Creating a script

Put your command in an executable file whose first line is the following she-bang:

```
#!/bin/bash
```

Script arguments

Arguments passed to the script via command line are stored in the variables 1, 2, ... Do not forget to use $ to get their value.

Command substitution

To use the result of a command (to store it in a variable for instance):

```
MY_PATH = `command`
MY_PATH = $(command)
```

Conditionals

```
if [[ condition ]] ; then
  commands1
[else
  commands2
]
fi
```

Tests

`! EXPRESSION`	`EXPRESSION` is false
`EXPRESSION1 -a EXPRESSION2`	both expressions are true
`EXPRESSION1 -o EXPRESSION2`	one of the expression is true
`STRING1 = STRING2`	strings are equal
`STRING1 != STRING2`	strings are not equal
`INTEGER1 -eq INTEGER2`	`INTEGER1` is equal to `INTEGER2`
`INTEGER1 -gt INTEGER2`	`INTEGER1` is greater than `INTEGER2`
`INTEGER1 -ge INTEGER2`	`INTEGER1` is greater than or equal to `INTEGER2`
`INTEGER1 -lt INTEGER2`	`INTEGER1` is less than `INTEGER2`
`INTEGER1 -le INTEGER2`	`INTEGER1` is less than or equal to `INTEGER2`
`-e FILE`	`FILE` exists
`-d FILE`	`FILE` exists and is a directory
`-f FILE`	`FILE` exists and is a regular file

For and while loops

```
for variable in [list]; do
  commands
done

while [[ condition ]] ; do
  commands
done
```

References

[1] M. Cooper. *Advanced Bash-Scripting Guide – An in-depth exploration of the art of shell scripting*. http://tldp.org/LDP/abs/html/. 2012.

[2] V. G. Gite. *Linux Shell Scripting Tutorial – A Beginner's handbook*. http://www.freeos.com/guides/lsst/. 2002.

[3] *Bash Hackers Wiki*. http://wiki.bash-hackers.org/doku.php.