Variables:

- We can use any name to create a variable so that it contains only letters (a to z or A to Z), numbers (0 to 9) or the underscore character (_). For example:

```
NAME="John"            //creates a variable called NAME that has the value “John”
```

- echo is a statement that’s used to print a text or a variable value. For example:

```
echo NAME              //prints the text “NAME”
echo $NAME             //prints “John”
echo "$NAME"           //prints “John”
echo "${NAME}!"        //prints “John!”
```

- read is a statement that's used to read value and save it in a variable name. For example:

```
read x                 //reads a value from the user and save it in x
read -s x              //reads a value from the user (silently) and save it in x
```

- Here are some important variables we can deal with:

```
$HOME                  //the current user home directory
$PWD                   //the current working directory
$0                     //current filename
$1                     //argument number 1
$2                     //argument number 2
$*                     //all arguments “double quoted”
$@                     //all arguments “individually double quoted”
$#                     //the number of given arguments
```

Expressions:

- You can run expressions such as +, -, *, /, % using $(()) such as:

```
      $((x+y))        or        `expr $x+$y`
```

- Other types of expressions (executing statements can be run using $():

```
      $(cat myfile.txt)
```

Note: a shell script file should start with: #! /bin/sh

If [] then fi:

Files verification operators	
if [-e "$FILE"]	______//verifies if its a file that exist
if [-d "$FILE"]	______//verifies if it was a directory
if [-r "$FILE"]	______//verifies if the file is readable
if [-x "$FILE"]	______//verifies if the file is executable
if [-x "$FILE"]	______//verifies if the file is writable
Text comparison operators	
if [$var=’some_text’]	______//verifies if the value of v equals “some_text”
if [$var!=’some_text’]	______//verifies if the value of v not equal to “some_text”
Numbers comparison operators	
if [$var -eq 10]	______//verifies if the value of v equals to 10
if [$var -ne 10]	______//verifies if the value of v is not equal to 10
if [$var -gt 10]	______//verifies if the value of v is greater than 10

```
if [ $var -ge 10 ]                ______//verifies if the value of v is greater or equal to 10
if [ $var -lt 10 ]                ______//verifies if the value of v is less than 10
if [ $var -le 10 ]                ______//verifies if the value of v is less or equal to 10
```

Boolean comparison operators

```
if [ condition1 -a  condition2] _______//verifies if both conditions are true
if [ condition1 -o  condition2] _______//verifies if one of the conditions or both are true
```

For loop:

for i in SOME_VALUES
do
      echo $i
done
	for i in 1 2 3 4; do //loop through values 1 2 3 4
	for i in $(cat filename); do //loop through file lines
	for i in $@ //loop through all args

While loop:

// loop while x<10
x=10
while $x le- 10
do
      echo $x
      x=$((x+1))
done
	// loop through file content and print it
	while read LINE
	do
	  echo $LINE
	done < data.txt