Shell arrays are dynamic structures; they don't require advance declaration before use.

Array Initialization

The most convenient way to initialize an array is to enclose the initial element values in a pair of parenthesis. The following code initializes an array named nameList:

```
nameList=(Stan Kyle Cartman)
```

Alternatively, one can initialize the array by directly assigning a value to each individual element:

```
nameList[0]=Stan
nameList[1]=Kyle
nameList[2]=Cartman
```

Array Operations

Array length:	`echo ${#nameList[@]}`	`# 3`
All elements:	`echo ${nameList[@]}`	`# Stan Kyle Cartman`
All indexes:	`echo ${!nameList[@]}`	`# 0 1 2`
Element at index 0:	`echo ${nameList[0]}`	`# Stan`
Length of element at index 0:	`echo ${#nameList[0]}`	`# 4`
Slice 2 elements starting at index 1:	`echo ${nameList[@]:1:2}`	`# Kyle Cartman`

Note: The curly braces are required.

Looping over Arrays

The following example illustrates how to iterate through all the elements of the array and print them to standard output:

```
n=0
while [ $n -lt ${#nameList[@]} ]
do
    echo ${nameList[$n]}
    (( n++ ))
done
```

where ${#nameList[@]} is the number of elements in the array and ${nameList[$n]} is the syntax to access the element at index n.

Alternatively, one can iterate, without indexing, over the members of an array using a for loop:

```
for name in ${nameList[@]}
do
    echo $name
done
```

Note: The above examples are available for download via Moodle under the title "Shell Arrays Script".