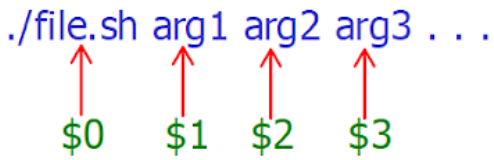
6. Shell Scripting – Command Line Arguments

Command Line Arguments are inputs to the script passed on the command line along with script name while executing the script. This is another way of passing inputs to the script like in case of indirection.



- Method of execution of script can be either using bash command or using ./exectable_file_name.
- A number of arguments can be passed through command line depending on the requirement.

Command Line Argument formats:

$@	→	**to get all the command line arguments**
$#	→	**total number of command line arguments**
$0	→	**to access filename**
$i	→	**i^{th} command line argument**

- Command line arguments can be accessed using the order in which they are entered on the command line.
- $# will be zero by default if no command line arguments are passed through command line
- To avoid logical errors, always check for required number of arguments in the beginning of scripts that access command line arguments as given in example below:

```
if [ $# -ge 2 ]
then
        #logic
else
        #error
        #logic
fi
```

Here, condition is checked so that the execution of logic will be continued only if minimum of 2 command line arguments are passed to the script. If the required number of arguments are not passed, then error or logic in else part of if-block should be executed.

Command line arguments with array formats:

The command line arguments can be converted to elements of array using the following format and even command line argument formats can be used along with array formats to access array:

array_name=($@)

The different methods to access command line arguments stored as elements of array:

$@ → **to get all command line arguments**

${array_name[@]} → **to get all command line arguments**

${array_name[*]} → **to get all command line arguments**

${array_name} → **to access 1st command line argument**

$1 → **to access 1st command line argument**

${array_name[i]} → **to access ith command line argument**

${array_name[-1]} → **to access last command line argument**

${array_name[$# - 1]} → **to access last command line argument**

$# → **to get total number of arguments**

${#array_name[@]} → **to get total number of arguments**

${#array_name[*]} → **to get total number of arguments**

${#array_name} → **to get length of 1st command line argument**

${#array_name[i]} → **to get length of ith command line argument**

${#array_name[-1]} → **to get length of last command line argument**

${#array_name[$# - 1]} → **to get length of last command line argument**

${array_name[${#array_name[@]} - 1]}

→ **to access last command line argument**

${#array_name[${#array_name[@]} - 1]}

→ **to get length of last command line argument**