Shell Scripting/Loops:

While Loops: As long as the result of the condition returns True, the code block under while loop runs. When the condition returns to False, the loop execution is terminated and the program control moves further to the next operation.

#!/bin/bash

number=1

while [[ $number -le 10 ]]

do

  echo $number

  ((number++))

done

echo "Now, number is $number"

Until loops: It will execute the commands within it until the test becomes true.

#!/bin/bash

number=1

until [[ $number -ge 10 ]]

do

  echo $number

  ((number++))

done

echo "Now, number is $number"

For loops: To iterate a block of code for each of the items in a given list.

#!/bin/bash

echo "Numbers:"

for number in 0 1 2 3 4 5 6 7 8 9

do

   echo $number

done

Using arrays with the for loop:

#!/bin/bash

devops\_tools=("docker" "kubernetes" "ansible" "terraform" "jenkins")

for tool in ${devops\_tools[@]}

do

   echo $tool

done

Continue and Break Statements:

#!/bin/bash

number=1

until [[ $number -lt 1  ]]

do

  echo $number

  ((number++))

  if [[ $number -eq 100 ]]

  then

    break

  fi

done

#!/bin/bash

number=1

until [[ $number -lt 1  ]]

do

  ((number++))

  tens=$(($number % 10))

  if [[ $tens -eq 0 ]]

  then

    continue

  fi

  echo $number

  if [[ $number -gt 100 ]]

  then

    break

  fi

done

Shell Scripting/Functions:

#!/bin/bash

Welcome () {

    echo "Welcome to Linux Lessons $1 $2 $3"

}

Welcome Joe Matt Timothy