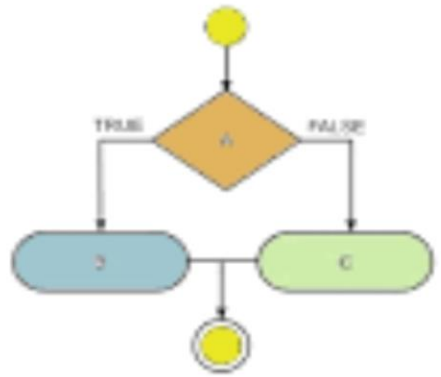
Python

DIGITAL REGENESYS

Python Conditional Statements



DIGITAL REGENESYS

Decision Making Statements

Some times in Programming Languages we need to make some decisions and based on these decisions we will execute the next block of code.

Decision-making statements in programming languages decide the direction of the flow of program execution.

In Python, **if-else elif** statement is used for decision making.

It is also called control flow statements, conditional statements.

if statement

if statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

if test expression:

    statement(s)

Python if Statement Flowchart

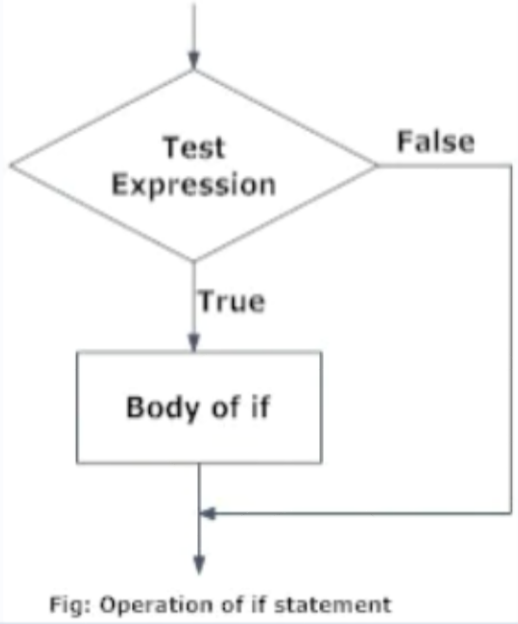


Fig: Operation of if statement

Flowchart of if statement in Python programming

DIGITAL REGENESYS

else statement

We can use the *else* statement with *if* statement to execute a block of code when the condition is false.

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

Python if..else Flowchart

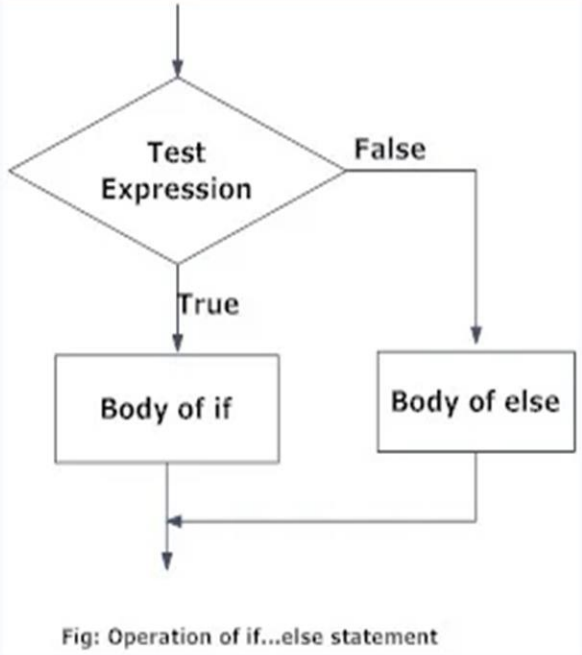


Fig: Operation of if...else statement

Flowchart of if...else statement in Python

Python if...elif...else Statement

The elif is short for else if. It allows us to check for multiple expressions. If the condition for if is False, it checks the condition of the next elif block and so on. If all the conditions are False, the body of else is executed.

Only one block among the several if...elif...else blocks is executed according to the condition. The if block can have only one else block. But it can have multiple elif blocks.

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

DIGITAL REGENESYS

Flowchart of if..elif..else

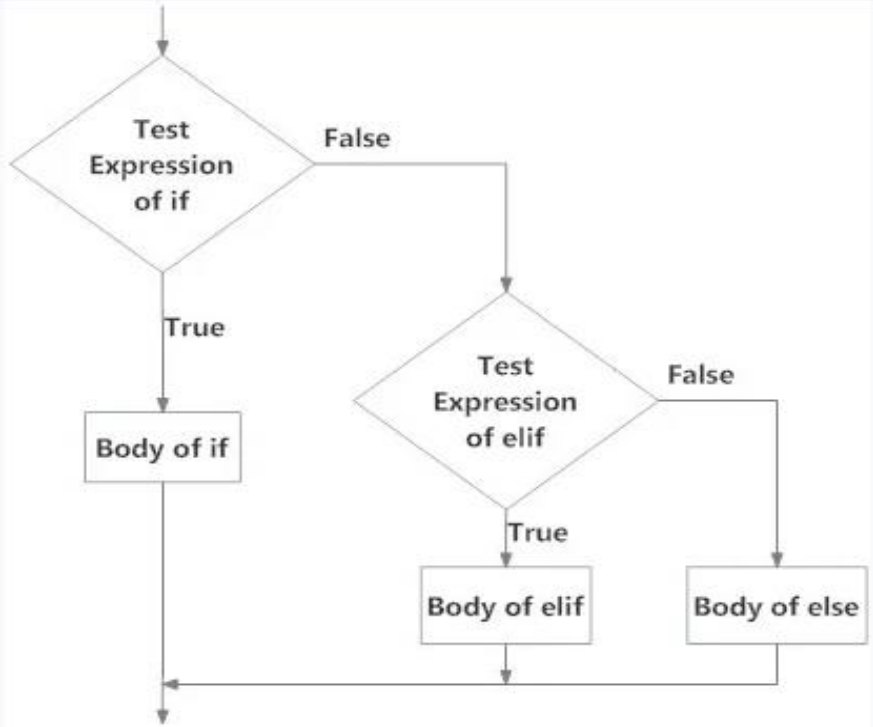


Fig: Operation of if...elif...else statement

Flowchart of if...elif....else statement in Python

Loops

Python programming language provides the following types of loops to handle looping requirements.

1. For loop
2. While loop

Python for loop

The 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.

Flow chart for loop

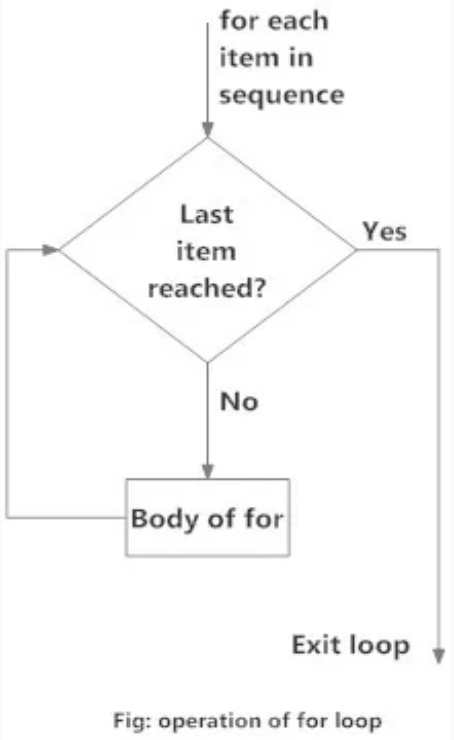


Fig: operation of for loop

Flowchart of for Loop in Python

While loop

The while loop in Python is used to iterate over a block of code as long as the test expression (condition) is true.

We generally use this loop when we don't know the number of times to iterate beforehand.

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

Flowchart

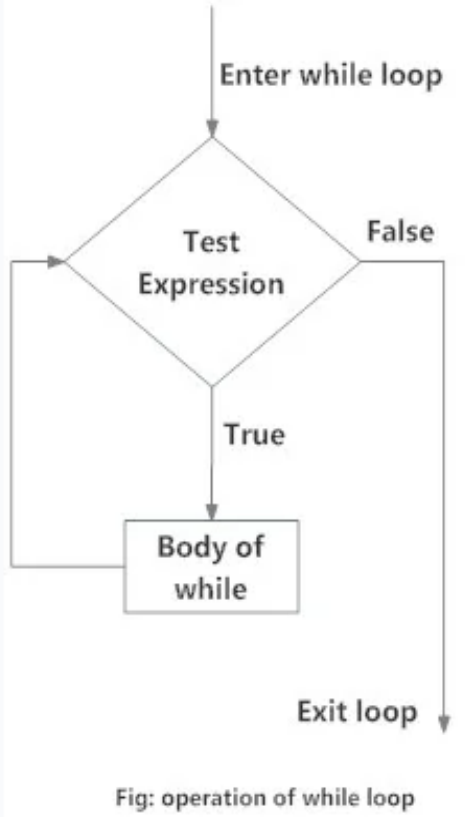


Fig: operation of while loop

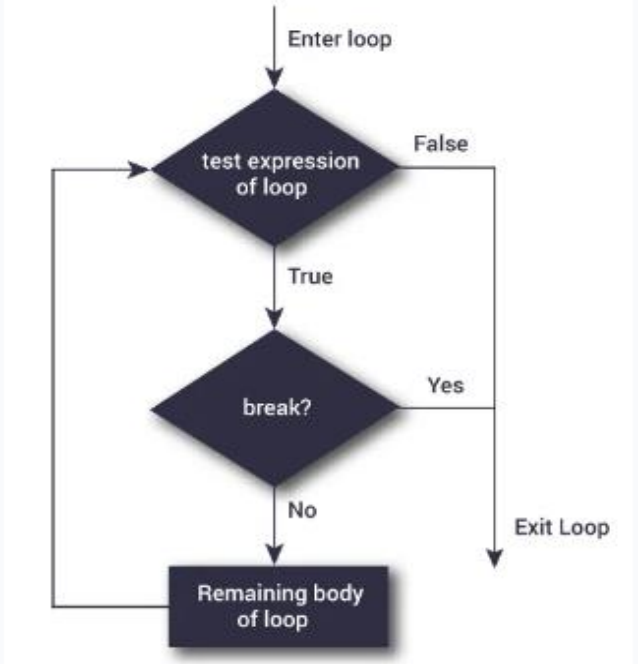
Flowchart for while loop in Python

Python break statement

The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.

If the break statement is inside a nested loop (loop inside another loop), the break statement will terminate the innermost loop.

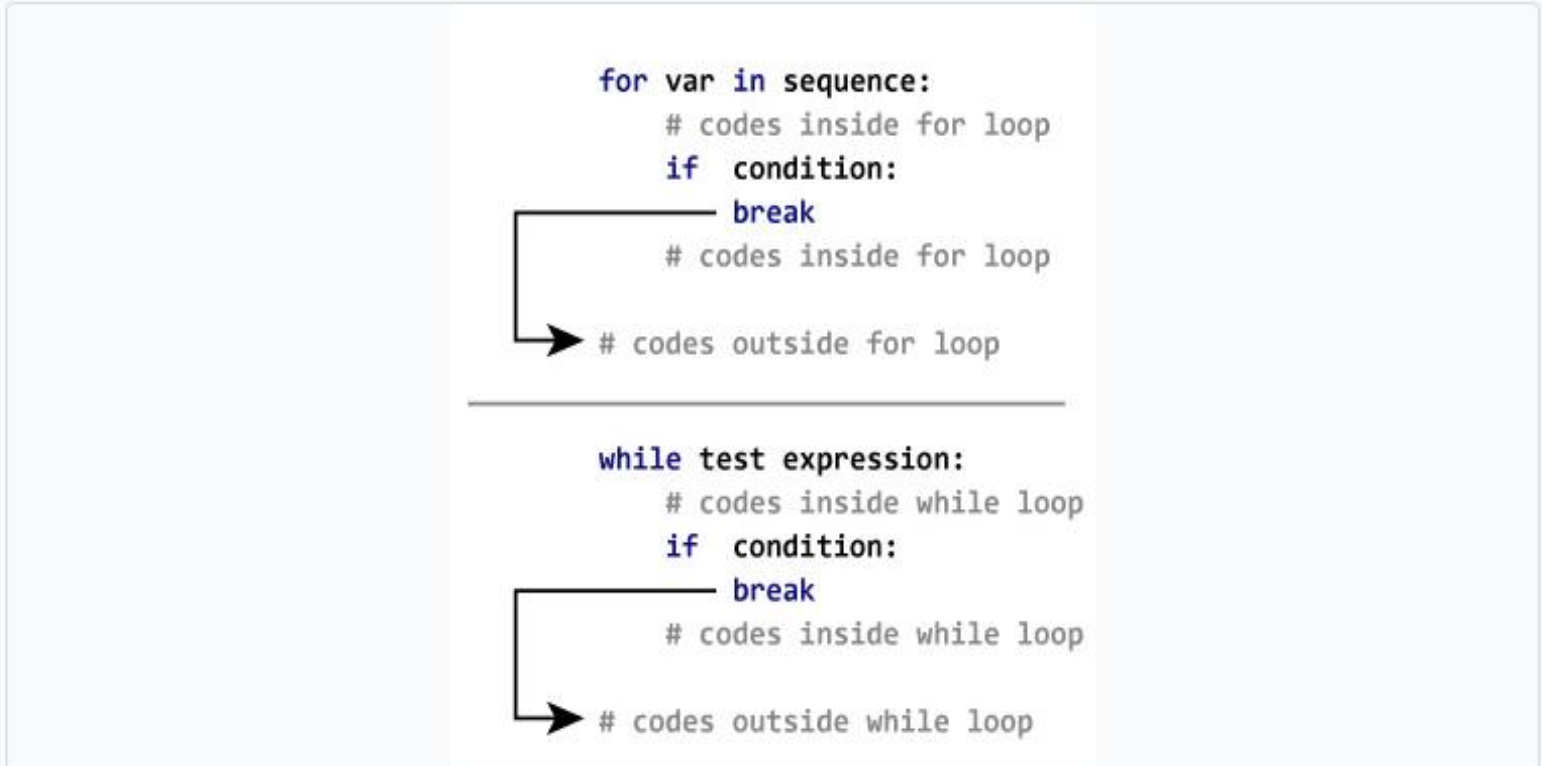
Flowchart



Flowchart of break statement in Python

DIGITAL REGENESYS

Working of break with for and while loop

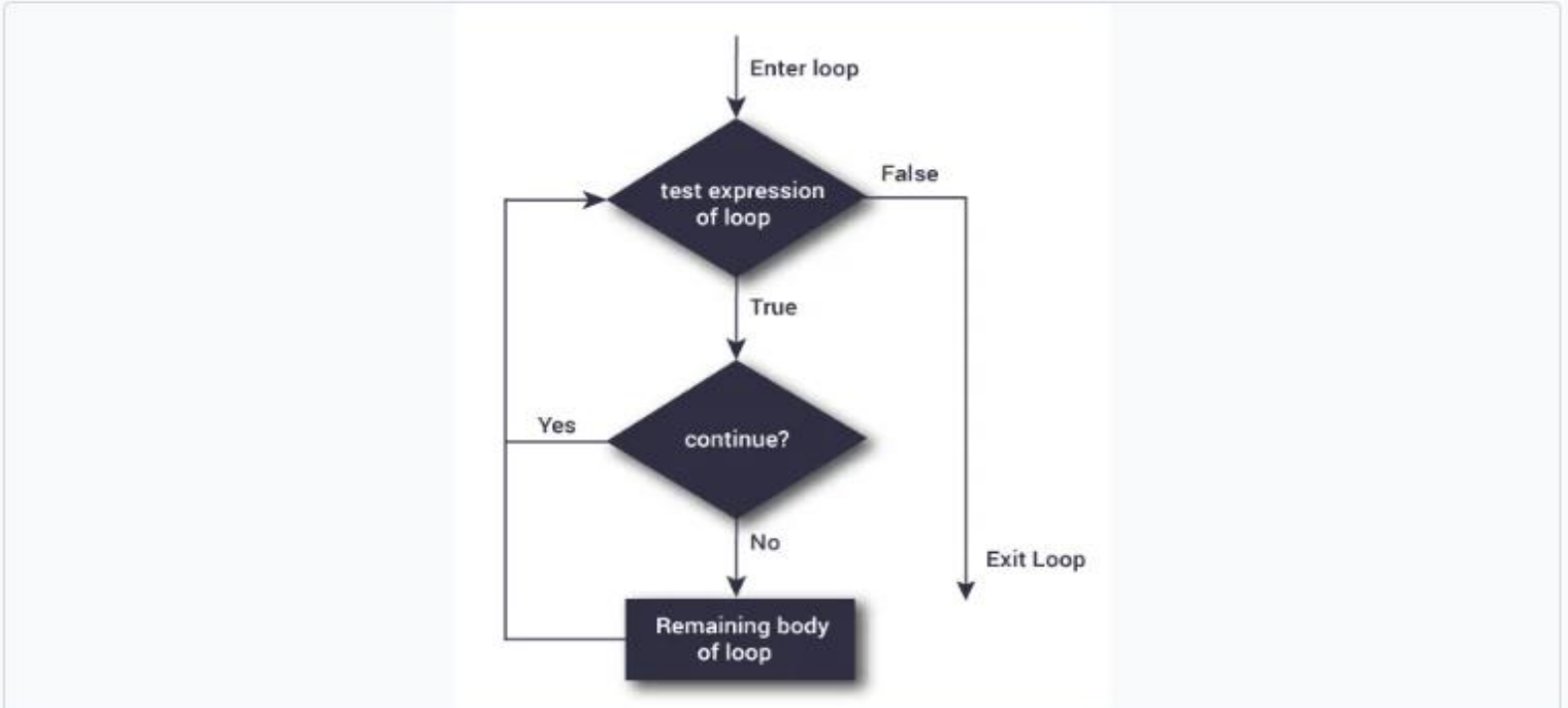


Working of the break statement

Python continue statement

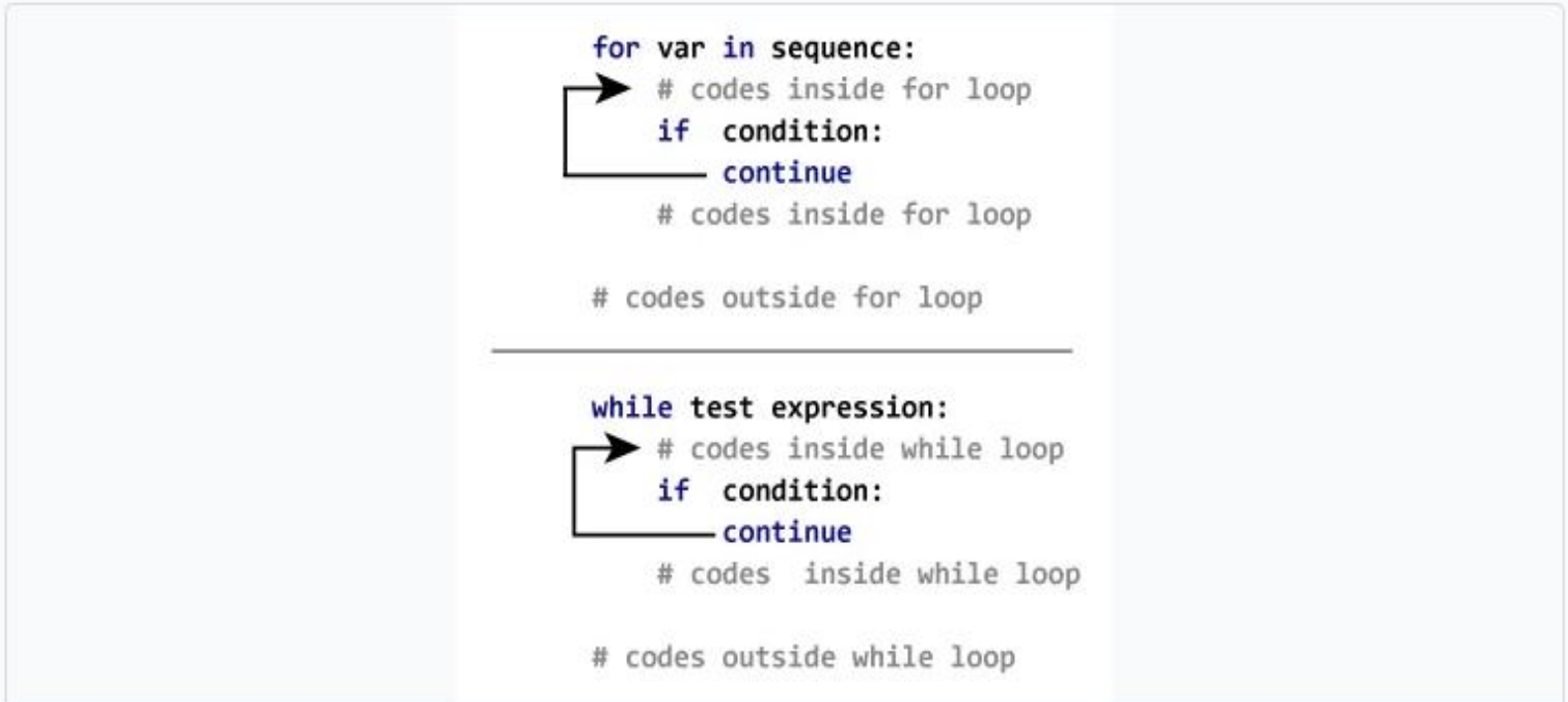
The continue statement is used to skip the rest of the code insi
de a loop for the current iteration only. Loop does not terminate
but continues on with the next iteration.

Flow chart



Flowchart of continue statement in Python

Working of continue with for and while loop



How continue statement works in python

Assignments:

- **Write a python code to print table of 5,6,7**
- **Write a python code to print the even numbers from 1 to 20**
- **Write a python code to print odd numbers from 1 to 20**
- **Write a python code to print reverse numbers from 50 to 1**

DIGITAL REGENESYS

Thank You

DIGITAL REGENESYS