**Reading input from files**

* Lets assume we have list of server ip addresses or hosts in the file called as servers.txt.
* We are asked to find which servers are up
* This script is working but it is unable to redirect the output to a text

#!/bin/bash

# Usage: ./checkservers <servers-filepath>

# servers-filepath is a text file with each server in new line

if [ ! -f "$1" ] ;

then

echo "The input to $0 should be a file"

fi

echo "The following servers are up on $(date +%x)" > checkservers.out

while read server;

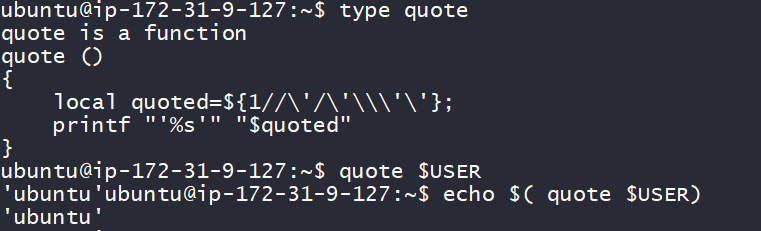
do

ping -c1 "$server"&& echo "Serverup $server" >> checkservers.out

done < $1

cat checkservers.out

Building Blocks for reusability : Functions

* DRY principle(Don’t Repeat Your Self)
* We will cover the following aspects
  + Introduction
  + Passing Parameters to the functions
  + Variable scope
  + Returning values from functions
  + Recursive functions
* Functions are internally represented as blocks of code in memory as *named elements*. These elements can be created within shell environment, as well as within the script execution.
* Execute declare -F in the bash. The output of this command might vary with distribution 
* Functions can be created using the following two syntaxes

Syntax 1:

function-name() {

<code to be executed>

}

Syntax 2:

function <function-name> {

<code to be executed>

}

* Lets start with a simple function

show\_system\_details() {

echo "Uptime is"

uptime

echo "Cpu details"

lscpu

echo "User list"

who

}

is\_file() {

if [ ! -f "$1" ]; then

echo "$1 is not a file"

exit 2

fi

}

backup\_file() {

is\_file "$1"

new\_file\_loc="${1}.bak"

cp $1 $new\_file\_loc

echo "file is copied to $new\_file\_loc"

} backup\_file "/home/ubuntu/1.txt"