

Bash Until Loop

In this topic, we have defined how to use until loop statement in Bash Script.

The while loop is a great option to execute a set of commands when some condition evaluates to true. Sometimes, we need to execute a set of commands until a condition evaluates to true. In such cases, Bash until loop is useful.

Bash Until Loop in a bash scripting is used to execute a set of commands repeatedly based on the boolean result of an expression. The set of commands are executed only until the expression evaluates to true. It means that when the expression evaluates to false, a set of commands are executed iteratively. The loop is terminated as soon as the expression evaluates to true for the first time.

In short, the until loop is similar to the while loop but with a reverse concept.

Syntax

The syntax of until loop looks almost similar to the syntax of bash while loop. But there is a big difference in the functionalities of both. The syntax of bash until loop can be defined as:

1. until [expression];
2. do
3. command1
4. command2
5. ...
6.
7. commandN
8. done

If there are multiple conditions in the expression, then the syntax will be as follows:

1. until [[expression]];
2. do
3. command1
4. command2
5. ...
6.
7. commandN
8. done

Some of the key points of until loop are given below:

- The condition is checked before executing the commands.
- The commands are only executed if the condition evaluates to false.
- The loop is terminated as soon as the condition evaluates to true.
- The program control is transferred to the command that follows the 'done' keyword after the termination.

The while loop vs. the until loop

- The 'until loop' commands execute until a non-zero status is returned.
- The 'while loop' commands execute until a zero status is returned.
- The until loop contains property to be executed at least once.

Examples of Bash Until Loop

Following are some examples of bash until loop illustrating different scenarios to help you understand the usage and working of it:

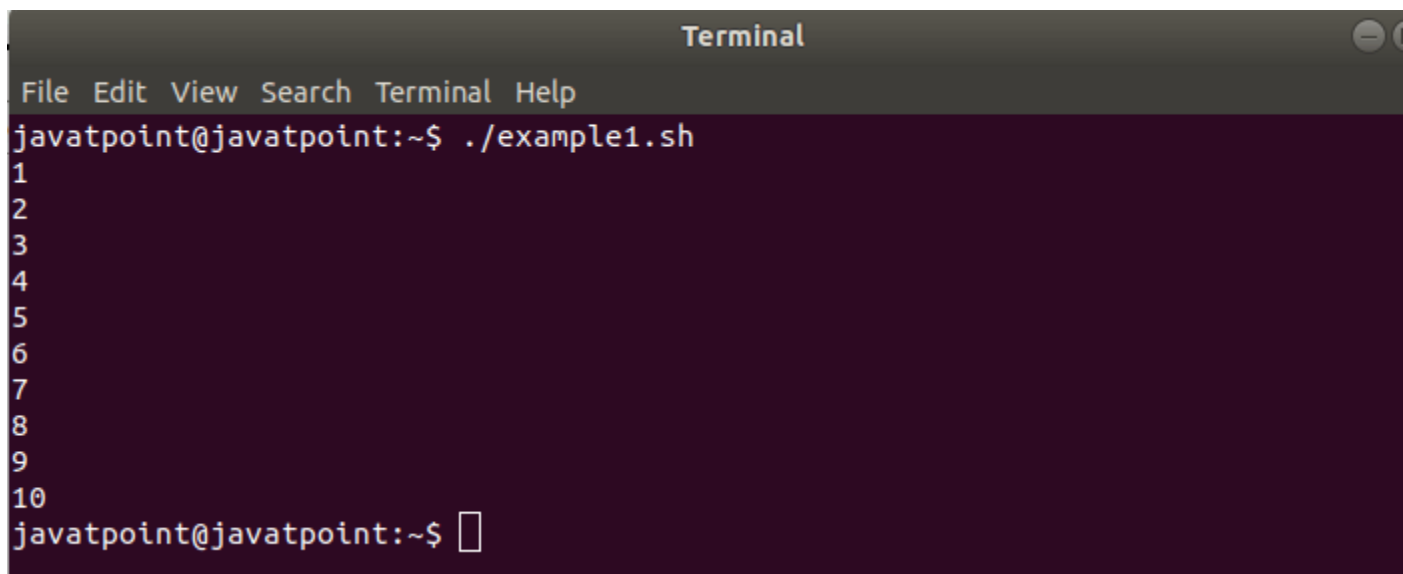
Until Loop with Single Condition

In this example, the until loop contains a single condition in expression. It is the basic example of until loop which will print series of numbers from 1 to 10:

Example

1. `#!/bin/bash`
2. `#Bash Until Loop example with a single condition`
- 3.
4. `i=1`
5. `until [$i -gt 10]`
6. `do`
7. `echo $i`
8. `((i++))`
9. `done`

Output

A screenshot of a macOS Terminal window. The title bar is dark gray with the word "Terminal" in white. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help" in light gray. The main area of the terminal has a dark purple background. It shows the command prompt "javatpoint@javatpoint:~\$./example1.sh" followed by the output "1", "2", "3", "4", "5", "6", "7", "8", "9", and "10" on separate lines. The prompt "javatpoint@javatpoint:~\$" is followed by a white cursor block.

```
Terminal
File Edit View Search Terminal Help
javatpoint@javatpoint:~$ ./example1.sh
1
2
3
4
5
6
7
8
9
10
javatpoint@javatpoint:~$
```

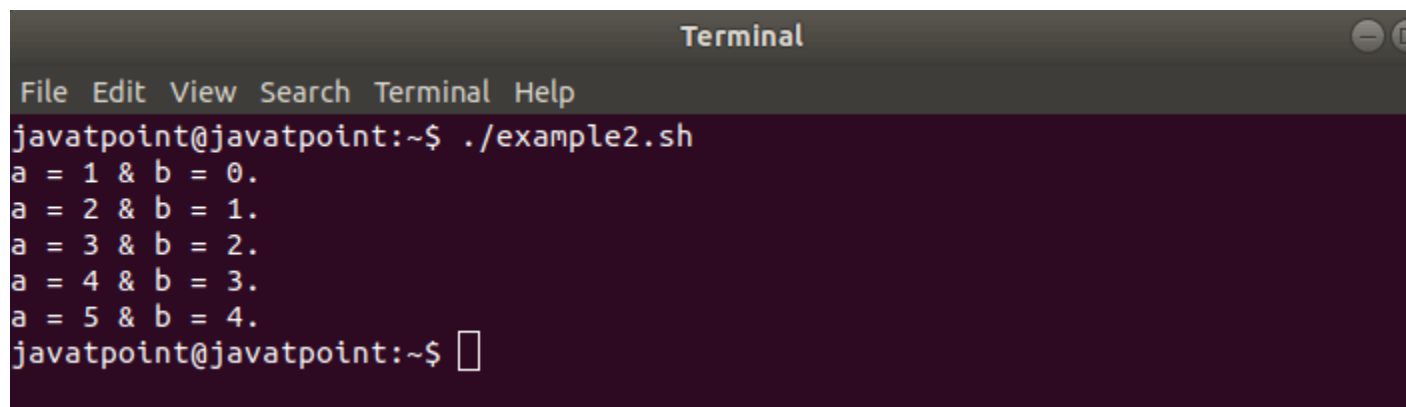
Until Loop with Multiple Conditions

Following is an example with multiple conditions in an expression:

Example

1. `#!/bin/bash`
2. `#Bash Until Loop example with multiple conditions`
- 3.
4. `max=5`
5. `a=1`
6. `b=0`
- 7.
8. `until [[$a -gt $max || $b -gt $max]];`
9. `do`
10. `echo "a = $a & b = $b."`
11. `((a++))`
12. `((b++))`
13. `done`

Output

A screenshot of a macOS Terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a user at the "javatpoint" prompt running a script named "example2.sh". The script's output consists of five lines: "a = 1 & b = 0.", "a = 2 & b = 1.", "a = 3 & b = 2.", "a = 4 & b = 3.", and "a = 5 & b = 4.". After the last line, the prompt returns to "javatpoint@javatpoint:~\$" with a cursor. The terminal background is dark purple, and the text is white.

```
Terminal
File Edit View Search Terminal Help
javatpoint@javatpoint:~$ ./example2.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
javatpoint@javatpoint:~$
```