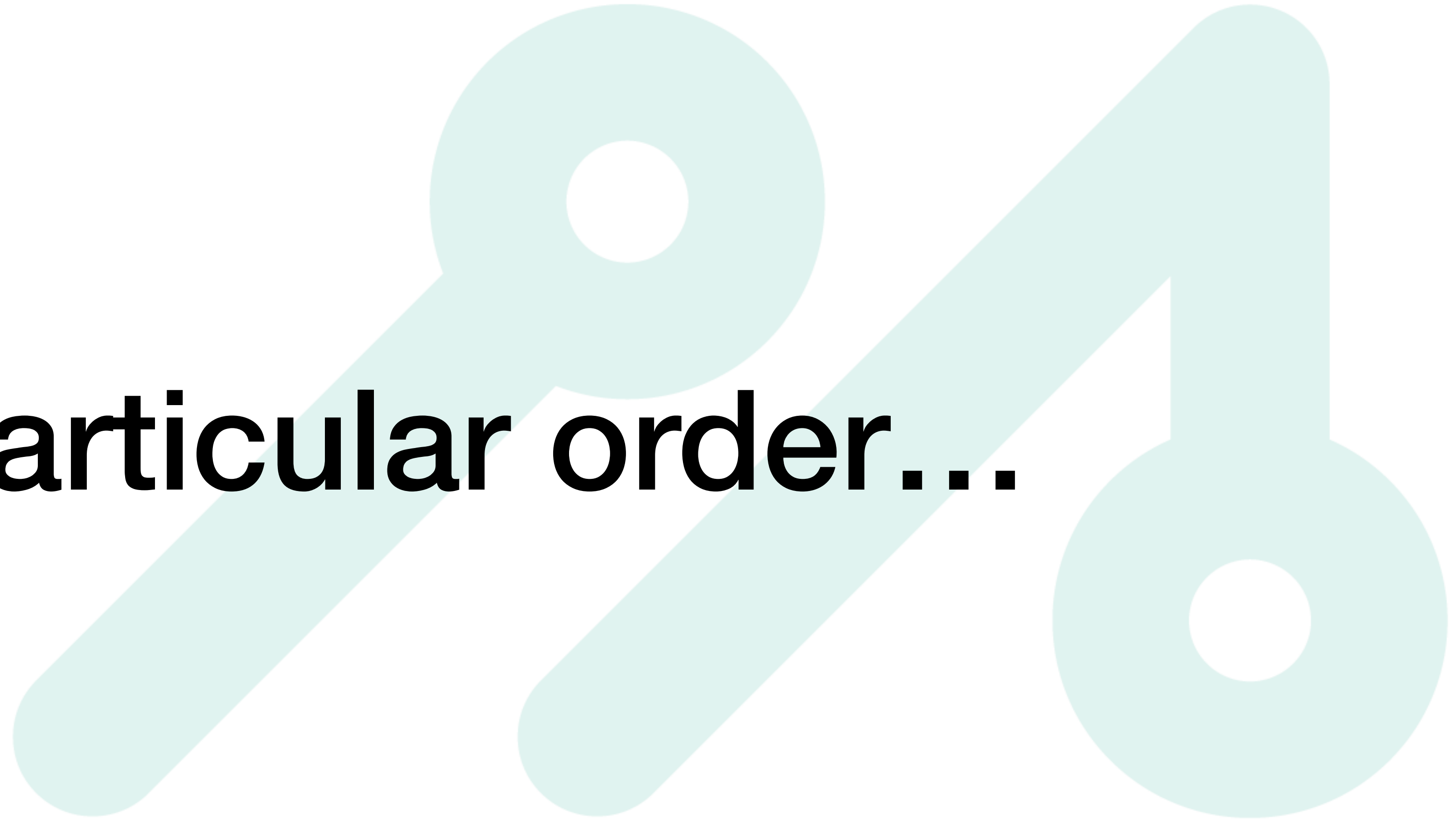


A large, stylized teal graphic in the background, resembling a large number '5' or a similar shape, with a circular cutout in the upper right and a circular base at the bottom right.

50/50

Duncan McCracken
Mondada Pty Ltd

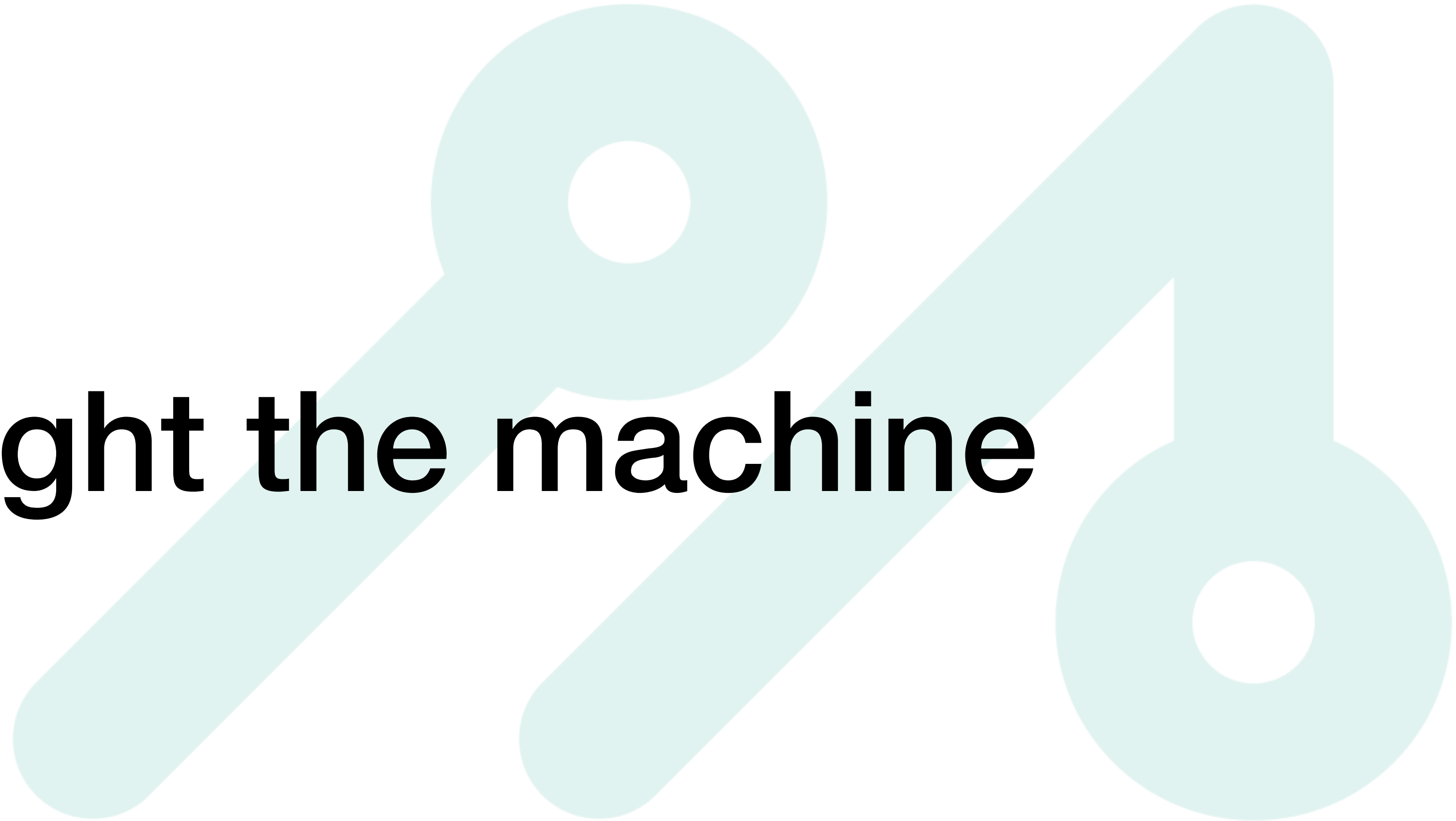
In no particular order...



Have external interests



Don't fight the machine





Use a Single Source of Truth



**Store your original data
prior to manipulation**



**When things go wrong ...
Don't Panic!**

Be methodical



Document your actions





Know when to call it



**Keyboard shortcuts
are your friend**



**(Where possible)
Use what's already there**



Learn a high-level language



There is no undo in the CLI

Insert tab into line command

Control-v <tab>

Check execution time

```
time <cmd>
```




basename

basename <path to something>



dirname

dirname <path to something>



file

file <path to something>



Hard links

In `<source>` `<target>`



pkgutil

pkgutil --expand <pkg> <dir>

pkgutil --flatten <dir> <pkg>



open

open [-a <Application>] <path to something>



strings

strings /path/to/binary

pkgutil --expand-full

```
pkgutil --expand-full <pkg> <dir>
```




pkgutil --flatten-full

pkgutil --flatten-full <dir> <pkg>



**Leverage undocumented
features carefully**

Remove code signature

```
security cms -D -i <signed file> -o <unsigned file>
```



Re-format a plist

```
plutil -convert xml1 <plist>
```

Add a code signature

```
security cms -S -N <signing identity> -i <unsigned file> -o <signed file>
```



Install a profile

```
profiles -l -F <profile>
```

Install a profile at startup

remove: `/private/var/db/ConfigurationProfiles/Setup/.profileSetupDone`
put profile here: `/private/var/db/ConfigurationProfiles/Setup/<profile>`

Brace bash variables

`${var}`



bash variable substitution

```
${var/<str>/<repl>}
```

bash field separator

```
IFS=$'\n'  
unset IFS
```

Find using Spotlight

```
mdfind [-onlyin <path>] kMDItemCFBundleIdentifier = "com.mycompany.app"
```



Find by name

```
find <search path> -name <filename>
```

Find by case insensitive name

```
find <search path> -iname <filename>
```



Find by user

```
find <search path> -user <uid|username>
```

Find by group

```
find <search path> -group <gid|groupname>
```

Find by relative time

```
find <search path> -newerct 'x seconds ago'
```


Delete find results

```
find <search path> -name <filename> -delete
```

Execute single command on find results

```
find <search path> <condition> -exec <cmd> {} \;
```



while loop

```
while <something> ; do <cmd> ; done
```

Redirect **STDERR** to **STDOUT**

```
<cmd> 2>&1
```

Read XML node value

```
xpath <xml file> "//path/to/node/text()"
```

Silence STDERR

`2>/dev/null`

bash command variables

`$(<cmd>)`

Read XML node attribute

```
xpath <xml file> "string(//path/to/node/@attr)" 2>/dev/null
```


Count XML nodes

```
xpath <xml file> "count(//path/to/node)"
```



for loop

```
for i in <sequence> ; do <cmd> ; done
```

Write nested nodes with defaults

```
defaults write <path to plist> Node "<dict><key>boolean</key><true/><key>array</key><array><string>string1</string></array></dict>"
```

Determine the state of Java without annoying requests

```
/usr/libexec/java_home 2>/dev/null
```



Thank You

www.mondada.com.au