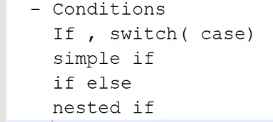
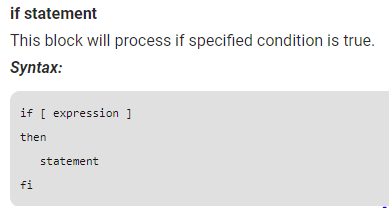
* $If we want to validate the values, we can use loops as below
* If
* While
* For

**Conditions:**

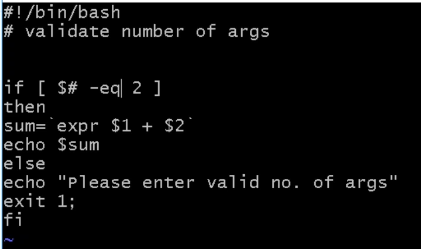
* We don’t have datatypes in shell scripting



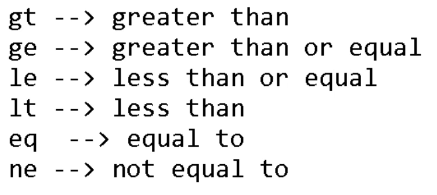
**If:**



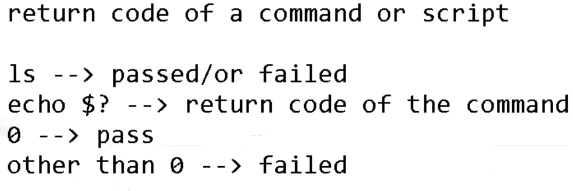
**If else:**



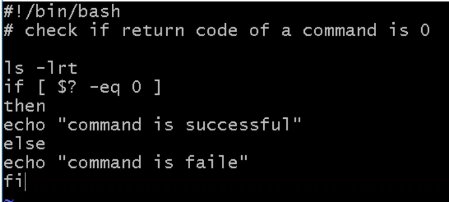
* If we need to close with fi



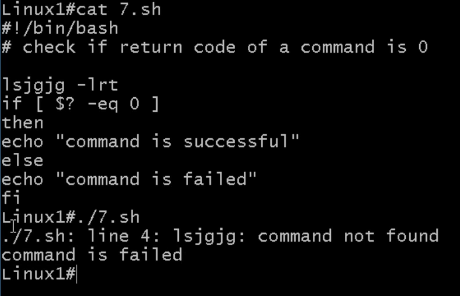
* Now, user has to give 2 values along with command. Otherwise the script fails
* Even though if we don’t use if condition but use only expr, then also the script fails if user gives unmatched values
* But if we want to display a proper message, then use as above
* After if we need to give the syntax condition in [ ]
* Exit 1 is the return code of script



* $? Displays the return code of previous command in Linux
* If the command is successful, then the return code is 0
* Other than 0 means failed
* We can use this feature in scripting if we have many commands in that as below



* As above, it runs ls -lrt, and if return code is 0 then it prints as command is successful
* If return code is not zero, then it prints as failed



* As above, if we have given any other command which gets failed. It prints as command failed

**Redirection operators:**

**>**

**>>**

**Example:**

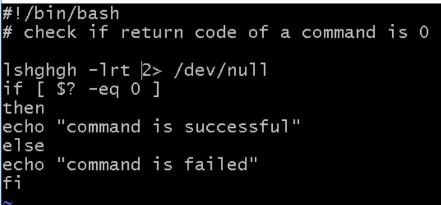
**Ls -ltr > file1**

As above, ls -ltr command redirects to file1 and file1 will have the output of ls -ltr inside it. If the file is already existing, then it will be overwritten

* **Ls -ltr >> file1**

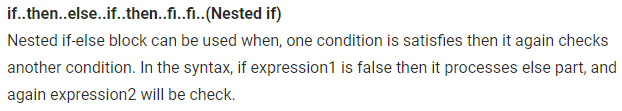
As above, if we use >> append operator. It won’t overwrite the existing file, but it will append to the old data inside file

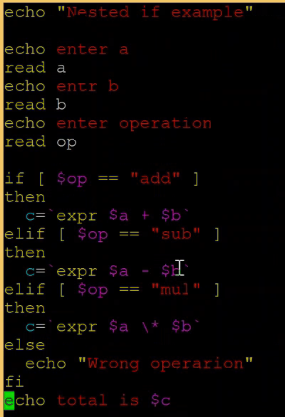
* This method we can use it in script. So, when we run the script. The output won’t be displayed on screen. It will be stored in a file
* If command is successful, then the output will be stored in a file. But if its failed then the error will be displayed on screen. If we don’t want, then we need to mention as below



* We need to give 2> to redirect the errors to a file

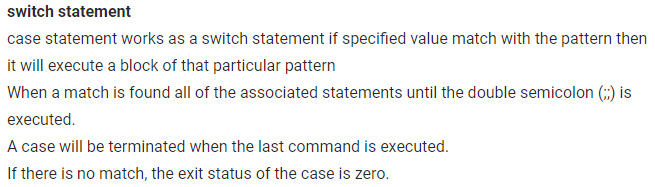
**Nested if:**



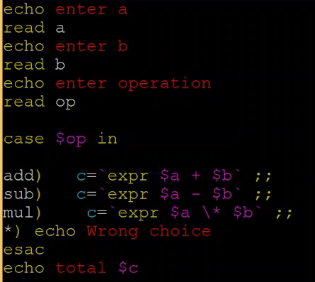


* Simple if includes only if and else. But nested if will be as above
* We can pass many conditions as above

**If switch:**



* If switch (case) will be useful as below



* Case ends with esac (reverse of case)