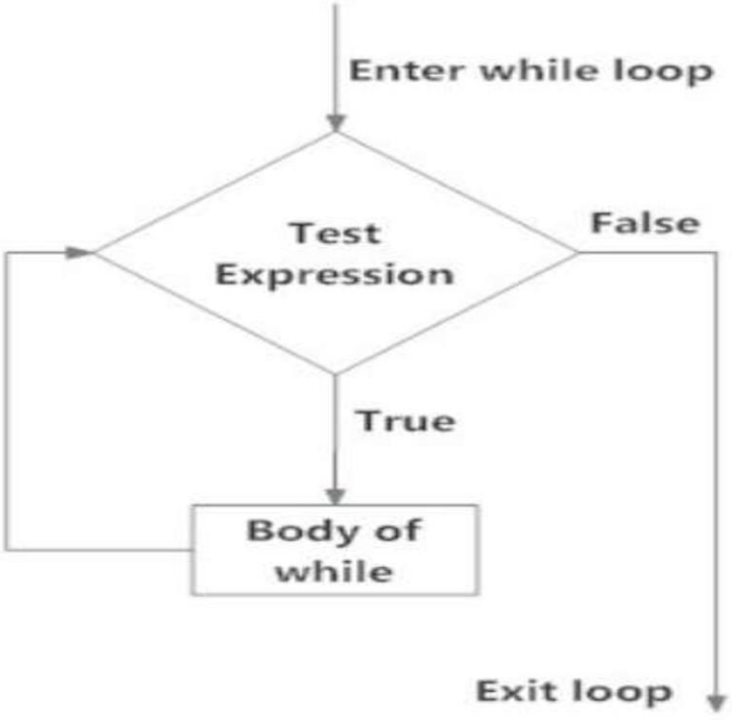
- Is used to iterate over a block of code as long as the test expression (condition) is true
- Generally this loop is used when we don't know beforehand, the number of times to iterate

Syntax of while Loop in Python

```
while test_expression:
    Body of while
```

- Python interprets any non-zero value as True. None and 0 are interpreted as False.

Flowchart of while Loop

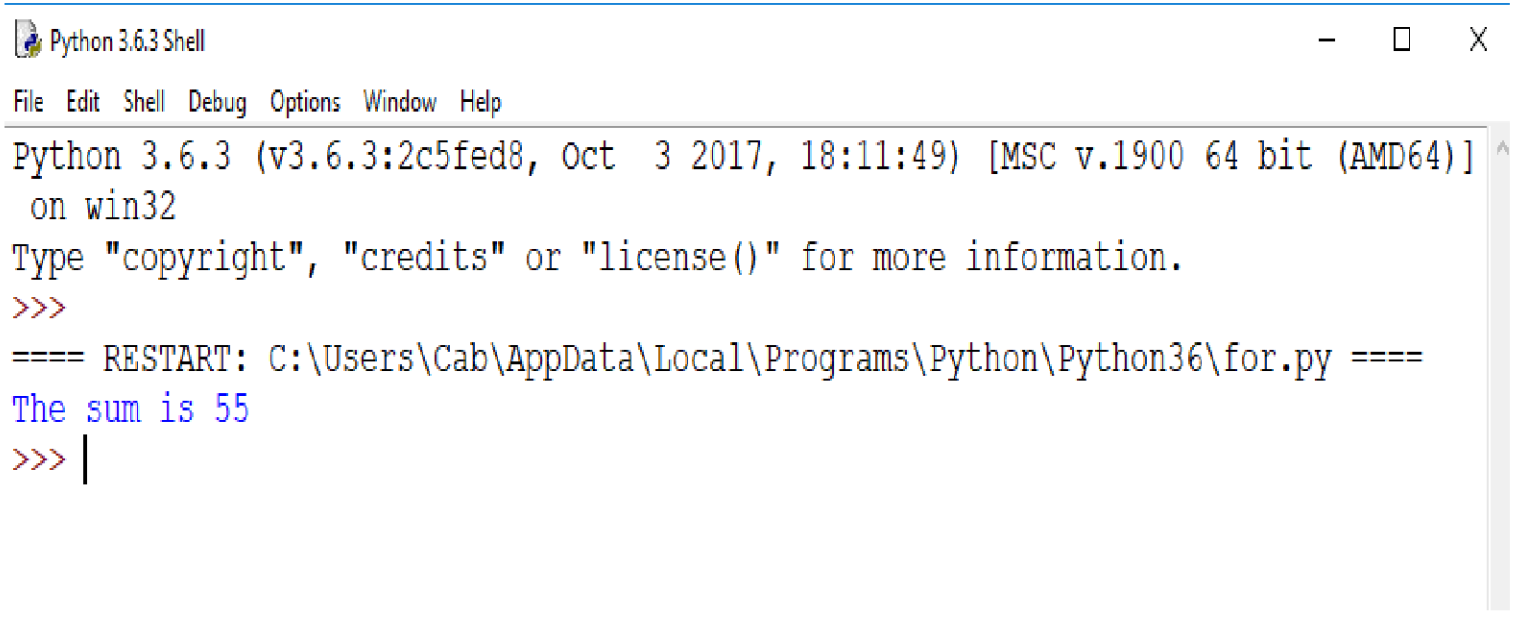


Example

for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)

File Edit Format Run Options Window Help

```
# Program to add natural numbers
# sum = 1+2+3+...+n

# To take input from the user,
# n = int(input("Enter n: "))

n = 10

# initialize sum and counter
sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1     # update counter

# print the sum
print("The sum is", sum)
```

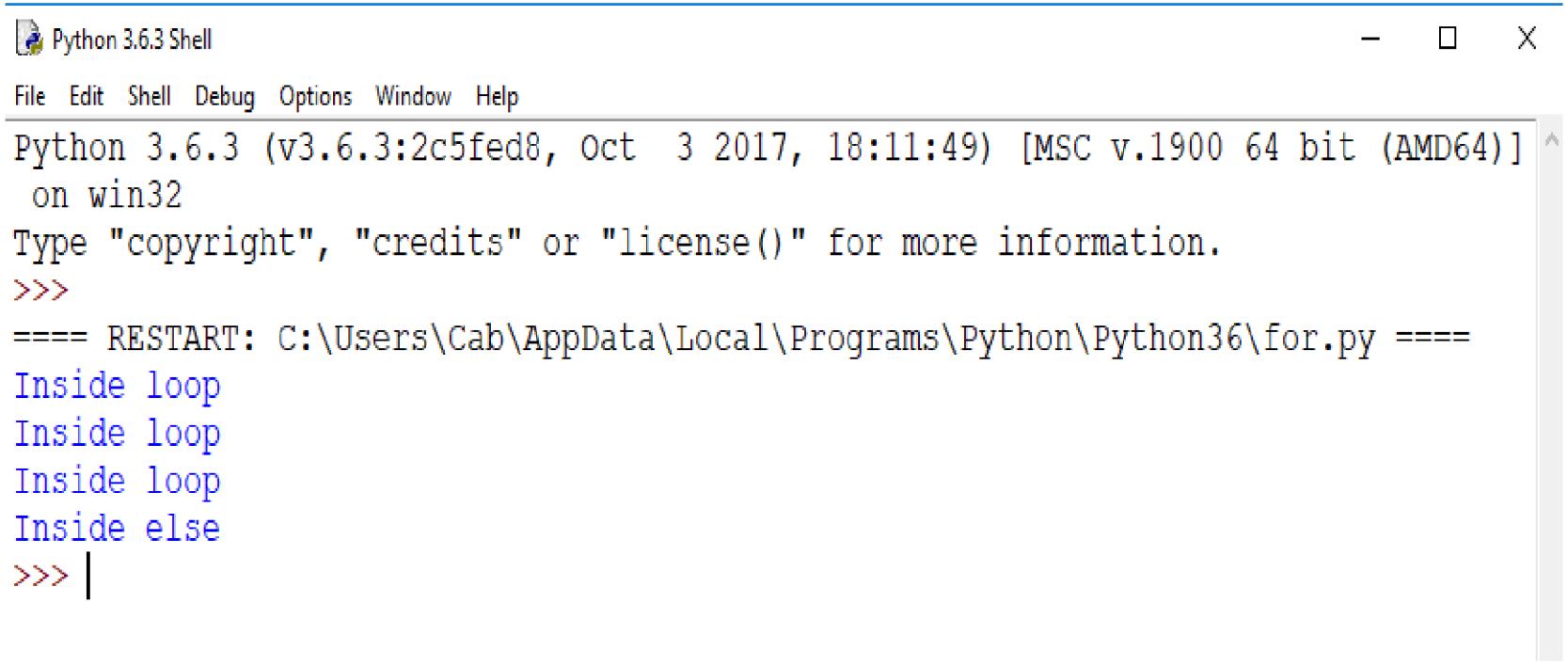

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
The sum is 55
>>>

while loop with else

for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)

File Edit Format Run Options Window Help

```
# Example to illustrate
# the use of else statement
# with the while loop

counter = 0

while counter < 3:
    print("Inside loop")
    counter = counter + 1
else:
    print("Inside else")
```


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
Inside loop
Inside loop
Inside loop
Inside else
>>>

Python break and continue

- What is the use of break and continue in Python?
- Python break statement
  - Syntax of break
  - Flowchart of break
  - Example of break
- Python continue statement
  - Syntax of Continue
  - Flowchart of continue
  - Example: Python continue

What is the use of break and continue in Python?

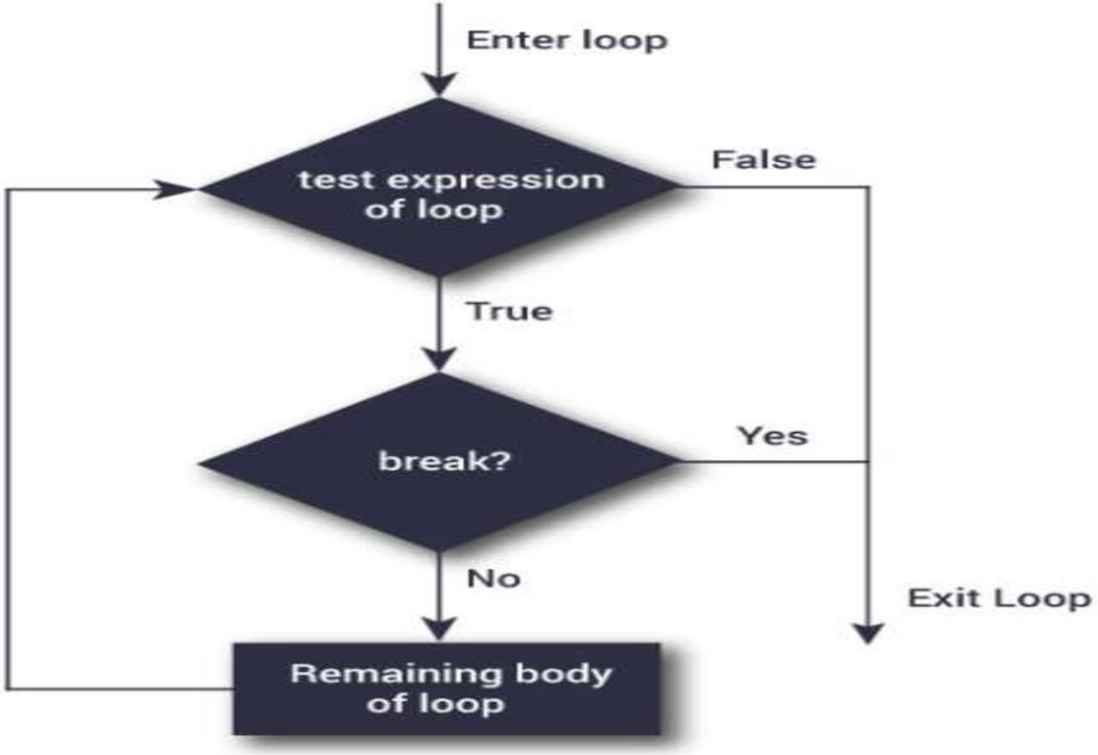
- break and continue statements can alter the flow of a normal loop
- Loops iterate over a block of code until test expression is false, but sometimes we wish to terminate the current iteration or even the whole loop without checking test expression
- break and continue statements are used in these cases

Python break statement

- Terminates the loop containing it
- Control of the program flows to the statement immediately after the body of the loop.
- If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop
- Syntax of break

```
break
```

Flowchart of break

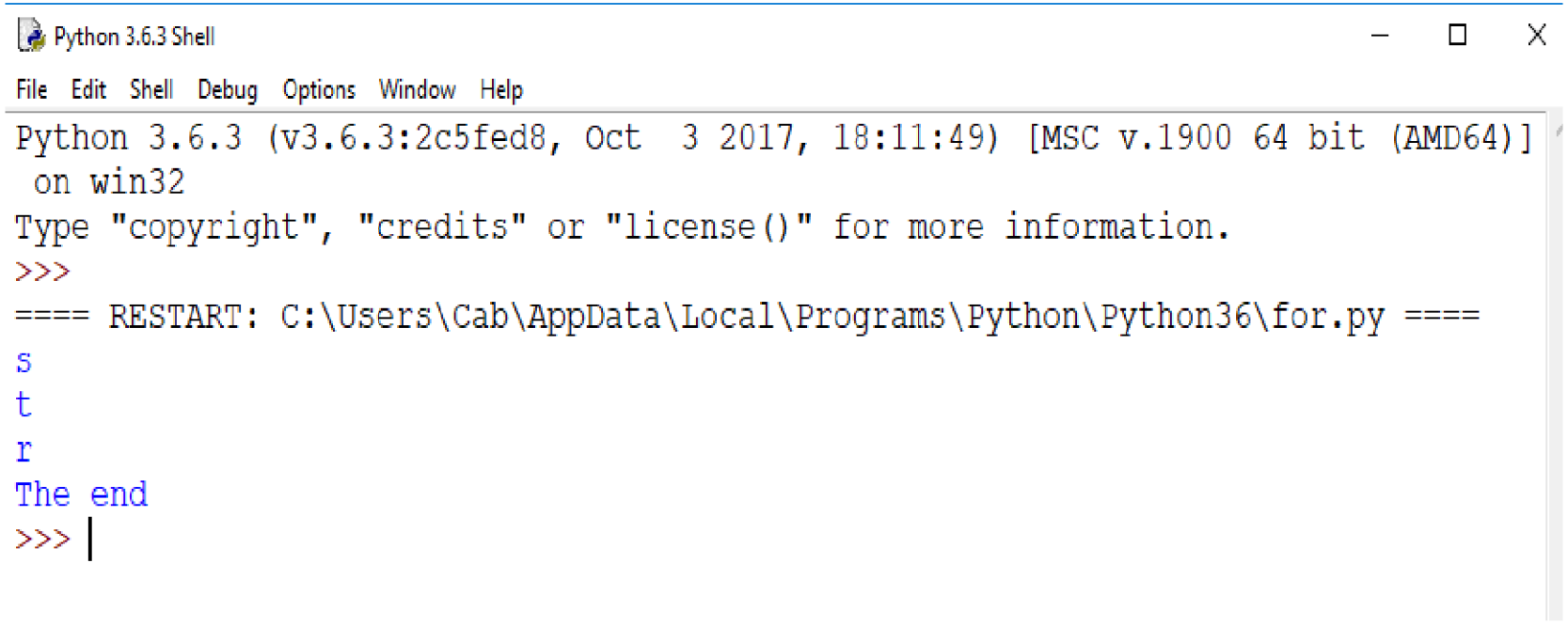


Example

for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)

File Edit Format Run Options Window Help

```
# Use of break statement inside loop

for val in "string":
    if val == "i":
        break
    print(val)

print("The end")
```

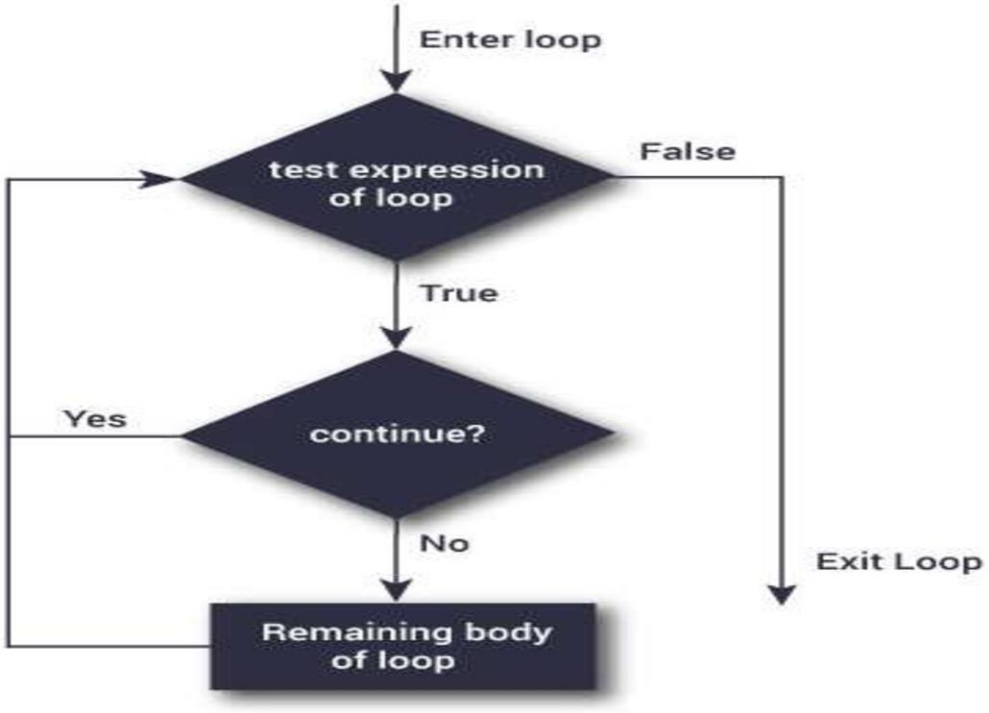

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
s
t
r
The end
>>>

Python continue statement

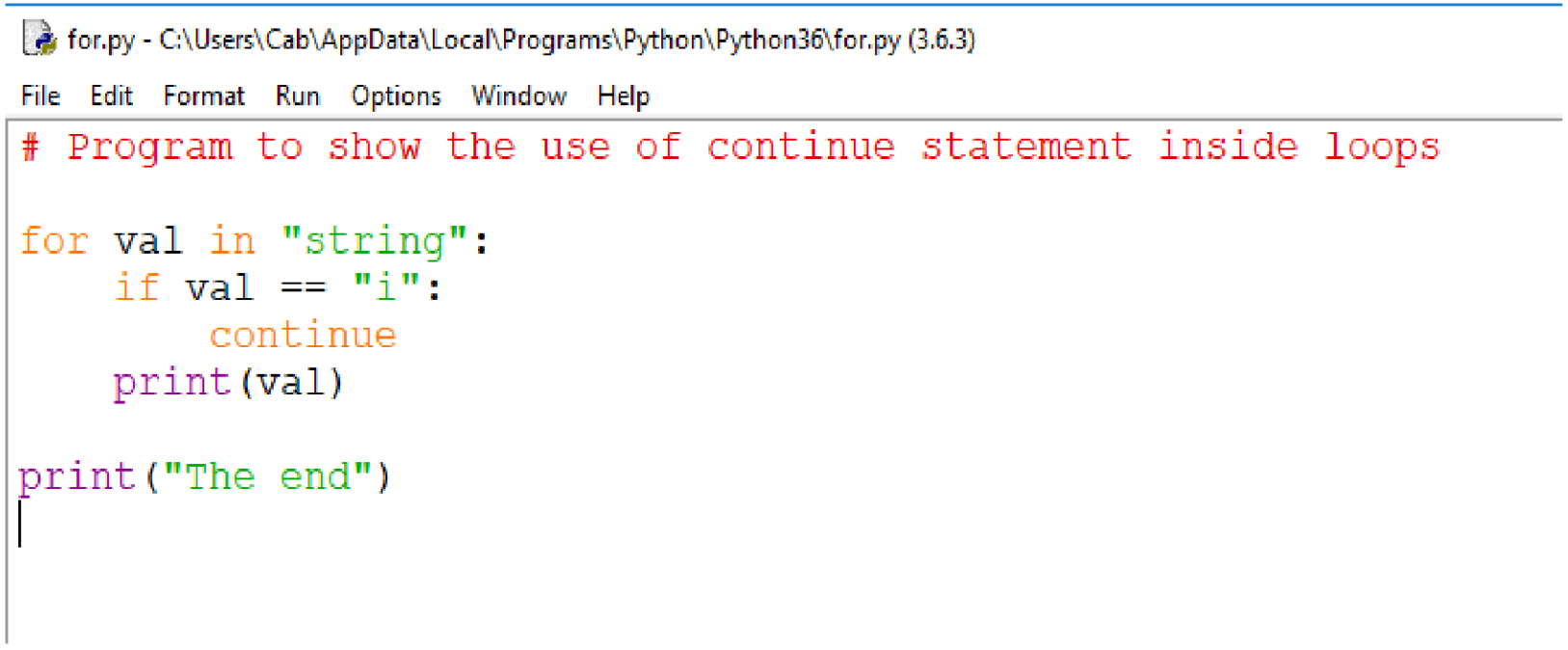
- Is used to skip the rest of the code inside a loop for the current iteration only
- Loop does not terminate but continues on with the next iteration
- Syntax of Continue

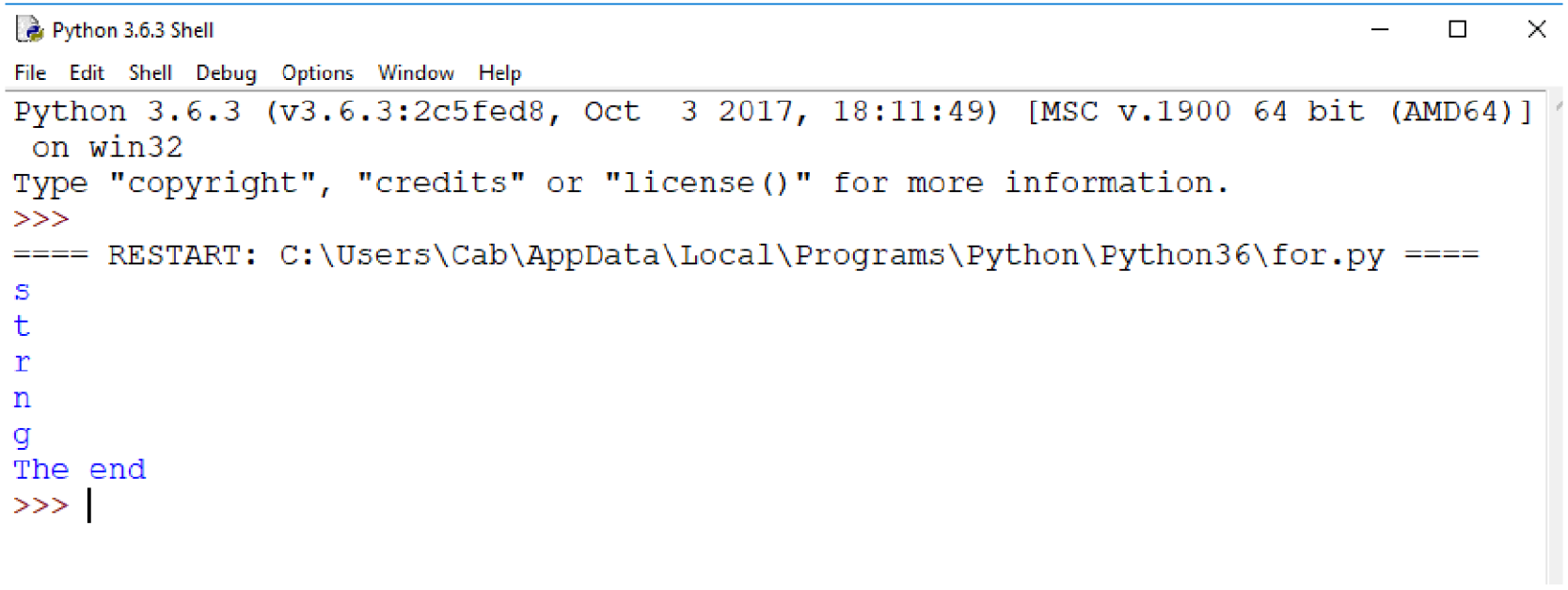
```
continue
```

Flowchart of continue



Example


for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)
File Edit Format Run Options Window Help
# Program to show the use of continue statement inside loops
for val in "string":
if val == "i":
continue
print(val)
print("The end")


Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
s
t
r
n
g
The end
>>>

Python pass statement

- What is pass statement in Python?
  - Syntax of pass
  - Example: pass Statement

What is pass statement in Python?

- pass is a null statement
- The difference between a comment and pass statement in Python is that, while the interpreter ignores a comment entirely, pass is not ignored
- However, nothing happens when pass is executed
- It results into no operation (NOP)
- Syntax of pass

```
pass
```

Example

for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)

File Edit Format Run Options Window Help

```
# pass is just a placeholder for
# functionality to be added later.
sequence = {'p', 'a', 's', 's'}
for val in sequence:
    pass
```

Python 3.6.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct  3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py ====
>>>
```

Example

for.py - C:\Users\Cab\AppData\Local\Programs\Python\Python36\for.py (3.6.3)

File Edit Format Run Options Window Help

```
def fun():
        pass
print("I have called function")
fun()
```

Thank You !