**How to pass arrays to the functions**

Lets pass arguments to shell script which in turn passes array to the function

#!/bin/bash

display\_array() {

array=$@

echo "The array inside the function is ${array[\*]}"

for item in "$@"

do

echo "Argument is ${item}"

done

}

test\_input\_1=(1 2 3 4 5)

for item in "${test\_input\_1[@]}"

do

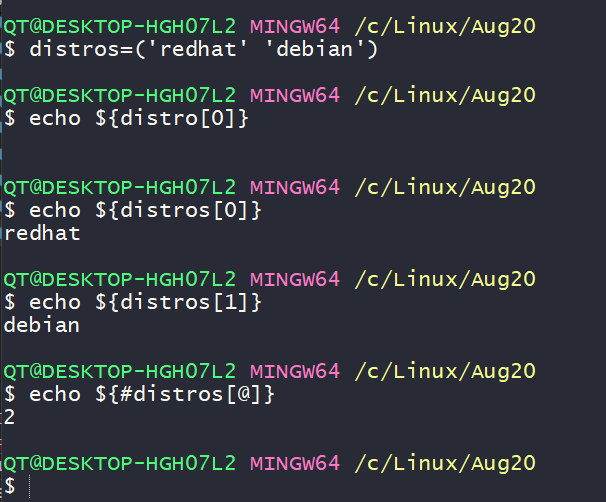
echo "External $item"

done

echo "The orginal array for test input 1 is ${test\_input\_1[\*]}"

echo "Calling function"

display\_array ${test\_input\_1[\*]}

Execute the following commands as shown below 

**Variable scope**

By default variables inside the function are global variables

#!/bin/bash

test\_variable\_1=10

testfunction() {

test\_variable\_1=50

test\_variable\_2=100

}

testfunction

echo "$test\_variable\_1"

echo "$test\_variable\_2"

In shell if your variable has to be local inside function, you have to explicit

#!/bin/bash

test\_variable\_1=10

testfunction() {

test\_variable\_1=50

test\_variable\_2=100

local test\_variable\_3=200

echo "Inside function: $test\_variable\_3"

}

testfunction

echo "$test\_variable\_1"

echo "$test\_variable\_2"

echo "Outside function: $test\_variable\_3"

**Returning values from functions**

Shell sample

#!/bin/bash

print\_message() {

input="$1"

output="Message is ${input}"

return $output

}

$reply=$print\_message "1"}

echo $reply