Concepts on your tips

Flow Control In Python

1. Python if..else
2. Python for loop
3. Python while loop
4. Python break and continue
5. Python pass

Python if...else Statement

There are basically three forms of the if...else statement.

1. **if statement**
2. **if-else statement**
3. **if-elif-else statement**

if Statement in Python

- In control statements, The if statement is the simplest form. It takes a condition and evaluates to either True or False.
- If the condition is True, then the True block of code will be executed, and if the condition is False, then the block of code is skipped, and The controller moves to the next line
- **Syntax:-** **if** condition:
  # body of if statement

If – else statement

- The if-else statement checks the condition and executes the if block of code when the condition is True, and if the condition is False, it will execute the else block of code.
- **Syntax:-**

```
if condition:
    statement 1
else:
    statement 2
```

If – else statement

- In Python, the if-elif-else condition statement has an elif blocks to chain multiple conditions one after another. This is useful when you need to check multiple conditions.
- **Syntax:-**

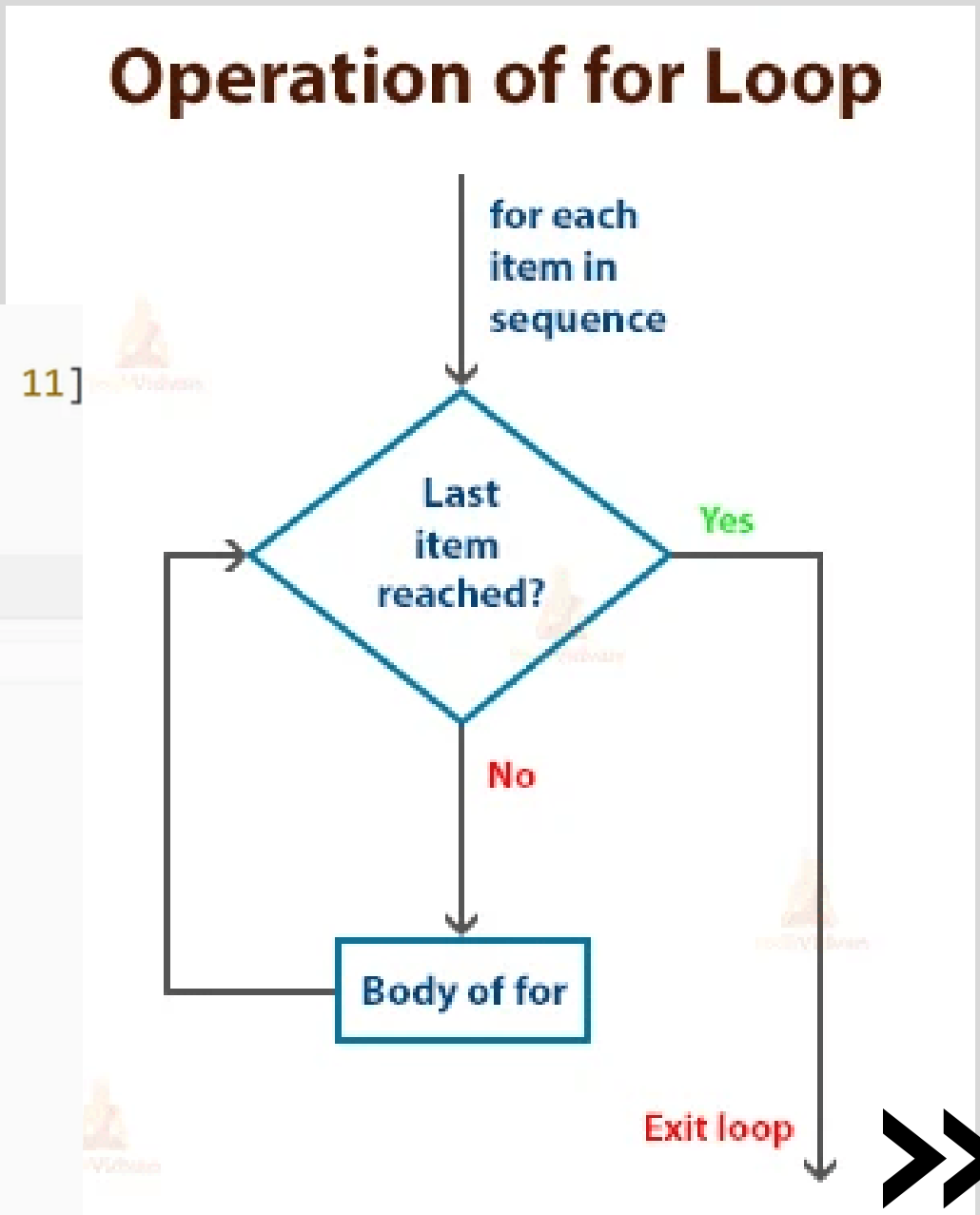
```
if condition1:
    # code block 1
elif condition2:
    # code block 2
else:
    # code block 3
```

Python for Loop

- Using for loop, we can iterate any sequence or iterable variable. The sequence can be string, list, dictionary, set, or tuple.
- **Syntax:-**

```
for val in sequence:
        # statement(s)
```

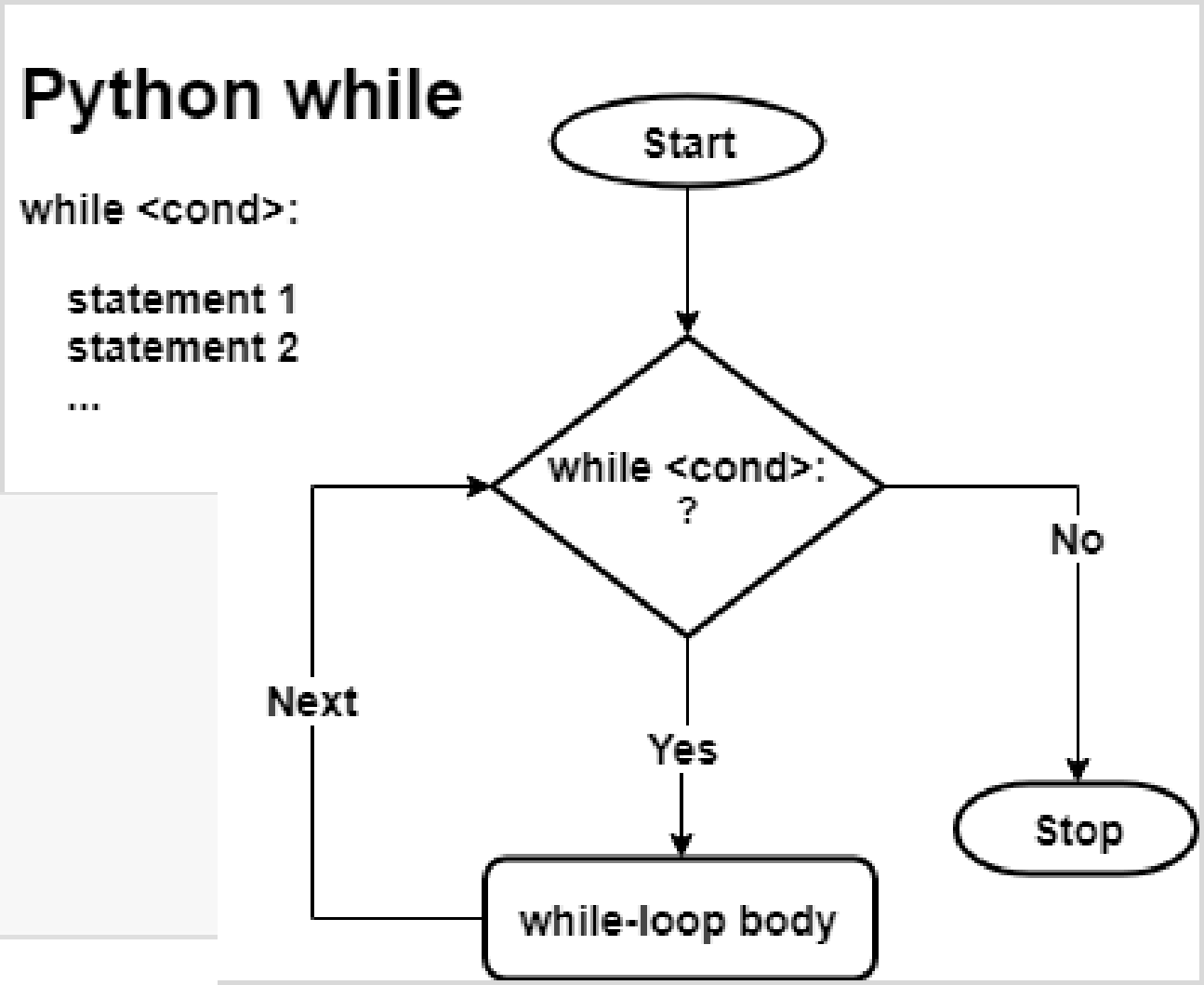
```
Python Tutorial.py
1   numbers = [10, 23, 43, 54, 65, 64, 87, 43, 21, 11]
2
3   for number in numbers:  #2
4       print (number)              #3
    for number in numbers
Run:  rw_visual  Python Tutorial
    10
    23
    43
    54
    65
    64
    87
    43
    21
```



Python while Loop

- In Python, The while loop statement repeatedly executes a code block while a particular condition is true.
- **Syntax:-**

```
while condition:
    # body of while loop
```

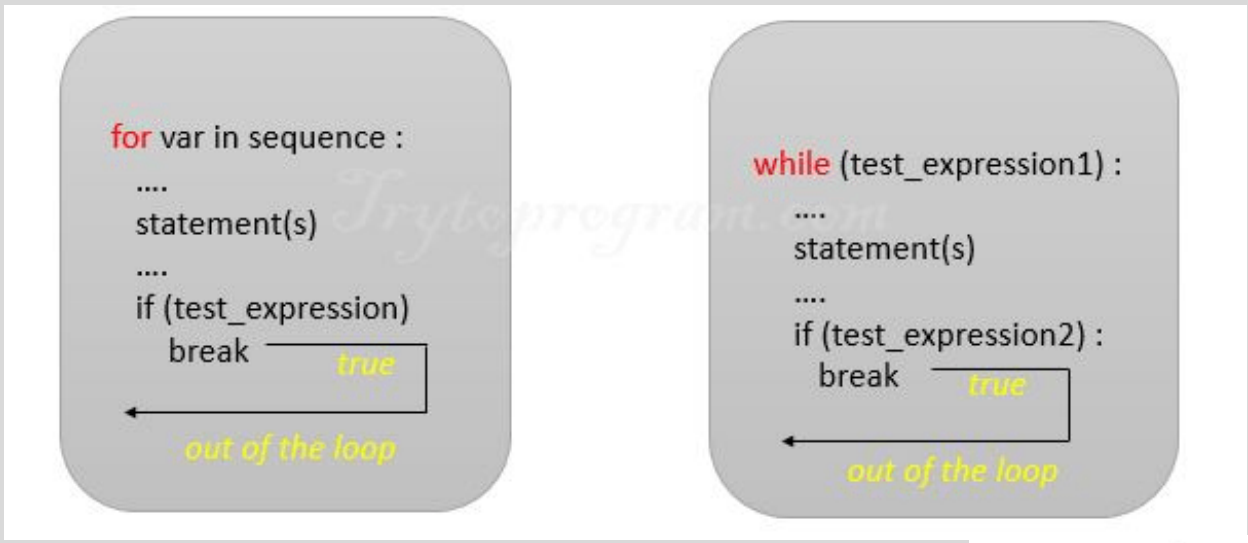



```
i = 5
while i > 0:
    i = i -1
    if i == 2:
        continue
    print("inside loop", i)
else:
    print('Inside else')
```

```
inside loop 4
inside loop 3
inside loop 1
inside loop 0
Inside else
```

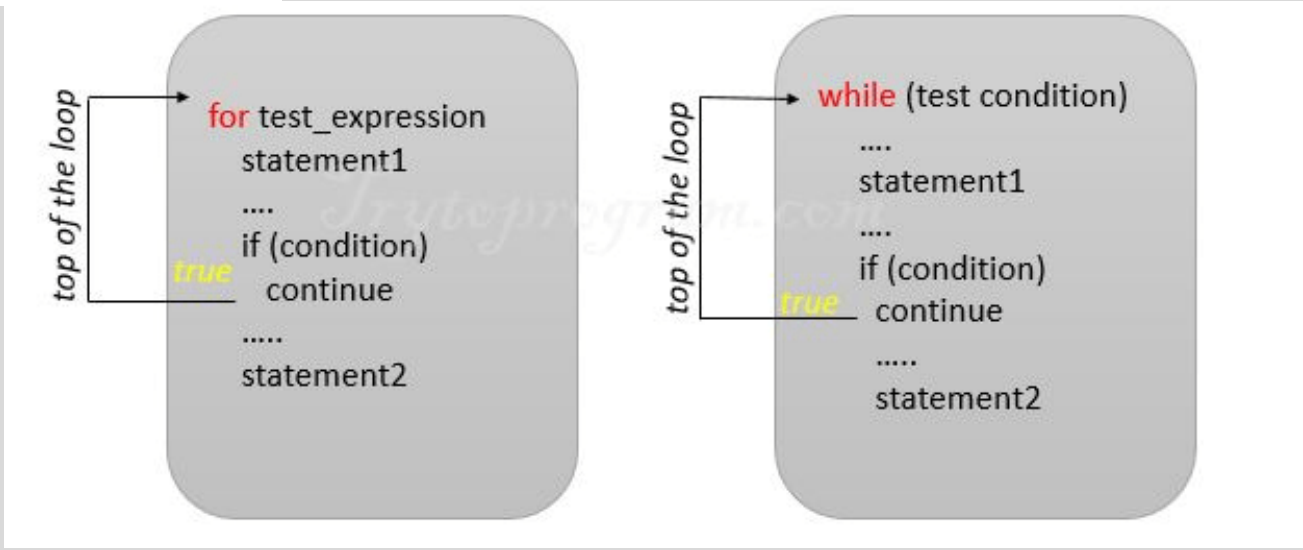
Python break & continue

- The break statement is used to terminate the loop immediately when it is encountered.
- The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration



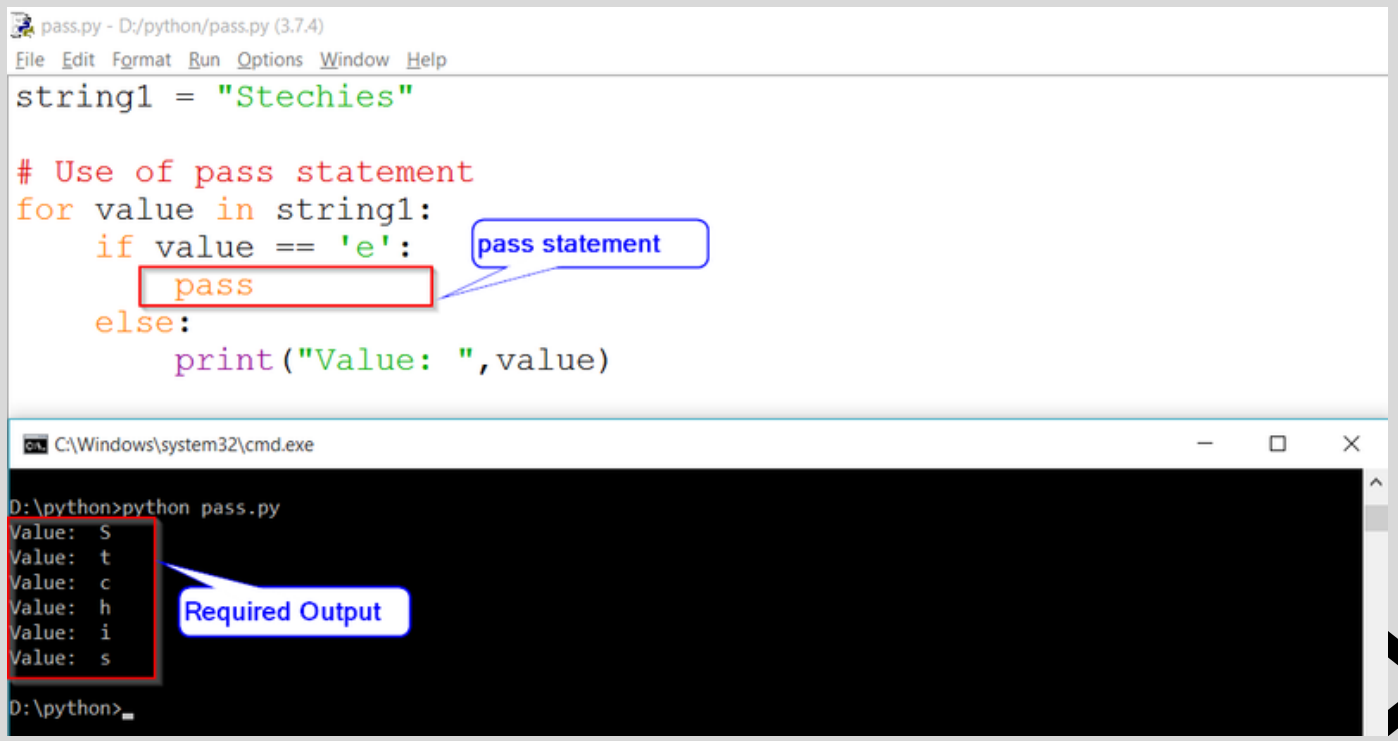
break

continue



Python pass

- In Python programming, the pass statement is a null statement which can be used as a placeholder for future code.
- A pass statement is a Python null statement. When the interpreter finds a pass statement in the program, it returns no operation. Nothing happens when the pass statement is executed.
- It is useful in a situation where we are implementing new methods or also in exception handling. It plays a role like a placeholder.



If you really like my content please don't forget to like , comment and share

Thank You

Vaishnavi Pandey