Variable

Variable	Meaning
$0	Name of the current script
$1 - $9	Positional parameters #1 - #9
${10}	Positional parameters #10
$#	Number of positional parameters
"$*"	All the positional parameters as one single word
"$@"	All the positional parameters as separated strings
${#*} or ${#@}	Number of command line parameters passed to script
$?	Return value
$$	Process ID of the script
$-	Flags passed to Script
$_	Last argument of previous command
$!	Process ID of Last Job in background

Miscellaneous Constructs

Expression	Meaning
if [...] then --- --- fi	If [value] is true, then do ---
if [[...]] then --- fi	If [[value]] is true, then do --- (extended condition)
while [...] do --- done	while [value] is true, do ---
((...))	Arithmetic operation
Be very careful not to forget the space in the brackets!	

Bash Operators

Operator for Variables	Meaning	Operator for Strings
-eq	Equal to	==
-ne	Not equal to	!=
-lt	Less than	\<
-le	Less than or equal to	
-gt	Greater than	\>
-ge	Greater than or equal to	
	String is empty	-z
	String is not empty	-n
/ Not needed if in [[double-bracket]]		

Operators between ((double parentheses))

Arithemetic Operator	Meaning
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Generate a random number between 0 and 10

```
(( ( RANDOM % 10 ) + 1 ))
```

File Operator

Operator	Meaning
-e	File Exists
-f	file is a regular file
-d	file is a directory
-h or -L	file is a symbolic link
-b	File is a block device
-N	File modified since it was last read
-O	You own the file
-G	Group ID of the file is the same as yours
!	Reverse the sense of the above tests
-s	Size of the file is not 0
-r	File has read permission
-w	File has write permission
-x	File has execute permission
-nt	Is newer than
-ot	Is older than