

Bash Else If – elif

In this [Bash Tutorial](#), we will learn the syntax and usage of **Bash Else If** statement with example Bash Scripts.

Bash Else If

Bash Else If is kind of an extension to [Bash If Else](#) statement. In Bash else-if, there can be multiple elif blocks with a boolean expression for each of them. If elif if ladder appears like a conditional ladder.

Syntax of Bash Else IF – elif

Following is the syntax of **Else If** statement in Bash Shell Scripting.

```
if <expression>; then
    <commands>
elif <expression>; then
    <commands>
else
    <commands>
fi
```

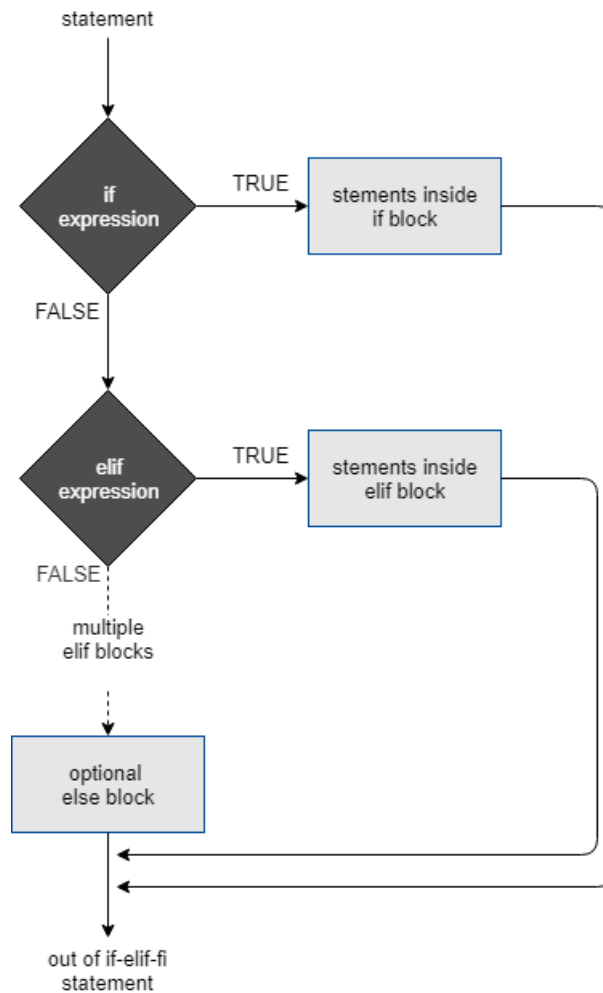
where

<expression>	Similar to Bash If statement, you may provide a single condition or multiple conditions after if keyword.
<commands>	Set of commands to be run when the <condition> is true.
elif	Else If

In this if-else-if ladder, the expressions are evaluated from top to bottom. Whenever an expression is evaluated to true, corresponding block of statements are executed and the control comes out of this if-else-if ladder.

If none of the expressions evaluate to true, the block of statements inside else block is executed. Also, note that the else block is optional.

Whenever a default action has to be performed when none of the if and else-if blocks get executed, define else block with the default action.



Example 1: Basic Bash Else If (elif)

In this example, we will look into two scenarios wherein one elif expression the condition evaluates to true and in the other to false.

Bash Script File

```
#!/bin/bash

n=2

if [ $n -eq 1 ]; then
    echo value of n is 1
elif [ $n -eq 2 ]; then
    echo value of n is 2
else
    echo value of n is other than 1 and 2
fi
```

Output

```
~/workspace/bash$ ./bash-elif-example
value of n is 2
```

In the first if expression, condition is false, so bash evaluates the expression for elif block. As the expression is

In the first expression, condition is false, so bash evaluates the expression for `elif` block. As the expression is true, the commands in the `elif` (else if) block are executed.

Example 2: Bash Else If with Multiple Conditions

In this example, we shall look into a scenario where there could be multiple conditions for `elif` (else if) expression.

Bash Script File

```
#!/bin/bash

n=2

if [ $n -eq 1 ]; then
    echo value of n is 1
elif [[ $n -eq 2 && $n -lt 5 ]]; then
    echo value of n is less than threshold
fi
```

Note : Else block is optional.

Output

```
~/workspace/bash$ ./bash-elif-example
value of n is less than threshold
```

Conclusion

Concluding this [Bash Tutorial](#), we have learned about the syntax and usage of Bash Else IF statement with example bash scripts.

Bash Shell Scripting

- ◆ [Bash Tutorial](#)
- ◆ [Bash Script Example](#)
- ◆ [Bash File Extension](#)
- ◆ [Bash Echo](#)
- ◆ [Bash Comments](#)
- ◆ [Bash Variable](#)
- ◆ [Bash Command Line Arguments](#)

◆ Bash Read User Input

◆ Bash Read Password

◆ Bash Date Format

◆ Bash Sleep

Operators

◆ Bash Arithmetic Operators

Conditional Statements

◆ Bash If

◆ Bash If Else

⇒ **Bash Else If**

◆ Bash Case

Loops

◆ Bash For Loop

◆ Bash While Loop

◆ Bash Until Loop

Strings

◆ Bash String Manipulation Examples

◆ Bash String Length

◆ Bash If String Equals

◆ Bash Split String

◆ Bash SubString

◆ Bash Concatenate String

◆ Bash Concatenate Variables to Strings

Functions

◆ Bash Function

◆ Bash Override Builtin Commands

Arrays

◆ Bash Array

Files

- ◆ Bash Write to File
- ◆ Bash Read File
- ◆ Bash Read File line by line
- ◆ Bash If File Exists
- ◆ Bash If File is Directory
- ◆ Bash If File is Readable

Bash Others

- ◆ Bash Check if variable is set