Quick Reference Card 7 UNIX/Linux Scripting (3038)

Your? host logo

#01_hello.sh

```
#!/bin/bash                    #Print "Hello world"
# Author:   Robert Zondervan
# Created:  2005-06-05
# Changelog (date, code, reason):
echo -e "\aHello\nworld"
exit 0
```

#02_name1_read_vars.sh

```
#!/bin/bash                    #Read and print first and last name
echo "Please enter your first name:"
# first name gets assigned to variable FIRSTNAME:
read FIRSTNAME
echo "Please enter your last name:"
# last name gets assigned to variable LASTNAME:
read LASTNAME
# print the greeting:
echo "Welcome to the club, $FIRSTNAME $LASTNAME"
exit 0
```

#03_name2_read_vars.sh

```
#!/bin/bash                    #Read and print first and last name
echo "Please enter your first name:"
# first name gets assigned to variable FIRSTNAME
read FIRSTNAME
echo "Please enter your last name:"
# last name gets assigned to variable LASTNAME
read LASTNAME
# create a new NAME variable
NAME="$FIRSTNAME $LASTNAME"
# print the greeting:
echo "Welcome back home, $NAME"
exit 0
```

#04_info_vars.sh

```
#!/bin/bash                    #Print info about the current login
login=`whoami`
path=`pwd`
echo "The current login is: $login"
echo "The current path is: $path"
exit 0
```

#05_sum_formats_var.sh

```
#!/bin/bash                    #Add two whole numbers together
# All arithmetic formats possible under Bash are used, one after another
# First declare INTEGERS1,2 and SUM as integer variables
declare -i INTEGER1 ; declare -i INTEGER2
declare -i SUM
echo "Please enter first integer: " #Read first integer
read INTEGER1                  #Read second integer:
echo "Please enter second integer: "
read INTEGER2
# this uses 'expr' for Bourne shell compatibility:
RESULT=`expr $INTEGER1 + $INTEGER2`
echo "The 'expr' command returns the result: $RESULT."
# this uses the Bash built-in 'let':
let RESULT="$INTEGER1 + $INTEGER2"
echo "The 'let' built-in returns the result: $RESULT."
# this uses a Bash-specific arithmetic expression:
RESULT=$[$INTEGER1 + $INTEGER2]                #or:
#RESULT=$(($INTEGER1 + $INTEGER2))
echo "Using an arithmetic expression in Bash, the result is: $RESULT."
# this one uses the variables declared as integers above:
SUM=INTEGER1+INTEGER2
echo "Using the variables declared as integers, the sum is: $SUM."
exit 0
```

#06_find_file_var.sh

```
#!/bin/bash                    #Search for files in the current directory
# The user is prompted to enter a file name; if no name is entered, we
# search for the default value anyway, which is set to "*.bak"
echo "Please enter the file to be searched for (default is: *.bak):"
read FILE
find . -name "${FILE:="*.bak"}"
exit 0
```

#07_find_check_existence.sh

```
#!/bin/bash                    #Check whether an executable file exists
echo "Please enter a file name: "
read FILENAME
if test -e $FILENAME
then
  if test -x $FILENAME
  then
    echo "The file exists and is executable."
  else
    echo "The file exists but is not executable."
  fi
else
  echo "The file does not exist."
fi
exit 0
```

#08_yes_no.sh

```
case "$VARIABLE" in
  [yY] | [yY][eE][sS] | [yY] [eE] [aA] [hH] )
    ... ;;
  [nN] | [nN][oO] | [nN][oO][pP][eE] )
    ... ;;
    echo error message ;;
esac
```

#09_counter1_while_loop.sh

```
#!/bin/bash                    #Iterate over a "while" loop 100 times.
# this declares the COUNTER variable as an integer
# which gets assigned the initial value of 1
declare -i COUNTER=1
while   test $COUNTER -le 100
do
        echo "The counter stands at $COUNTER."
        COUNTER=COUNTER+1
        sleep 1
done
exit 0
```

#09_counter2_until_loop.sh

```
#!/bin/bash
# A script to iterate over a simple \texttt{until} loop 100 times.
# this declares the COUNTER variable as an integer
# which gets assigned the initial value of 1
declare -i COUNTER=1
until   test $COUNTER -gt 100
do
        echo "The counter stands at $COUNTER."
        COUNTER=COUNTER+1
        sleep 1
done
exit 0
```

#10_lowercase1_for_loop.sh

```
#!/bin/bash
# This script renames all files in the current
# directory so that they have all-lowercase file names.
for     FILE in `find . -type f -maxdepth 1`
do
        NEWFILE=`echo $FILE | tr [A-Z] [a-z]`
        if      test $FILE != $NEWFILE
        then
                echo mv $FILE $NEWFILE
        fi
done
exit 0
```

#11_lowercase2_for_loop_test_file.sh

```
#!/bin/bash
# This script renames all files in the current
# directory so that they have all-lowercase file names.
# 2nd version: Now we also check whether the file
# already exists with lowercase lettering.
for     FILE in `find . -type f -maxdepth 1`
do
        NEWFILE=`echo $FILE | tr [A-Z] [a-z]`
        if      test $FILE != $NEWFILE
        then
                if      test -e $NEWFILE
                then
                        echo "There is already a file with the name $NEWFILE."
                        echo "$FILE will not be renamed."
                        # Skip the rest and begin next loop iteration:
                        continue
                fi
                echo mv $FILE $NEWFILE
        fi
done
exit 0
```

#12_userdel1_function_case_if.sh

```
#!/bin/bash
# This script prompts for a user name and then deletes
# the corresponding account. Optionally, the user's
# home directory is deleted as well.
#yesno-Define function
yesno (){
        while   true
        do
                echo "$*"
                echo "Please answer by entering (y)es or (n)o:"
                read ANSWER
                case "$ANSWER" in
                        [yY] | [yY][eE][sS] )
                                return 0
                                ;;
                        [nN] | [nN][oO] )
                                return 1
                                ;;
                                echo "I can't understand you over here."
                                ;;
                esac
        done
}
read -p "Delete which user? " user
if      yesno "Also delete home directory of $user?"
then
        home=yes
fi
if yesno "Really delete user $user?"
then
        if test "$home" = yes
        then
                echo userdel -r $user
        else
                home="/home/$user"
                echo chown -R root.root $home
                echo userdel $user
        fi
fi
exit 0
```

#13_userdel2_getopts.sh

```
#!/bin/bash
# This script prompts for a user name and then deletes
# the corresponding account. Optionally, the user's
# home directory is deleted as well.
while getopts u:r variable
do
  case $variable in
     u ) user="$OPTARG" ;;
     r ) home=yes ;;
  esac
done
if test "$home" = yes
  then
    echo userdel -r $user
  else
    home="/home/$user"
    echo chown -R root.root $home
    echo userdel $user
fi
exit 0
```