

# bash - scripting

Now that all platforms (beside the apple ones) can be built on linux, I want a fabulous **build-all** script, that automagically creates a folder with all runtimes. Time to refresh my **bash-basics**.

And to say it from the start: **Spaces do matter!!**

- How to set a variable

```
# GOOD:
varname="fortuna"
# WRONG:
varname = "fortuna"
```

- Show argument count

```
#!/bin/bash
echo number of arguments: $#
echo script itself: $0
echo arg1:$1 arg2:$2 arg3:$3 ...
```

- basic conditionals for number comparison with **if-elif-else-construct** using some checks: **-eq**, **-ne**, **-lt**, **-le**, **-gt**, **-ge**

```
#!/bin/bash

#equals
if [[ $# -eq 0 ]]; then
    echo "NO ARGS"
elif [[ $# -eq 1 ]]; then
    echo "ONE ARGUMENT"
else
    echo "$# Arguments"
fi

# not equals
if [[ $# -ne 1 ]]; then
    echo "arg-count != 1"
else
    echo "arg-count == 1"
fi

# check if a directory does NOT exist
if [[ ! -d "linux" ]]; then
    echo "linux-directory does not exist"
fi
```

For string-comparison use: **=,!=**

- Some useful directory-magic:

```
# get the directory from which the script is called
CURRENT_DIR=$PWD

# get the directory where the script is located (relative)
REL_BASEDIR=$(dirname $0)

# get the directory where the script is located (absolute)
ABS_BASEDIR=$( cd $(dirname $0) ; pwd )

# get the directory where the script is located (absolute with resolved symlinks)
ABS_BASEDIR_RESOLVED=$( cd $(dirname $0) ; pwd -P )
```

- Check if something went wrong (e.g. with cmake).

```
cmake -DCMAKE_BUILD_TYPE=Release .. || {  
  echo "Something went wrong with CMAKE! Exiting..."  
  exit 1;  
}  
  
echo "Everything went fine!"
```

- exit the script:

```
# just exit with code 0  
exit  
  
# give it a special exit-code you can use on the shell  
exit 95
```

There are some [special exit-codes](#)

- creating functions:

```
#!/bin/bash  
function myEcho {  
  echo $1  
}  
myEcho Hello
```

- Prevent a command to output errors

```
# if a.txt is not present do not output any error  
rm a.txt 2>/dev/null
```

Links:

\* Bash Conditionals

<http://www.gnu.org/software/bash/manual/bash.html#Bash-Conditional-Expressions>

\* Testing exit values <https://sanctum.geek.nz/arabesque/testing-exit-values-bash/>

\* Functions <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-8.html>

*Published by Thomas Trocha*

*Thu Dec 08 12:14:00 CET 2016*

<http://thomas.trocha.com:80/pebble/bash-scripting>