

Bash Cheatsheet

This document contains bash specific commands / syntax which may not be completely POSIX compliant.

Variables

- *Define*

```
1 name='PJ'
2 age=10
3 os=$(uname)
```

- *Use*

```
1 echo $name
2 printf '- %s\n' \
3         $age \
4         $os
```

- *Execute*

```
$ bash tmp.sh
```

- *or if it has shebang and execute permission*

```
$ ./tmp.sh
PJ
- 10
- Linux
$_
```

- *Length*

```

1 a='Hello World'
2 b=1917
3 echo ${#a} # 11
4 echo ${#b} # 4

```

Special Variables

Variable	Description
\$0	Name of script
\$1, \$2, \$3, ...	First, Second, Third, ... argument of script
\$#	Number of arguments were passed to the script
\$@	All arguments of the script (can be iterated)
\$*	All arguments of the script (cannot be iterated)
\$?	Return value of the last execution in script
\$\$	The PID of the script
\$USER	The user which is running the script (username)
\$HOSTNAME	The host name of the machine
\$LINENO	Current line number inside script
\$RANDOM	Random number

Input

- *Basic*

```

1 read input_variable
2 echo ${input_variable}

```

- *with prompt message*

```

1 read -p 'are you sure? ' input_variable

```

- *silent input*

```

1 read -sp 'Input is silent: ' input_variable

```

Arrays

- *Define*

```
1 files=('f1.txt' 'f2.txt' 'f3.txt')
2 echo ${files[0]} # the first element
3 echo ${files[*]} # all elements
4 echo ${files[@]} # same
5 echo ${#files}   # size/length of array
```

- *Add elements*

```
1 files+=('f4.txt')
```

- *Remove elements*

```
1 unset files[0]
```

Arithmetic

Basic Expressions

Operator	Operation
+	Addition
-	Subtraction
*	Multiply
/	Deviation
%	Remainder
var++	Increase by 1
var--	Decrease by 1

- *let*

```
1 let a=4+5 # 9
2 let 'A = 4 + 6' # 10
3 let a++ # 10
4 let A-- # 9
```

- *expr*

```
1 var_two=$( expr 4 \* 5 ) # 20
```

- *Double Parentheses*

```
1 a=$((3 + 5)) # 8
2 b=$(( a + 3 )) # 11
3 (( b++ )) # 12
```

Conditions

- Test Operations

Operator	Operation
!	Not

- String Operations

Operator	Operation
-z	Is null
-n	Is not null
==	Is equal
!=	Is not equal

- Numerical Operations

Operator	Operation
-eq	equal
-lt	less than
-gt	greater than
-le	less-equal to
-ge	greater-equal to

- File Operations

Operator	Operation
-e	Exists
-d	Exists and it's a directory
-f	Exists and it's a file
-r	Exists and has <i>read</i> permission
-w	Exists and has <i>write</i> permission
-x	Exists and has <i>execute</i> permission
-s	Exists and it's not empty

- *if*

```

1 if [[ `echo $(date +%s) % 5 | bc` -eq 0 ]]; then
2     echo "It can be divided by 5 without any reminder"
3 elif [[ ${second_condition} ]]; then
4     echo "The second is true"
5 else
6     printf '%s\n' \
7         "Nothing is true" \
8         "Everything is permitted"
9 fi

```

- *Nested*

```

1 if [ 10 -gt 5 ]; then
2     echo True
3     if (( 10 % 2 == 0 )); then
4         echo and Even
5     fi
6 fi

```

- *inline*

```

1 [[ ${some_condition} ]] && echo "it's true" || echo 'false'

```

Loops

- *for*

```

1  for i in {1..10}
2  do
3      echo ${i}
4  done

```

- ***while***

```

1  counter=1
2  while [[ ${counter} -le 9 ]]; do
3      echo "${counter}"
4      ((counter++))
5  done

```

- ***until***

```

1  counter=1
2  until [[ ${counter} -gt 9 ]]
3  do
4      echo "${counter}"
5      ((counter++))
6  done

```

- ***select***

```

1  names='Kyle Cartman Stan Quit'
2  PS3='Select character: '
3  select name in ${names}; do
4      [[ $name == 'Quit' ]] && break
5      echo Hello ${name}
6  done
7  echo Bye

```