

Agile

Containers

DevOps  
Automation

DevOps Linux

DevOps Services

Languages

Network

Scripting

Security

Solutions

Virtualization

Windows

Glib Examples

Go Examples

Javascript  
Examples

DHD Examples

APIs

Bash Associative Array

Bash Functions

Bash Regex

Bash

Color Distance [↗](#)Shell Problems [↗](#)

Shell-Scripting

awk

packages.json [↗](#)

sed



Haircare

## Shell-Scripting Cheat Sheet

[Edit Cheat Sheet ↗](#)

See Also:

### Date Handling

#### Convert Date To Unix Timestamp

```
date -d "$date" +%s
```

Note that this only works for American style dates. European "25.06.2014" like dates are not supported. The simple solution is to convert them first to "2014-06-25" for example with

```
sed 's/\([0-9]*\)\\.([0-9]*\)\\.([0-9]*)/\3-\2-\1/'
```

#### Convert From Unix Timestamp

```
date -d "1970-01-01 1234567890 sec GMT"
```

#### Calculate Last Day of Month

Found [here ↗](#):

```
cal $(date "+%M %y") | grep -v ^$ | tail -1 | sed 's/^.* \([0-9]*\)$/\1/'
```

### Lock Files

Using "flock":

```
flock /tmp/myapp.lock <some command>
flock -w 10 /tmp/myapp.lock <some command>
```

Using "lockfile-\*" commands:

## CHEAT SHEETS

Agile

APIs

Containers

Bash Associative Array

DevOps  
Automation

Bash Functions

Bash Regex

Bash

Color Distance [🔗](#)

DevOps Linux

Shell Problems [🔗](#)

DevOps Services

## Shell-Scripting

awk

Languages

packages.json [🔗](#)

sed

Network

## Scripting

Security

Solutions

Virtualization

Windows

Glib Examples

Go Examples

Javascript  
Examples

D4D Examples

```
lockfile-create /tmp/myapp.lock
lockfile-touch /tmp/myapp.lock
lockfile-remove /tmp/myapp.lock
```

Parameter Handling**getopt**

getopt is a standalone command, supporting GNU style long parameters and parameters mixed with options and can be used like this

```
PARAMS=`getopt -o a::bc: --long arga::,argb,argc: -n 'myscript.sh' -- "$@"`
eval set -- "$PARAMS"

while true ; do
  case "$1" in
    -a|--arga)
      case "$2" in
        "") ARG_A='some default value' ; shift 2 ;;
        *) ARG_A=$2 ; shift 2 ;;
      esac ;;
    -b|--argb) ARG_B=1 ; shift ;;
    -c|--argc)
      case "$2" in
        "") shift 2 ;;
        *) ARG_C=$2 ; shift 2 ;;
      esac ;;
    --) shift ; break ;;
    *) echo "Unknown option!" ; exit 1 ;;
  esac
done
```

**getopts**

getopts is shell-builtin

```
while getopts ":ap:" opt; do
  case $opt in
    a)
      echo "Option -a ist set"
      ;;
    p)
      echo "Parameter -p is given with value '$OPTARG'"
      ;;
    \?)
      echo "Unknown option: -$OPTARG"
      ;;
  esac
done
```

**shflags - portable getopts**

## CHEAT SHEETS

Agile	APIs
Containers	Bash Associative Array
DevOps Automation	Bash Functions
DevOps Linux	Bash Regex
DevOps Services	Bash
Languages	Color Distance <a href="#">↗</a>
Network	Shell Problems <a href="#">↗</a>
<b>Scripting</b>	<b>Shell-Scripting</b>
Security	awk
Solutions	packages.json <a href="#">↗</a>
Virtualization	sed
Windows	
Glib Examples	
Go Examples	
Javascript Examples	
D4D Examples	

If you ever need to port between different Unix derivatives use [shflags](#) [↗](#) a Google library providing standard parameter handling. Example:

```
source shflags

DEFINE_string 'value' '0' 'an example value to pass with default value "0"' 'v'

FLAGS "$@" || exit $?
eval set -- "${FLAGS_ARGV}"

echo "${FLAGS_value}!"
```

Other Topics

- [Shell Linting](#) [↗](#)
- [Mail Attachments](#) [↗](#): Dozens of variants to send mail attachments using Unix tools.
- [tail -f until removed](#): When you want to tail a file until it gets removed

```
tail --follow=name myfile
```

- [join - DB-like joining of CSV files](#):

```
join -o1.2,2.3 -t ";" -1 1 -2 2 employee.csv tasks.csv
```

(Note: works only on CSVs without quotes or multi-line strings)

- [awk-based CSV Parser](#) [↗](#)
- list all commands:

```
compgen -c | sort -u
```

- Check for interactive terminal: Run "tty" in silent mode and check the exit code

```
tty -s
```

- [ANSI color matrix](#)
- Sorting column: Use the -k switch of "sort" to sort lines by a column. E.g.

```
cat access.log | sort -k 1
```

- [watch](#) [↗](#): wait for file/directory changes and run a command

## CHEAT SHEETS

Agile	APIs
Containers	Bash Associative Array
DevOps Automation	Bash Functions
DevOps Linux	Bash Regex
DevOps Services	Bash
Languages	Color Distance <a href="#">↗</a>
Network	Shell Problems <a href="#">↗</a>
<b>Scripting</b>	<b>Shell-Scripting</b>
Security	awk
Solutions	packages.json <a href="#">↗</a>
Virtualization	sed
Windows	
Glib Examples	
Go Examples	
Javascript Examples	
D4D Examples	

```
watch -d ls -l
```

- [Unbuffer Output](#) [↗](#):

```
stdbuf -i0 -o0 -e0 <some command> # Best solution  
unbuffer <some command> # Wrapper script from expect
```

- [Apply ulimit change in /etc/security/limits.conf immediately](#)
- [ulimit check list](#)
- [ulimit change Apache limits](#)
- [ulimit Nagios Check Plugin](#)
- dos2unix with vi:

```
:%s/^V^M//g
```

- Resolving any file path to absolute path

```
readlink -e <file path>
```

or use

```
pwd -P
```

- Timeouts in shell:

```
timeout <nr seconds> <cmd>
```

- Convert encodings (HTTP URI encoding, HTML entity encoding...)

```
recode ascii..html  
recode html..ascii
```

[Comment on Disqus](#)