#! /bin/bash

var="Welcome to the geekstuff"

echo ${#var}

................

24

Length must be the number greater than or equal to zero.

#! /bin/bash

var="Welcome to the geekstuff"

echo ${var:15}

echo ${var:15:4}

......................

geekstuff

geek

Write a shell script that takes a command line argument and reports on whether it is directory, a file, or something else.

if [ -f $1 ]

then

echo "it is a file"

elif [ -d $1 ]

then

echo it is a directory

else

echo "something else"

fi

OUTPUT:

$sh file1 abc

abc is a file

...........

### Search and Replace in an array elements

The following example, searches for Ubuntu in an array elements, and replace the same with the word ‘SCO Unix’.

#!/bin/bash

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

echo ${Unix[@]/Ubuntu/SCO Unix}

....

Debian Red hat SCO Unix Suse Fedora UTS OpenLinux

In this example, it replaces the element in the 2nd index ‘Ubuntu’ with ‘SCO Unix’. But this example will not permanently replace the array content.

### Add an element to an existing Bash Array

The following example shows the way to add an element to the existing array.

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

Unix=("${Unix[@]}" "AIX" "HP-UX")

echo ${Unix[7]}

...

AIX

In the array called Unix, the elements ‘AIX’ and ‘HP-UX’ are added in 7th and 8th index respectively.

### Remove an Element from an Array

unset is used to remove an element from an array.unset will have the same effect as assigning null to an element.

$cat arraymanip.sh

#!/bin/bash

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

unset Unix[3]

echo ${Unix[3]}

The above script will just print null which is the value available in the 3rd index. The following example shows one of the way to remove an element completely from an array.

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

pos=3

Unix=(${Unix[@]:0:$pos} ${Unix[@]:$(($pos + 1))})

echo ${Unix[@]}

...

Debian Red hat Ubuntu Fedora UTS OpenLinux

In this example, ${Unix[@]:0:$pos} will give you 3 elements starting from 0th index i.e 0,1,2 and ${Unix[@]:4} will give the elements from 4th index to the last index. And merge both the above output. This is one of the workaround to remove an element from an array.

### Remove Bash Array Elements using Patterns

In the search condition we can give the patterns, and stores the remaining element to an another array as shown below.

#!/bin/bash

declare -a Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora');

declare -a patter=( ${Unix[@]/Red\*/} )

echo ${patter[@]}

...............

Debian Ubuntu Suse Fedora

The above example removes the elements which has the patter Red\*.

### Copying an Array

Expand the array elements and store that into a new array as shown below.

#!/bin/bash

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

Linux=("${Unix[@]}")

echo ${Linux[@]}

....

Debian Red hat Ubuntu Fedora UTS OpenLinux

### Concatenation of two Bash Arrays

Expand the elements of the two arrays and assign it to the new array.

#!/bin/bash

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

Shell=('bash' 'csh' 'jsh' 'rsh' 'ksh' 'rc' 'tcsh');

UnixShell=("${Unix[@]}" "${Shell[@]}")

echo ${UnixShell[@]}

echo ${#UnixShell[@]}

...

Debian Red hat Ubuntu Suse Fedora UTS OpenLinux bash csh jsh rsh ksh rc tcsh

14

It prints the array which has the elements of the both the array ‘Unix’ and ‘Shell’, and number of elements of the new array is 14.

### Deleting an Entire Array

unset is used to delete an entire array.

#!/bin/bash

Unix=('Debian' 'Red hat' 'Ubuntu' 'Suse' 'Fedora' 'UTS' 'OpenLinux');

Shell=('bash' 'csh' 'jsh' 'rsh' 'ksh' 'rc' 'tcsh');

UnixShell=("${Unix[@]}" "${Shell[@]}")

unset UnixShell

echo ${#UnixShell[@]}

...

0

After unset an array, its length would be zero as shown above.

### Load Content of a File into an Array

We can load the content of the file line by line into an array.

#Example file

$ cat logfile

Welcome

to

thegeekstuff

Linux

Unix

$ cat loadcontent.sh

#!/bin/bash

filecontent=( `cat "logfile" `)

for t in "${filecontent[@]}"

do

echo $t

done

echo "Read file content!"

....

Welcome

to

thegeekstuff

Linux

Unix

Read file content!