Use default values

${parameter:-word}

If the variable is undefined return an alternative

echo ${name:-Undefined} → Undefined

name=”Charlie”
echo ${name:-Undefined} → Charlie

Assign default values

${parameter:=word}

If the variable is undefined set and return an alternative

unset page
echo ${page:=portrait} → portrait

echo $page → portrait

page=”landscape”
echo ${page:=portrait} → landscape

echo $page → landscape

Display error if null or unset

${parameter:?word}

If the variable is undefined exit with an error message

echo ${length:?Length is unset} → bash: length: Length is unset

Use alternate value

${parameter:+word}

If the variable is undefined do nothing otherwise substitute the word

fish=”trout”
echo ${fish:+salmon} → salmon

echo $fish → trout

Substring expansion

${parameter:offset}
${parameter:offset:length}

Return a substring of the variable

animal=”aardvark”
echo ${animal:4} → vark

message=”No such file”
echo ${message:0:7} → No such

offset is zero-based

Names matching prefix

${!parameter*}

${!parameter@}

Expands to the names of variables whose names begin with parameter

```
echo ${!BASH*}            → BASH BASHOPTS BASHPID ...
```

```
export coord_x=42         → coord_x coord_y
export coord_y=100
echo ${!coord*}
```

List of array keys

${!parameter[@]}

${!parameter[*]}

If parameter is an array variable, expands to the list of array indices (keys) assigned in parameter

```
colours=( red green blue )   → 0 1 2
echo ${!colours[@]}
```

Parameter length

${#parameter}

Length in characters of the value of parameter

```
veg=”Broccoli”            → 8
echo ${#veg}
```

Remove matching prefix

${parameter#word}

${parameter##word}

Removes the prefix from parameter matching word

```
dir=”/home/dave/some/dir”
echo ${dir#/home/dave/}   → some/dir
echo ${dir#*/}            → home/dave/some/dir
echo ${dir##*/}           → dir
```

Remove matching suffix

${parameter%word}

${parameter%%word}

Removes the suffix from parameter matching word

```
dir=”/home/dave/some/dir”
echo ${dir%/some/dir}     → /home/dave
echo ${dir%/*}            → home/dave/some
echo ${dir%%/some/*}      → /home/dave
```

Pattern substitution

Finds pattern in parameter and replaces it with string

${parameter/pattern/string}

${parameter//pattern/string}

msg='An ant is an ant'

echo ${msg/ant/insect} → An insect is an ant

echo ${msg/%ant/insect} → An ant is an insect

echo ${msg//ant/insect} → An insect is an insect

colours=(red green blue)

echo ${colours[@]/green/yellow} → red yellow blue

scans each array element

Case modification

Modifies the case of alphabetic characters in parameter matching pattern

${parameter^pattern}

${parameter^^pattern}

${parameter,pattern}

${parameter,,pattern}

msg='the quick brown fox'

echo ${msg^} → The quick brown fox

echo ${msg^^} → THE QUICK BROWN FOX

echo ${msg^^o} → the quick brOwn fOx

echo ${msg^^[tqbf]} → The Quick Brown Fox