

# Infinite Loops

This lesson explains how infinite loops might arise in a while and for loop using an example.

## WE'LL COVER THE FOLLOWING ^

- What are Infinite Loops
- Example of Infinite loop

## What are Infinite Loops #

One common programming mistake is to create an **infinite** loop. An **infinite** loop refers to a loop, which under certain valid (or at least plausible) input, will never **exit**.

**Note:** Beginner programmers should be careful to examine all the *possible* inputs into a *loop* to ensure that for each such set of inputs, there is an **exit** condition that will eventually be reached.

*Compilers, debuggers, and other programming tools* can only help the programmer so far in detecting **infinite** loops.

In the fully general case, it is not possible to automatically detect an infinite loop. This is known as the **halting** problem. While the halting problem is not solvable in the fully general case, it is possible to determine whether a loop will **halt** for some *specific* cases.

## Example of Infinite loop #

Down below is an example of an *infinite* loop.

```
$a = 1;  
# the while condition will always be met as it will always return true  
while($a){  
    print "Infinite loop\n";  
}
```



```
print "Infinite loop\n";  
}
```

When we run the code above, it will print “**Infinite loop**” without stopping because the condition statement in the **while** loop will always resolve to **true**. There is no point at which the loop condition will evaluate to **false** hence we’ll get stuck in an *infinite* loop, and the code will execute forever. Similarly, the **for** loop given below runs infinite time without stopping.

```
for ($i = 0; $i < 1; ){ # as the condtion always met and never ends  
    print "Infinite loop"  
}
```



---

Now that you know all there is to know about the loops, let’s solve a quiz in the next lesson.