Bash

- Bash Cheat Sheet

Bash Cheat Sheet

BASH stands for Bourne-Again SHell

each BASH script should start with the shebang : #!/bin/bash

Comparison operators

Operator	Description		Operator	Description
-eq	Is equal to		-n	Is not null
-ne	Is not equal to		-z	Is null
-gt	Is greater than		-r	file exists and is readable by the user
-ge	Is greater than or equal to		-e	file exists
-lt	Is less then		-f	file is a regular file (not directory or other special type)
-le	Is less than or equal to		-s	file size is greater then 0
==	is equal to		-x	file is executable by the user
!=	is not equal to		-w	file is writable by the user
<	Is less than		!	negates the operator
<=	Is less than or equal to			
>	Is greater then			
>=	Is greater than or equal to			

Variables for script arguments

Variable	Description		Variable	Description

$1 ... $n	Argument number		$#	Number of arguments
$0	Name of the file		$?	Return value (0 = ok, > 0 = error)
$*	List of arguments			

Built in variables

Variable	Description		Variable	Description
$BASH	path to the bash binaries		$BASH_VERSION	version of bash
$BASH_ENV	bash environment (used within the scripts)		$BASH_VERSINFO	more version details
$$	process id of the current script		$PATH	path variable
$EDITOR	default editor		$UID	users id
$EUID	effective user id		$FUNCNAME	name of the current function (executed from within the function)
$GROUPS	groups to which the current user belongs		$HOME	home directory of the current user
$HOSTTYPE	system type		$MACHTYPE	machine type
$LINENO	line number		$PWD	print working directory
$OLDPWD	old print working directory		$OSTYPE	operating system type
$REPLY	default answer for a "read" command without explicit variable declaration		$SECONDS	the time the script has run until this point
$RANDOM	Generate a random number between 0 and 32767			

Declare and a few howtos

Declare a variable as read only

```
declare -r my_read_only_variable=5
```

Declare a variable as an integer

```
declare -i my_integer_variable=5
```

Declare a variable as an array

```
declare -a myvar=(1 2 3 4 5)
```

Declare a function and then run it

```
declare -f myFunction

myFunction(){
echo "my function runs"
}

myFunction
```

Declare a variable and export it

```
declare -x myvar=3
```

Other commands

Variable	Description		Variable	Description