

Shell variables

Creation, inspection, modification, lists...

Creating a variable

Can mix alpha-numeric
chars and _

myvar="value string"

Can't start with
a number

No space around
= sign

number, string or
'command'

Exporting a variable

```
export myvar="value string"
```

or

```
myvar="value string"
```

```
export myvar
```

Using variable values

```
echo $myvar
```

```
echo ${myvar}
```

```
echo "${myvar}_something"
```

Removing a variable

```
unset myvar
```

Removing value of a variable

```
myvar=
```

Test if a variable is set

```
[[ -v myvar ]];  
echo $?
```

Return codes:

0 : success (variable myvar is set)

1 : failure (variable myvar is not set)

Test if a variable is *not* set

```
[[ -z ${myvar+x} ]];  
echo $?
```

Return codes:

0 : success (variable myvar is not set)

1 : failure (variable myvar is set)



can be any string

Substitute default value

If the variable `myvar` is not set, use “default” as its default value

```
echo ${myvar:-"default"}
```

if `myvar` is set:
 display its value
else:
 display “default”

no spaces

Set default value

If the variable `myvar` is `not` set, then set “default” as its value

```
echo ${myvar:=“default”}
```

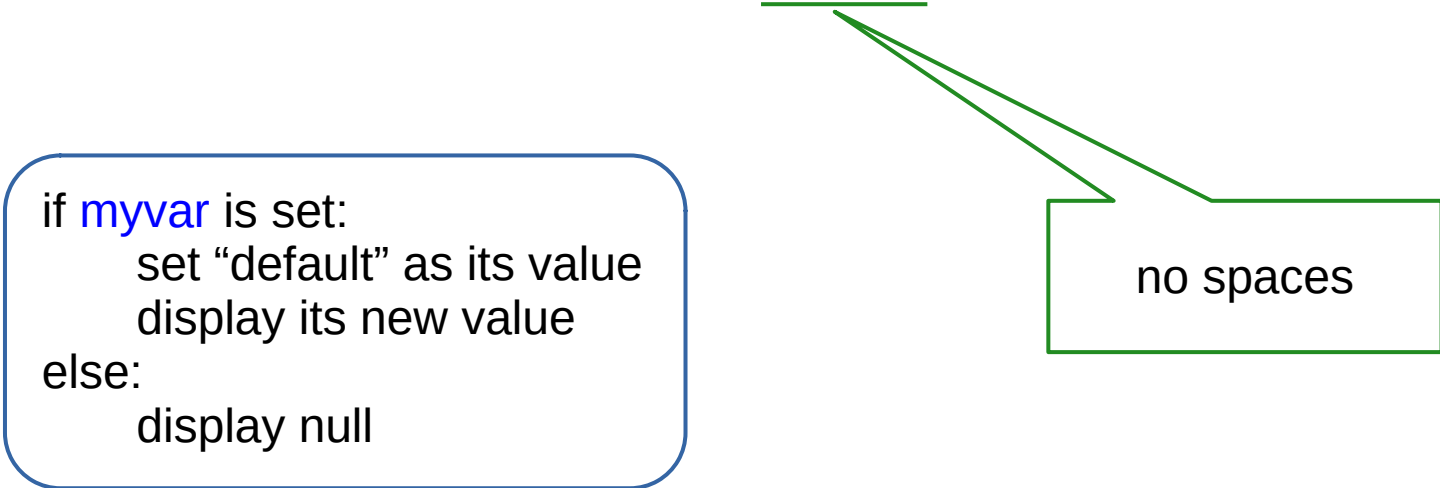
if `myvar` is set:
display its value
else:
set “default” as its value
display its new value

no spaces

Reset value if variable is set

If the variable `myvar` is set, then set “default” as its value

```
echo ${myvar:+"default"}
```



if `myvar` is set:
 set “default” as its value
 display its new value
else:
 display null

no spaces

The diagram consists of a blue rounded rectangle on the left containing the logic for the shell command. A green line extends from the underlined portion of the command `${myvar:+"default"}` to a green rectangle on the right containing the text 'no spaces'.

List of variable names

```
echo ${!H*}
```

List of names of shell
Variables that start with H

Length of string value

```
echo ${#myvar}
```

Display length of the string value
of the variable `myvar`

If `myvar` is not set, display `0`

Slice of string value

```
echo ${myvar:5:4}
```

Display 4 chars of the string value
of the variable `myvar` skipping first
5 chars

offset

slice length

Remove matching pattern

match once

```
echo ${myvar#pattern}  
echo ${myvar##pattern}
```

match
max possible

Keep matching pattern

match once

```
echo ${myvar%pattern}  
echo ${myvar%%pattern}
```

match
max possible

Replace matching pattern

match once &
replace with **string**

echo \${myvar/**pattern**/**string**}

echo \${myvar//**pattern**/**string**}

match max possible
& replace with **string**

Replace matching pattern by location

match at **beginning** &
replace with **string**

```
echo ${myvar/#pattern/string}  
echo ${myvar/%pattern/string}
```

match at the **end**
& replace with **string**

Changing case

```
echo ${myvar,}
```

Change first char
to lower case

```
echo ${myvar,,}
```

Change all chars
to lower case

Change first char
to upper case

```
echo ${myvar^}
```

```
echo ${myvar^^}
```

Change all chars
to upper case

Restricting value types

declare -i myvar

Only integers
assigned

declare -l myvar

Only lower case
chars assigned

declare -u myvar

Only upper case
chars assigned

Variable is
read only

declare -r myvar

Removing restrictions

declare +i myvar

integer
restriction
removed

declare +l myvar

lower case
restriction
removed

declare +u myvar

upper case
restriction
removed

declare +r myvar

Can't do
once it is
read only

Indexed arrays

declare `arr`
as an
indexed array

```
declare -a arr  
$arr[0]="value"
```

set value of element
with index 0 in the array

Value of element with
index 0 in the array

```
echo ${arr[0]}
```

```
echo ${#arr[@]}
```

Number of elements
in the array

```
echo ${!arr[@]}
```

Display all
Indices used

```
echo ${arr[@]}
```

Display values of all
elements of the array

Delete element with
Index 2 in the array

```
unset 'arr[2]'
```

```
arr+=("value")
```

Append an element
with a value to
the end of the array

Associative arrays

```
declare -A hash  
$hash["a"]="value"  
echo ${hash["a"]}  
echo ${#hash[@]}  
echo ${!hash[@]}  
echo ${hash[@]}  
unset 'hash["a"]'
```

declare **hash**
as an
associative array

Value of element with
index "a" in the array

Display all
Indices used

Delete element with
Index "a" in the array

set value of element
with index "a" in the array

Number of elements
in the array

Display values of all
elements of the array

Shell variable manipulations are fast !