DEVHINTS.IO

Edit

Bash scripting cheatsheet

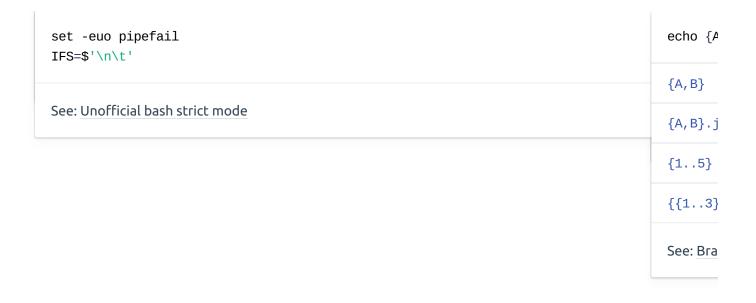


Get 10 free Adobe Stock photos. Start downloading amazing royaltyfree stock photos today.

ads via Carbon

```
Introduction
                                                                                        Example
                                                                                          #!/usr/
  This is a quick reference to getting started with Bash scripting.
                                                                                          name="J
     Learn bash in y minutes (learnxinyminutes.com)
                                                                                          echo "H
     Bash Guide (mywiki.wooledge.org)
                                                                                        Variable
     Bash Hackers Wiki (wiki.bash-hackers.org)
                                                                                          name="J
String quotes
                                                                                          echo $n
  name="John"
                                                                                                "$
  echo "Hi $name" #=> Hi John
  echo 'Hi $name'
                    #=> Hi $name
                                                                                               fall
Shell execution
                                                                                          wildcar
                                                                                          options
                                                                                               3op
  echo "I'm in $(pwd)"
  echo "I'm in `pwd`" # obsolescent
  # Same
                                                                                        Conditio
  See Command substitution
                                                                                          git com
                                                                                          git com
Functions
  get_name() {
                                                                                        Conditio
    echo "John"
  }
                                                                                          if [[ -
```

1 of 12



Parameter expansions

Basics Substitu

```
${foo%s
name="John"
echo "${name}"
                                                                                         ${foo#p
echo "${name/J/j}" #=> "john" (substitution)
echo "${name:0:2}"  #=> "Jo" (slicing)
echo "${name::2}"  #=> "Jo" (slicing)
                                                                                         ${foo%
echo "${name::-1}"  #=> "Joh" (slicing)
echo "${name:(-1)}" #=> "n" (slicing from right)
                                                                                         ${foo/%
echo "${name:(-2):1}" #=> "h" (slicing from right)
                                                                                         ${foo##
echo "${food:-Cake}" #=> $food or "Cake"
                                                                                         ${foo/#
length=2
echo "${name:0:length}" #=> "Jo"
                                                                                         ${foo/f
                                                                                         ${foo//
See: Parameter expansion
                                                                                         ${foo/%
str="/path/to/foo.cpp"
```

```
$foo, or val if unset (or null)
${foo:-val}
                                                                                                 'HE
                                                                                                 "$
${foo:=val}
                                                                 Set $foo to val if unset (or null)
                                                                                                 "$
                                                                  val if $foo is set (and not null)
${foo:+val}
                                                                                                 'he
                                                                                                 "$
                                             Show error message and exit if $foo is unset (or null)
${foo:?message}
                                                                                                 "$
Omitting the : removes the (non)nullity checks, e.g. ${foo-val} expands to val if unset otherwise
str="Hello world"
echo "${str:6:5}" # "world"
echo "${str: -5:5}" # "world"
src="/path/to/foo.cpp"
base=${src##*/} #=> "foo.cpp" (basepath)
dir=${src%$base} #=> "/path/to/" (dirpath)
```

‡ Loops

Basic for loop C-like fo

```
for i in {1..5}; do
echo "Welcome $i"

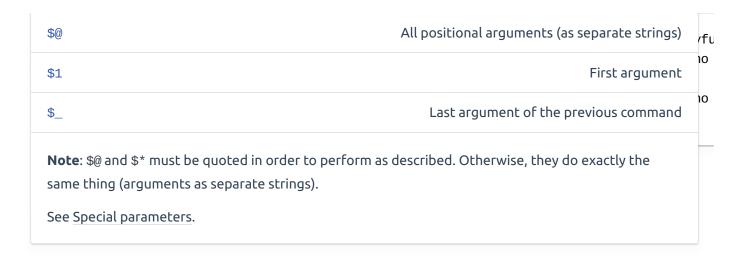
done

With step size

for i in {5..50..5}; do
echo "Welcome $i"
done
```

‡ Functions

```
Defining functions
                                                                                      Returnir
  myfunc() {
                                                                                        myfunc(
      echo "hello $1"
                                                                                            loc
                                                                                            ech
  }
  # Same as above (alternate syntax)
  function myfunc() {
                                                                                        result=
      echo "hello $1"
  }
                                                                                      Raising (
  myfunc "John"
                                                                                        myfunc(
                                                                                          retur
Arguments
                                                                       Number of arguments
  $#
  $*
                                                      All positional arguments (as a single word)
```



Conditionals

Conditions File conditions

```
[[ -e F
Note that [[ is actually a command/program that returns either 0 (true) or 1 (false). Any program
obeys the same logic (like all base utils, such as grep(1) or ping(1)) can be used as condition, se
                                                                                            [[ -r F
examples.
                                                                                            [[ -h F
[[ -z STRING ]]
                                                                                   Empty
                                                                                            [[ -d F
[[ -n STRING ]]
                                                                               Not empty
                                                                                            [[ -w F
[[ STRING == STRING ]]
                                                                                            [[ -s F
[[ STRING != STRING ]]
                                                                                      Not
                                                                                            [[ -f F
[[ NUM -eq NUM ]]
                                                                                            [[ -x F
[[ NUM -ne NUM ]]
                                                                                      Not
                                                                                            [[ FILE
[[ NUM -lt NUM ]]
                                                                                      Less
                                                                                            [[ FILE
                                                                              Less than or
[[ NUM -le NUM ]]
                                                                                            [[ FILE
[[ NUM -gt NUM ]]
                                                                                   Greater than
                                                                           Greater than or equal Example
[[ NUM -ge NUM ]]
                                                                                        Regovo
[[ STRING =~ STRING ]]
                                                                                            # Strin
((NUM < NUM))
                                                                             Numeric cond
                                                                                            if [[ -
                                                                                              echo
More conditions
                                                                                            elif [[
                                                                                              echo
                                                                     If OPTIONNAME is en
[[ -o noclobber ]]
                                                                                            else
```

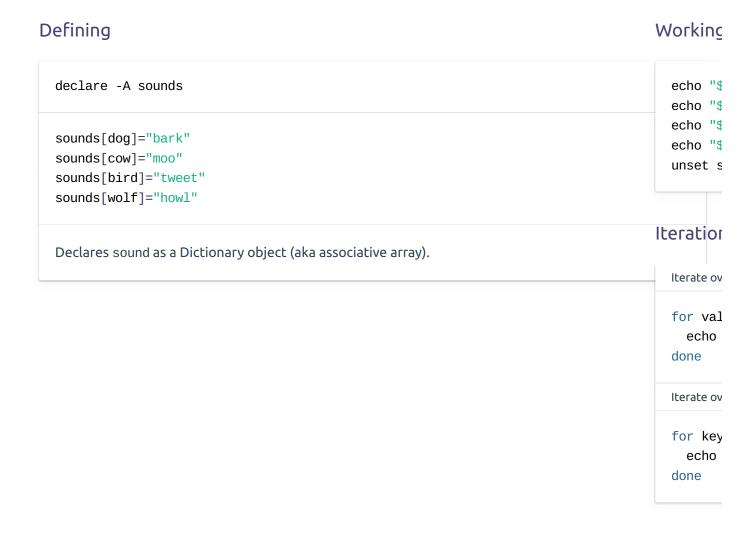
```
echo
[[ ! EXPR ]]
                                                                                          fi
[[ X && Y ]]
                                                                                          # Combi
[[ X || Y ]]
                                                                                          if [[ X
                                                                                            . . .
                                                                                          fi
                                                                                          # Equal
                                                                                          if [[ "
                                                                                          # Regex
                                                                                          if [[ "
                                                                                          if (( $
                                                                                             echc
                                                                                          fi
                                                                                          if [[ -
                                                                                            echo
                                                                                          fi
```

‡ Arrays

Defining arrays Working Fruits=('Apple' 'Banana' 'Orange') echo "\$ echo "\$ echo "\$ Fruits[0]="Apple" echo "\$ Fruits[1]="Banana" echo "\$ Fruits[2]="Orange" echo "\$ echo "\$ echo "\$ **Operations** Iteration Fruits=("\${Fruits[@]}" "Watermelon") # Push Fruits+=('Watermelon') # Also Push Fruits=("\${Fruits[@]/Ap*/}") # Remove by regex match for i i unset Fruits[2] # Remove one item echo Fruits=("\${Fruits[@]}") # Duplicate done Fruits=("\${Fruits[@]}" "\${Veggies[@]}") # Concatenate

```
lines=(`cat "logfile"`) # Read from file
```

Dictionaries



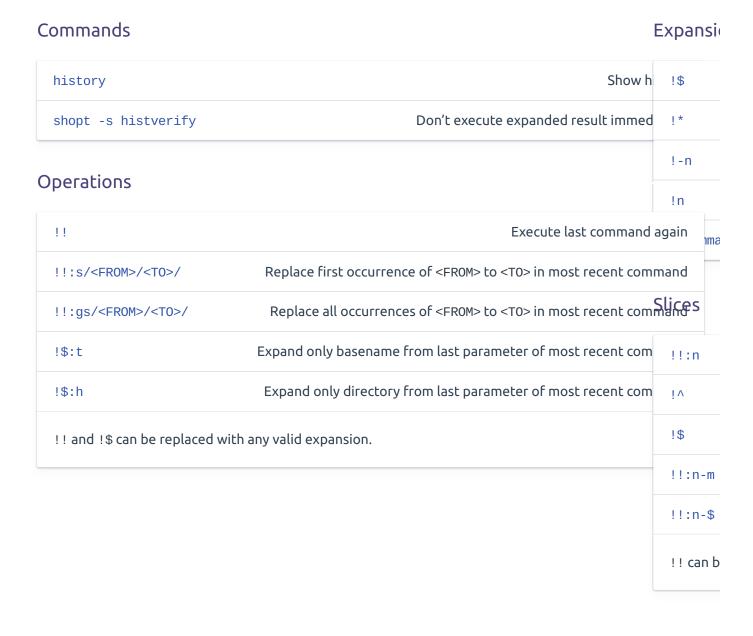
Options

Options Glob op

```
set -o noclobber # Avoid overlay files (echo "hi" > foo)
set -o errexit # Used to exit upon error, avoiding cascading errors
set -o pipefail # Unveils hidden failures
set -o nounset # Exposes unset variables

Set GLOE
```

History



Miscellaneous


```
if grep -q 'foo' ~/.bash_history; then
                                                                                            Lng
  echo "You appear to have typed 'foo' in the past"
                                                                                            10
pwd # /home/user/foo
                                                                                       $0
read -n 1 ans
               # Just one character
                                                                                       $_
                                                                       All lower case le
[:lower:]
                                                                                       ${PIPES
[:digit:]
                                                                                  All
                                                                                       See Spe
                                                                             All whitespace
[:space:]
[:alpha:]
                                                                                 All letters
                                                                        All letters and digits
[:alnum:]
Example
echo "Welcome To Devhints" | tr '[:lower:]' '[:upper:]'
WELCOME TO DEVHINTS
```

‡ Also see

Bash-hackers wiki (bash-hackers.org)
Shell vars (bash-hackers.org)
Learn bash in y minutes (learnxinyminutes.com)
Bash Guide (mywiki.wooledge.org)
ShellCheck (shellcheck.net)

Search 358+ cheatsheets



Top cheatsheets Other CLI cheatsheets Cron Homebrew Elixir ES2015+ cheatsheet cheatsheet cheatsheet cheatsheet adb (Android httpie React.js Vimdiff cheatsheet cheatsheet cheatsheet Debug Bridge) cheatsheet Vim scripting Vim

cheatsheet cheatsheet cheatsheet cheatsheet