

LUA - LOOPS

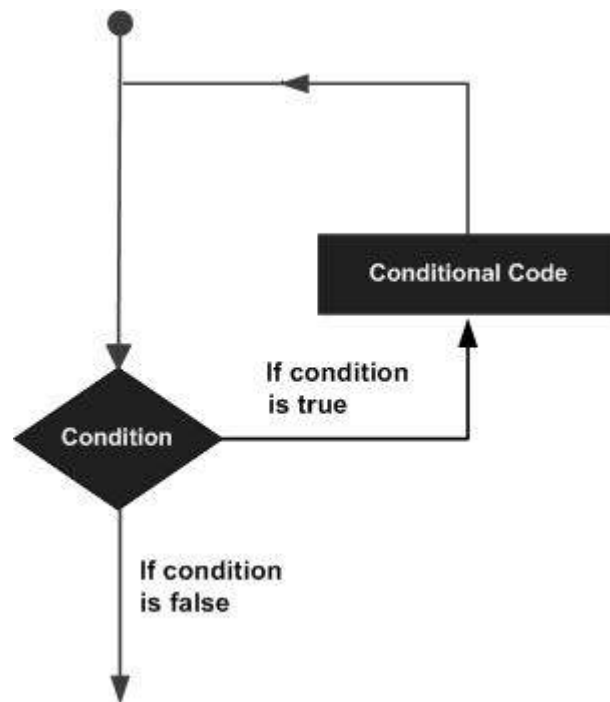
http://www.tutorialspoint.com/lua/lua_loops.htm

Copyright © tutorialspoint.com

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially – the first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times. Following is the general form of a loop statement in most of the programming languages –



Lua provides the following types of loops to handle looping requirements.

Loop Type	Description
while loop	Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
for loop	Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
repeat...until loop	Repeats the operation of group of statements till the until condition is met.
nested loops	You can use one or more loop inside any another <i>while</i> , <i>for</i> or <i>do..while</i> loop.

Loop Control Statement

Loop control statement changes execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Lua supports the following control statements.

Control Statement	Description
break statement	Terminates the loop and transfers execution to the statement immediately following the loop or switch.

The Infinite Loop

A loop becomes infinite loop if a condition never becomes false. The **while** loop is often used for this purpose. Since we directly give true for the condition, it keeps executing forever. We can use the break statement to break this loop.

```
while( true )  
do  
    print("This loop will run forever.")  
end
```