

Bash Functions – Syntax & Examples

Bash Functions

Bash Functions – In this [Bash Tutorial](#), we shall learn about functions in Bash Shell Scripting with the help of syntax and examples.

About Bash Functions

- Function has to be defined in the shell script first, before you can use it.
- Arguments could be passed to functions and accessed inside the function as \$1, \$2 etc.
- Local Variables could be declared inside the function and the scope of such local variables is only that function.
- Using functions, you may [override builtin commands](#) of Bash Shell.

Syntax of function

Any of the following two syntax could be used for defining a function in bash shell scripting.

```
function <function_name> {  
    # function body  
}
```

```
<function_name>() {  
    # function body  
}
```

<function_name>	Name of the function. Any word consisting only of alphanumeric characters and under?scores, and beginning with an alphabetic character or an under?score, can be used as a function name.
-----------------	---

Bash Function – Example

In the following example, we will create a function named `sampleFunction`, and call it.

Bash Script File

```
#!/bin/bash  
  
# bash function example
```

```
sampleFunction () {  
    echo Hello from Sample Function.  
}  
  
sampleFunction
```

Output

```
~$ ./bash-function  
Hello from Sample Function.
```

Bash Function – With function keyword

In this example, we will use function keyword to define a function.

Bash Script File

```
#!/bin/bash  
  
# bash function example  
function sampleFunction {  
    echo This is another way to define function in bash scripting.  
}  
  
sampleFunction
```

Output

```
~$ ./bash-function-2  
This is another way to define function in bash scripting.
```

Bash Function with Arguments

In this example, we shall learn to pass arguments to functions, and access the arguments inside the function.

Bash Script File

```
#!/bin/bash  
  
# bash function example with arguments  
function functionWithArgs {  
    echo $1 : $2 in this $3  
}  
  
functionWithArgs "`date +"[%m-%d %H:%M:%S]"` "Learn Functions" "Bash Tutorial"
```

Output

```
$ ./bash-function-arguments  
[11-21 19:30:21] : Learn Functions in this Bash Tutorial
```

Conclusion

In this [Bash Tutorial](#), we have learned about Bash Functions, how to define them, with examples scripts.

Bash Shell Scripting

- ◆ [Bash Tutorial](#)
- ◆ [Bash Script Example](#)
- ◆ [Bash File Extension](#)
- ◆ [Bash Echo](#)
- ◆ [Bash Comments](#)
- ◆ [Bash Variable](#)
- ◆ [Bash Command Line Arguments](#)
- ◆ [Bash Read User Input](#)
- ◆ [Bash Read Password](#)
- ◆ [Bash Date Format](#)
- ◆ [Bash Sleep](#)

Operators

- ◆ [Bash Arithmetic Operators](#)

Conditional Statements

- ◆ [Bash If](#)
- ◆ [Bash If Else](#)
- ◆ [Bash Else If](#)
- ◆ [Bash Case](#)

Loops

- ◆ [Bash For Loop](#)
- ◆ [Bash While Loop](#)
- ◆ [Bash Until Loop](#)

Strings

- ◆ Bash String Manipulation Examples
- ◆ Bash String Length
- ◆ Bash If String Equals
- ◆ Bash Split String
- ◆ Bash SubString
- ◆ Bash Concatenate String
- ◆ Bash Concatenate Variables to Strings

Functions

- ⇒ Bash Function
- ◆ Bash Override Builtin Commands

Arrays

- ◆ Bash Array

Files

- ◆ Bash Write to File
- ◆ Bash Read File
- ◆ Bash Read File line by line
- ◆ Bash If File Exists
- ◆ Bash If File is Directory
- ◆ Bash If File is Readable

Bash Others

- ◆ Bash Check if variable is set