The if Statement

Conditional decision making, using an **if** statement, is a basic construct that any useful programming or scripting language must have.

When an **if** statement is used, the ensuing actions depend on the evaluation of specified conditions, such as:

* + - Numerical or string comparisons
    - Return value of a command (0 for success)
    - File existence or permissions.

In compact form, the syntax of an **if** statement is:

**if TEST-COMMANDS; then CONSEQUENT-COMMANDS; fi**

A more general definition is:

**if condition**  
**then**  
**statements**  
**else**  
**statements**  
**fi**

## Using the if Statement

n the following example, an **if** statement checks to see if a certain file exists, and if the file is found, it displays a message indicating success or failure:

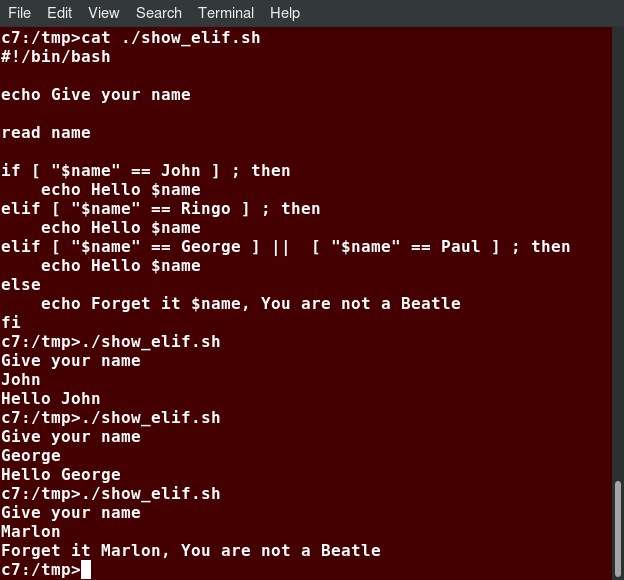
**if [ -f "$1" ]**  
**then**  
**echo file "$1 exists"**  
**else**  
**echo file "$1" does not exist**  
**fi**

We really should also check first that there is an argument passed to the script (**$1**) and abort if not.

Notice the use of the square brackets (**[]**) to delineate the test condition. There are many other kinds of tests you can perform, such as checking whether two numbers are equal to, greater than, or less than each other and make a decision accordingly; we will discuss these other tests.

In modern scripts, you may see doubled brackets as in **[[ -f /etc/passwd ]]**. This is not an error. It is never wrong to do so and it avoids some subtle problems, such as referring to an empty environment variable without surrounding it in double quotes; we will not talk about this here.

**if [ sometest ] ; then**  
**echo Passed test1**  
**elif [ somothertest ] ; then**  
**echo Passed test2**  
**fi**



# Testing for Files

Bash provides a set of file conditionals, that can be used with the **if** statement, including those in the table.

You can use the **if**statement to test for file attributes, such as:

* + - File or directory existence
    - Read or write permission
    - Executable permission.

For example, in the following example:  
  
**if [ -x /etc/passwd ] ; then**  
**ACTION**  
**fi**

the **if** statement checks if the file **/etc/passwd** is executable, which it is not. Note the very common practice of putting:

**; then**

on the same line as the **if** statement.

You can view the full list of file conditions typing:

**man 1 test**.

|  |  |
| --- | --- |
| **Condition** | **Meaning** |
| **-e file** | Checks if the file exists. |
| **-d file** | Checks if the file is a directory. |
| **-f file** | Checks if the file is a regular file (i.e. not a symbolic link, device node, directory, etc.) |
| **-s file** | Checks if the file is of non-zero size. |
| **-g file** | Checks if the file has **sgid** set. |
| **-u file** | Checks if the file has **suid** set. |
| **-r file** | Checks if the file is readable. |
| **-w file** | Checks if the file is writable. |
| **-x file** | Checks if the file is executable. |