## basic\_shell\_scripting\_lab

## exercise 1 - variables - aliases - functions

- Create a small script that asks the user for his or her name and age. then display the name and age of the user. (hint: use the echo command and the read command.)
- 2. Create an alias for each of the following commands:

```
rm -i
cd ..
cd ../../..
ls -lah
mv -i
```

3. Create a small script called **s1** that has a **function** that creates a directory and cd's into that directory.

When you run the script you give the new directory name as the first positional parameter.

This is a possible solution:

```
#!/usr/bin/bash
#this is the function mkcd
function mkcd {
 mkdir -p "$1" ; cd "$1"
}
#this is the function call
mkcd $1
```

Make the script executable and run it with the name of a new directory as the first positional parameter. So you should run it like this:

./s1 newdir

4. Create a script with a function that asks the user to input a random number. The name of the function should be: **getinput** 

The function displays a prompt to enter a number, then read the number and after the user has entered the number it will display the number.

The main part of the script should should look like this:

```
#main
echo "hello you will be asked for a random number 5 times."
getinput
getinput
getinput
getinput
getinput
```

exercise 2 - test - while

#!/bin/bash

# Initialize number to 0
<this is where you initialize the number to 0>

# Loop until the user enters a number greater than 10 <this is where you create the while loop and get the input>

echo "You entered \$num, which is greater than 10. Exiting..."

Try it... if you need help...see underneath.

```
#!/bin/bash
# Initialize number
num=0
# Loop until the user enters a number greater than 10
while test $num -le 10
    do
        echo -n "Enter a number greater than 10: "
        read num
done
```

echo "You entered \$num, which is greater than 10. Exiting..."

**Explanation:** 

- 1. test \$num -le 10: Checks if num is less than or equal to 10.
- 2. while test \$num -le 10: Repeats the loop until the condition becomes false.
- 3. Inside the loop, the script prompts the user to enter a number.
- 4. When the user enters a number greater than 10, the loop exits.

One more exercise with while and test. Please inspect this script. What does it do?

```
#!/bin/bash
# Initialize counter
count=5
# Loop while count is greater than 0
while test $count -gt 0; do
    echo "Countdown: $count"
    count=$((count - 1)) #this substracts 1 from the count variable
    sleep 1 # Pause for 1 second
done
```

echo "Liftoff!"

5. Write a script that creates 200 empty files using the **for** loop.

```
Hints: use the touch command
to count from 1 to 200 use {1..200}
```

```
if you need help...
#!/bin/bash
for i in {1..200}
do
    touch "$i"
done
echo "200 empty files created successfully."
```

6. Write a script that displays the following menu and use the case command to execute the command that the user wants to run.

- 1. count the number of files on my system
- 2. count the number of users on my system
- 3. display all mounted filesystems
- enter your choice :

try this one on your own, or use chatgpt....

## Final exercise:

Write a script that sends a log entry to the /var/log/messages file when a user logs into the system.