Functions

Function syntax

```
function functionName()
{
  # some code goes here...
}
```

Function name

We follow the given rules when naming functions.

- Name must start with letters (a-z or A-Z) or underscore _
- After that we can use letters (a-z and A-Z), underscore _ and digits (0-9)
- Don't put space in the function name
- Don't use special keywords like echo, printf etc to name your function

```bash
#!/bin/bash

# create a function
function greetings()
{
  echo "Hello World"
}

# call the function
greetings
```

```bash
#!/bin/bash

# create a function
greetings() {
  echo "Hello $1"
}

# call the function
greetings "Yusuf Shakeel"
```

Exercise:

Write a Shell Script to print greetings message using the name entered by the user

Write a shell script to find Simple Interest based on input from user

```
# simple interest function
simpleInterest () {
  p="$1"
  r="$2"
  t="$3"
  si=`expr "($p * $r * $t)/100" | bc -l`
}

# take user input
printf "Enter Principal: "
read pr

printf "Enter Rate [0-100]: "
read rt

printf "Enter Time [in years]: "
read tm

# call function
simpleInterest $pr $rt $tm
```

File Operators

-e To check if the file exists.

-r To check if the file is readable.

-w To check if the file is writable.

-x To check if the file is executable.

-s To check if the file size is greater than 0.

-d To check if the file is a directory.

```
#!/bin/bash

file=" "

# check
if [ -e $file ]
then
  echo "File exists!"
else
  echo "File does not exists!"
fi
```

Write a Shell Script to check if a file is readable, writable and executable

Command Line Arguments

The `$0` variable

This holds the name of the script.

The `$1` `$2` ... `$N` variables

These variables hold the arguments provided to the script.

The `$#` variable

This variable hold the total number of arguments passed to the script.

The `$@` and `$*` variables

They both holds the list of arguments provided to the script.

```
#!/bin/sh

# display the file name
echo "The name of the script file is $0"
# display total number of arguments passed to the script
echo "Total number of arguments passed to the script = $#"
# display all the arguments using for loop
if [ $# -gt 0 ]
then

  echo "List of arguments:"
  for arg in $@
  do
    echo "$arg"
  done
else
   echo "No argument provided to the script."

fi
```