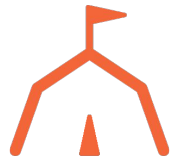
Loops

Loops

- The while loop
- The do loop
- The for loop



The For Loop

A for loop allows you to write a loop that needs to execute a specific number of times.

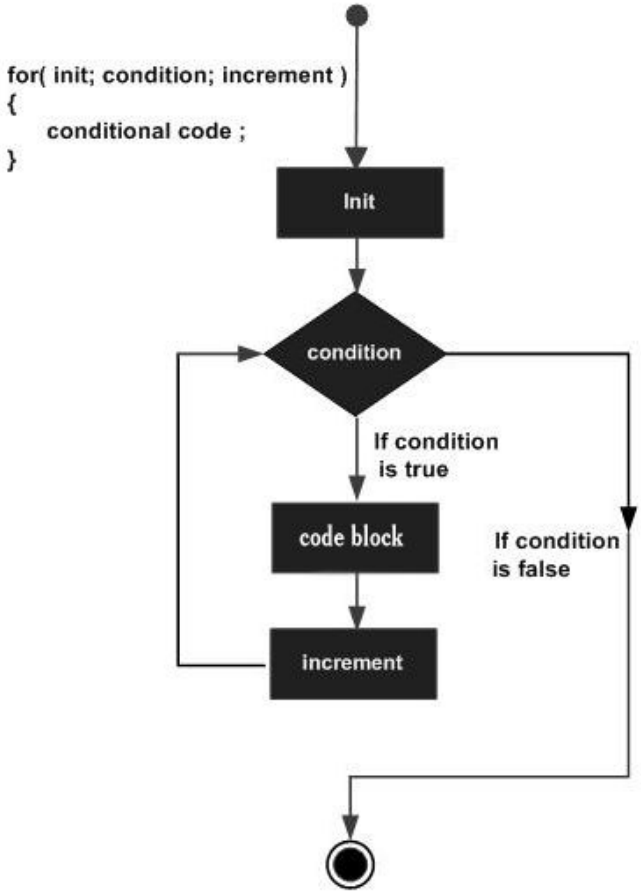
A for loop is useful when you know how many times a task is to be repeated.

```
for(initialization; Boolean_expression; update)
{
   //Statements
}
```

for loop

Used to iterate through the loop.

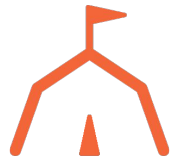
Useful when the number of times to loop is known.



The While Loop

A while loop statement repeatedly executes a target statement as long as a given condition is true.

```
while(Boolean_expression)
{
        //Statements
}
```

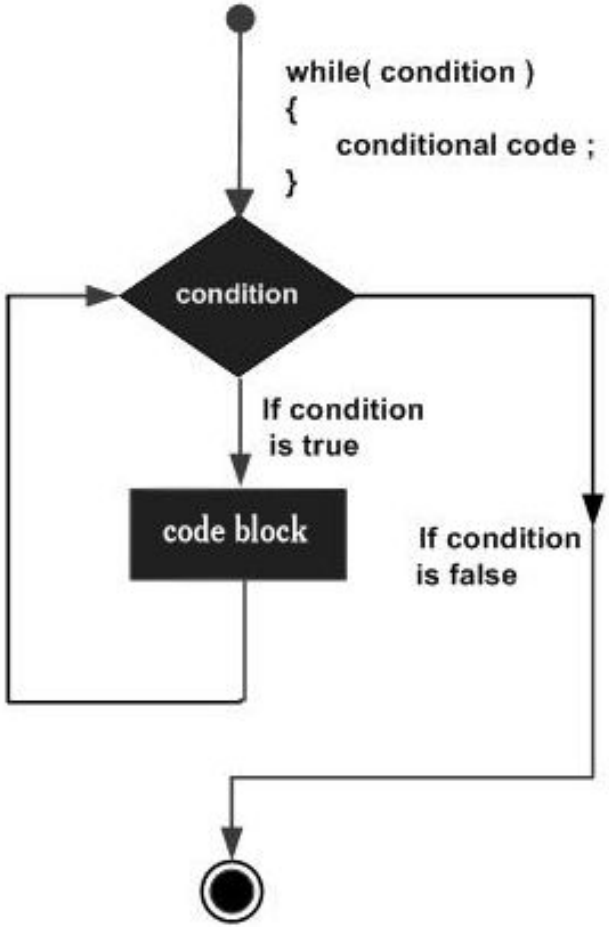


while loop

Loop while a condition is true.

May loop zero times since condition is checked before loop

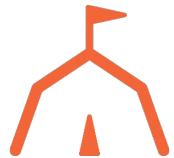
Somewhere in the code-block the condition must be made false at some point in the processing.



The Do Loop

A do loop is similar to a while loop, except that a do loop is guaranteed to execute at least one time.

```
do
{
        //Statements
}while(Boolean_expression);
```

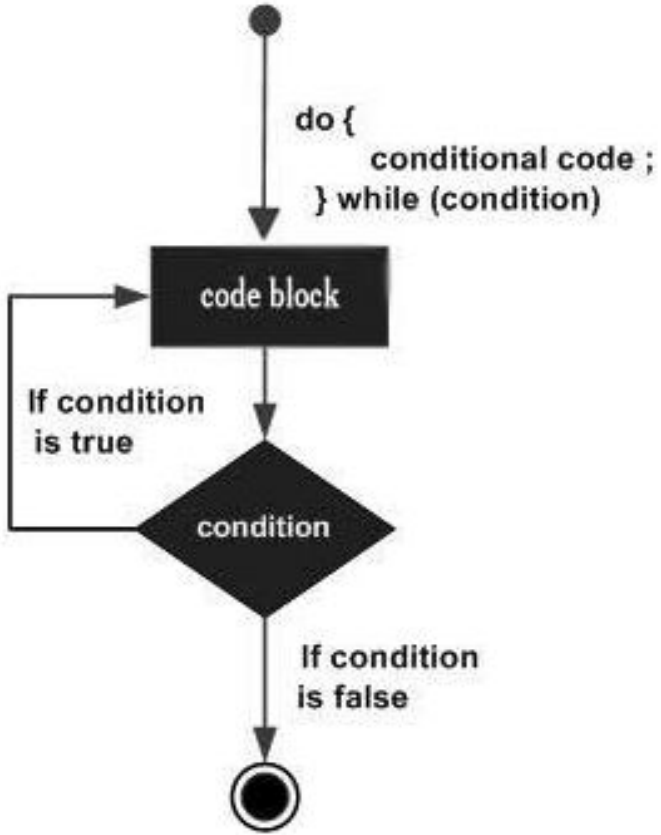


do-while loop

Loop while a condition is true.

Always loop at least once since condition is checked at end of loop

Somewhere in the code-block the condition must be made false at some point in the processing.



Control Statements

Break statement

The break statement is used to exit the current loop. In the case of multiple nested loops, a labeled break statement may be used to differentiate.

Control Statements

Continue statement

The continue statement is used to skip any remaining statements in the current loop and jump to the top of the current loop. A labeled continue statement may be used to jump to the top of a labeled loop.

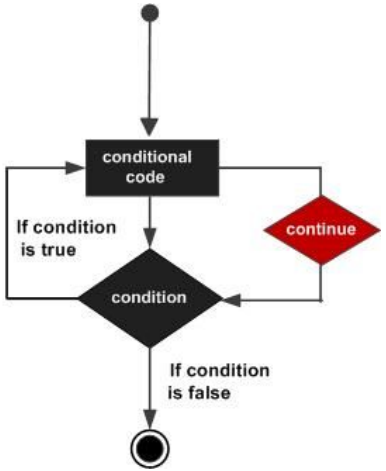
continue / break statement

continue will cause processing to skip to the end of the loop:

In a for loop: the increment

In a while / do while: the condition

break statement will immediately exit the loop.



Recap

- How to evaluate Boolean expressions.
- Know how to use logical and relational operators.
- Use for, while and do while loop.
- Comparing numeric variables.
- Comparing string variables.
- How to use continue and break statements.