Bash Function

Last updated by | Shakibe Hasan | Jun 9, 2023 at 6:48 PM GMT+6

A Bash function can be defined as a set of commands which can be called several times within bash script.

Following are some key points about bash functions:

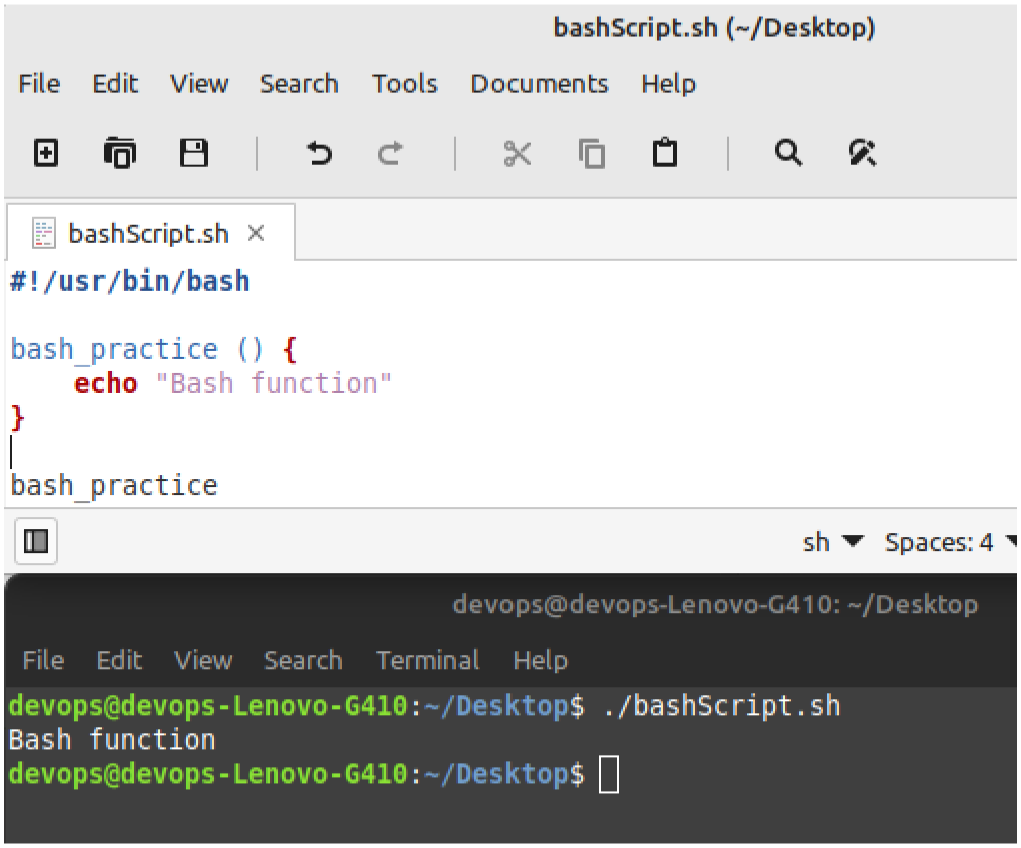
1. A function has to be declared in the shell script before we can use it.
2. Arguments can be passed to the functions and accessed inside the function as $1, $2, etc.
3. Local variables can be assigned within the function, and the scope of such variables will only be that particular function.
4. Built-in commands of Bash shell can be overridden using functions.

Function Syntax

function_name () {
echo"Most popular and widely used way"
}

or

function_name () { echo "Single line method"; }


bashScript.sh (~/Desktop)
File Edit View Search Tools Documents Help
bashScript.sh
sh Spaces: 4
devops@devops-Lenovo-G410: ~/Desktop
File Edit View Search Terminal Help

```
#!/usr/bin/bash

bash_practice () {
    echo "Bash function"
}

bash_practice
```

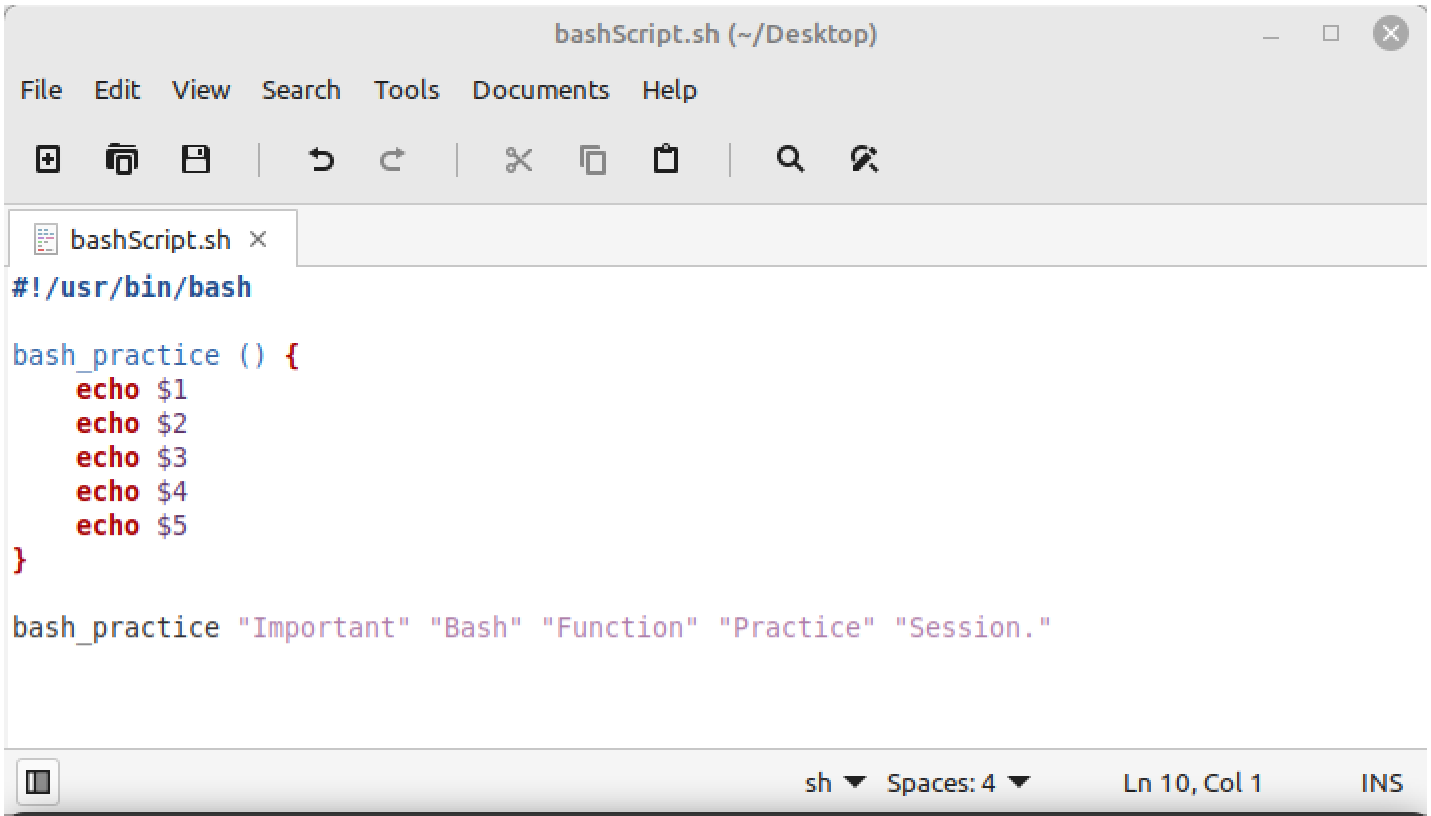
```
devops@devops-Lenovo-G410:~/Desktop$ ./bashScript.sh
Bash function
devops@devops-Lenovo-G410:~/Desktop$ 
```

Passing Arguments

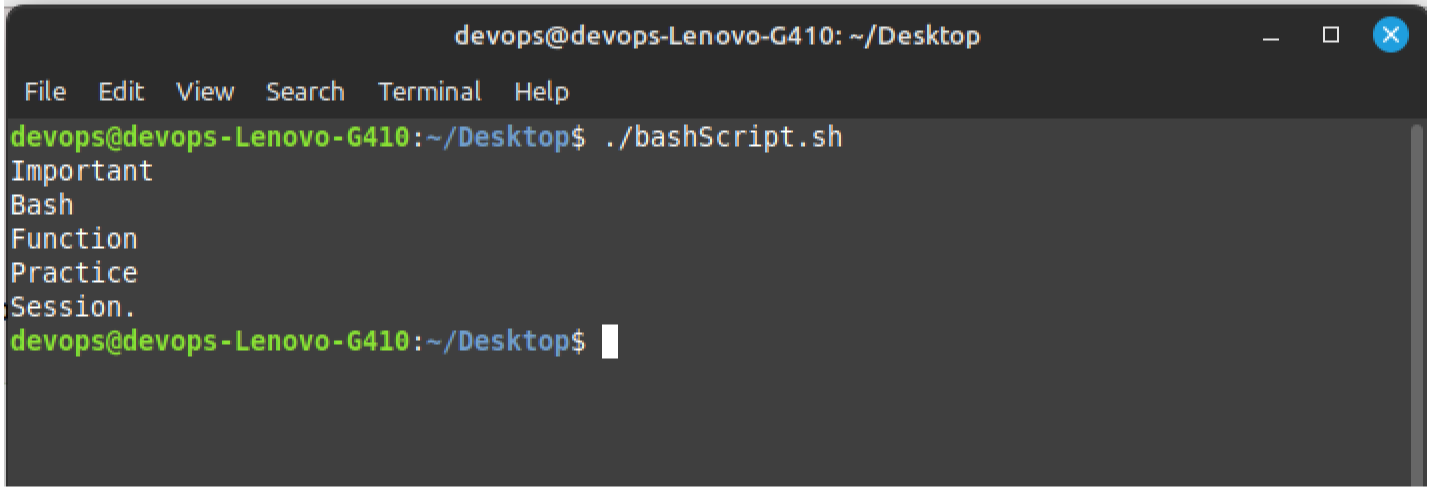
Following are some key points about passing arguments to the bash functions:

- The given arguments are accessed as $1, $2, $3 ... $n, corresponding to the position of the arguments after the function's name.

- The $0 variable is kept reserved for the function's name.
- The $# variable is used to hold the number of positional argument/ parameter given to the function.
- The $* and $@ variables are used to hold all the arguments/ parameters given to the function.
- When *"), it expands to a single string separated by the space. For example, "$1 $2 $n etc".
- When @"), it expands to the separate string. For example, "$1" " n" etc.
- When $* and $# are not used with the double quotes, they both are the same.


bashScript.sh (~/Desktop)
File Edit View Search Tools Documents Help
bashScript.sh
sh Spaces: 4 Ln 10, Col 1 INS

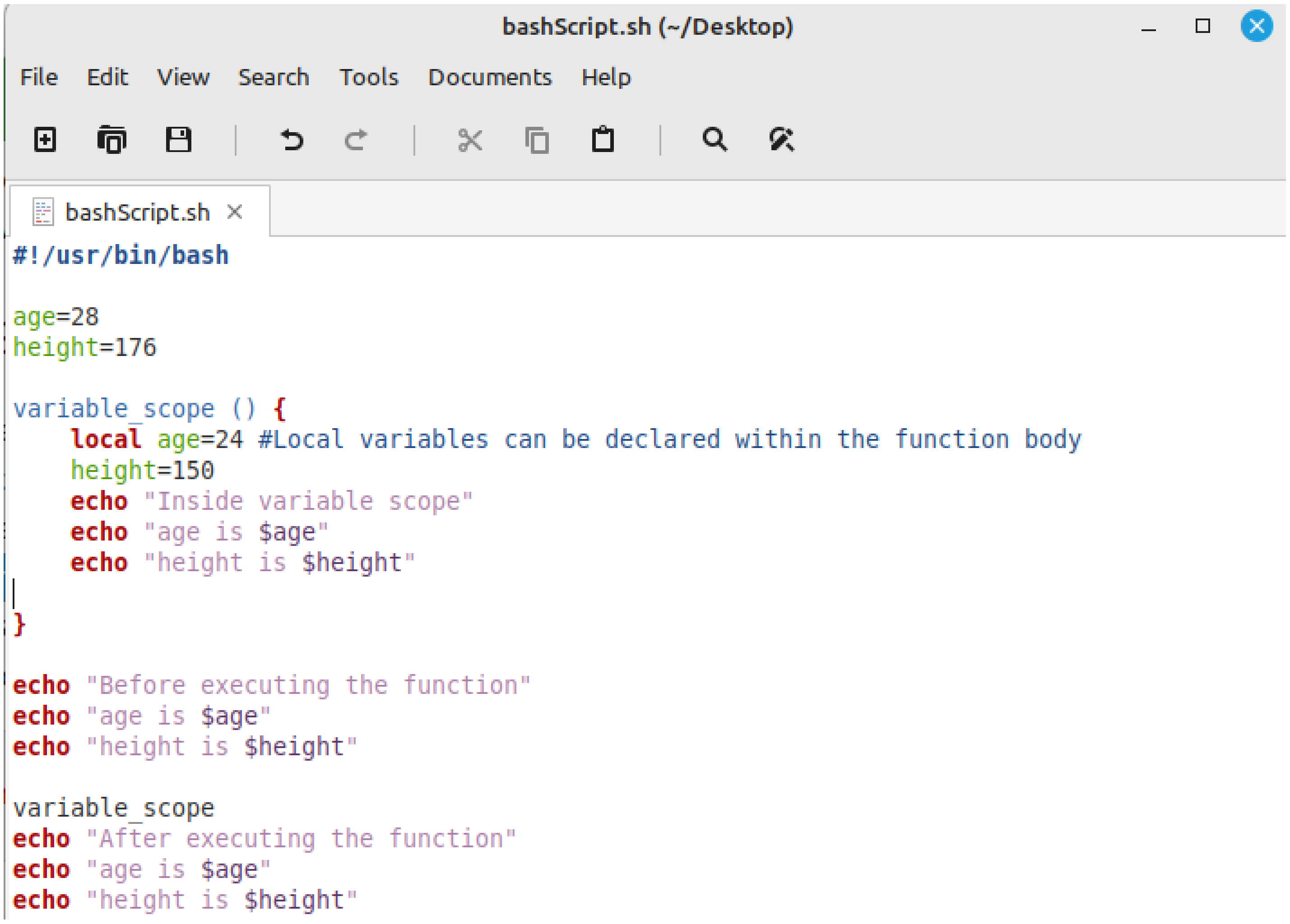
```
#!/usr/bin/bash

bash_practice () {
    echo $1
    echo $2
    echo $3
    echo $4
    echo $5
}

bash_practice "Important" "Bash" "Function" "Practice" "Session."
```


devops@devops-Lenovo-G410: ~/Desktop
File Edit View Search Terminal Help

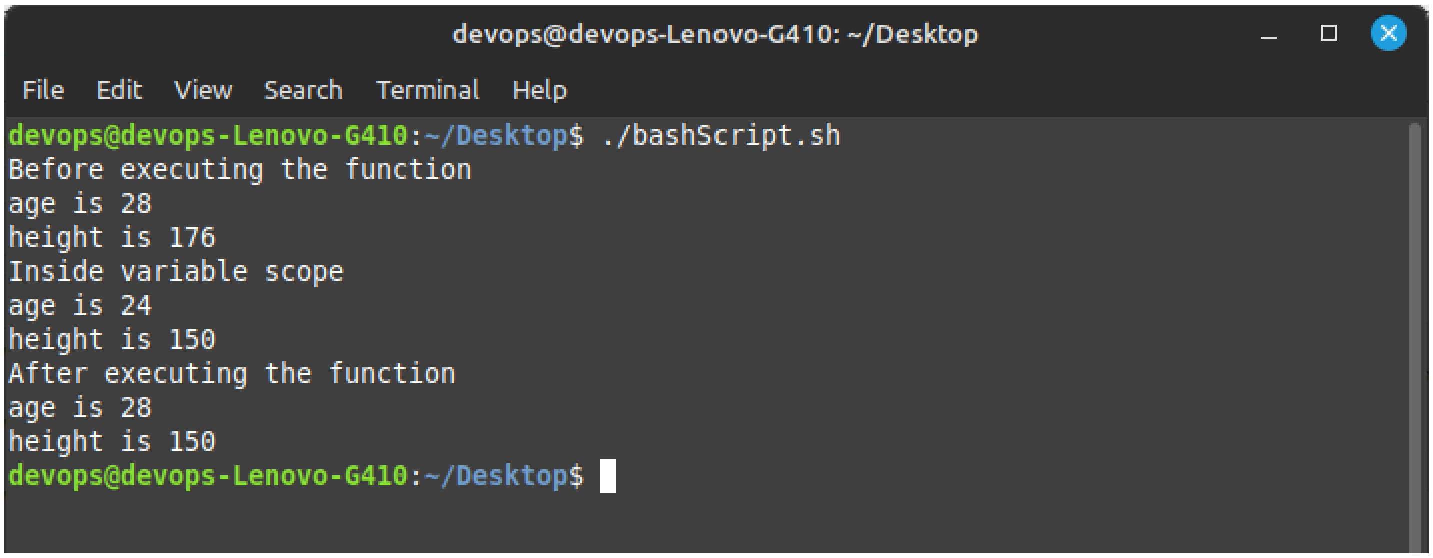
```
devops@devops-Lenovo-G410:~/Desktop$ ./bashScript.sh
Important
Bash
Function
Practice
Session.
devops@devops-Lenovo-G410:~/Desktop$
```

Variable Scope

- Global variable
- Local variable


bashScript.sh (~/Desktop)
File Edit View Search Tools Documents Help
bashScript.sh

```
#!/usr/bin/bash

age=28
height=176

variable_scope () {
    local age=24 #Local variables can be declared within the function body
    height=150
    echo "Inside variable scope"
    echo "age is $age"
    echo "height is $height"

}

echo "Before executing the function"
echo "age is $age"
echo "height is $height"

variable_scope
echo "After executing the function"
echo "age is $age"
echo "height is $height"
```


devops@devops-Lenovo-G410: ~/Desktop
File Edit View Search Terminal Help

```
devops@devops-Lenovo-G410:~/Desktop$ ./bashScript.sh
Before executing the function
age is 28
height is 176
Inside variable scope
age is 24
height is 150
After executing the function
age is 28
height is 150
devops@devops-Lenovo-G410:~/Desktop$
```

Ref: javatpoint.com