

# BASH SHELL-SCRIPTING ASSIGNMENT

"Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print ""File exists"", otherwise print ""File not found"".

```
main.bash  myfile.txt  ⋮
1  #!/bin/bash
2
3  # Create the file for demonstration purposes (optional, depends on the online compiler)
4  touch myfile.txt
5
6  FILE="myfile.txt"
7
8  if [ -e "$FILE" ]; then
9      echo "File exists"
10 else
11     echo "File not found"
12 fi
13
14
```

input

File exists

...Program finished with exit code 0  
Press ENTER to exit console.

Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

```
main.bash
1  read n
2  if [ $n -eq 0 ]; then
3      echo "exit because you entered 0 "
4  else
5      until [ $n -eq 0 ]; do
6          if (($n%2==0)); then
7              echo "$n is even"
8          else
9              echo "$n is odd"
10         fi
11         read n
12     done
13     echo "exit because you entered 0"
14 fi
```

15  
15 is odd  
20  
20 is even  
68  
68 is even

Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

```
main.bash  f.txt  ⋮  f1.txt  ⋮  f3.txt  ⋮
1 linecount(){
2     file="$1"
3     lines=$(wc -l < "$file")
4     echo "no.of lines in file:$lines"
5
6 }
7 linecount f.txt
8 linecount f1.txt
9 linecount f3.txt
```

```
no.of lines in file:4
no.of lines in file:3
no.of lines in file:1

...Program finished with exit code 0
Press ENTER to exit console.
```

Assignment 4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains ""File1.txt").

```
main.bash  Testdir/File1.txt  Testdir/File10.txt  Testdir/File2.txt  Testdir/File3.txt  Testdir/File4.txt  Testdir/File5.txt  Testdir/File6.txt  Testdir/File7.txt  Testdir/File8.txt
1 mkdir Testdir
2 cd Testdir
3 for ((i=1; i<=10; i++));
4 do
5     filename="File$i.txt"
6     echo $filename > $filename
7 done
```

Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.

Add a debugging mode that prints additional information when enabled.

```
main.bash TestDir/File1.txt TestDir/File10.txt TestDir/File2.txt TestDir/File3.txt TestDir/File4.txt TestDir/File5.txt TestDir/File6.txt TestDir/File7.txt TestDir/File8.txt
1  DEBUG=false
2  if [ "$DEBUG" = "true" ]; then
3      set -x
4  fi
5  # Function to print debug messages
6  debug_msg() {
7      if [ "$DEBUG" = "true" ]; then
8          echo "[DEBUG] $1"
9      fi
10 }
11 # Create a directory named TestDir
12 debug_msg "Creating directory TestDir"
13 if ! mkdir -p TestDir 2>/dev/null; then
14     echo "Error: Could not create directory TestDir"
15     exit 1
16 fi
17
18 # Change the current working directory to TestDir
19 debug_msg "Changing directory to TestDir"
20 if ! cd TestDir 2>/dev/null; then
21     echo "Error: Could not change to directory TestDir"
22     exit 1
23 fi
24
25 # Create ten files named File1.txt, File2.txt, ..., File10.txt
26 for i in {1..10}
27 do
28     filename="File$i.txt"
29     debug_msg "Creating file $filename with content $filename"
30     if ! echo $filename > $filename; then
31         echo "Error: Could not create file $filename"
32         exit 1
33     fi
34 done
35 debug_msg "Script completed successfully"
```

Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

Data Processing with sed

```
main.bash sample.log ⋮
1 logfile="sample.log"
2
3 grep "ERROR" "$logfile" | awk '{print $1, $2, $NF}'|

2024-05-19 12:35:01 application
2024-05-19 12:35:15 occurred

...Program finished with exit code 0
Press ENTER to exit console.
```

```
main.bash sample.log ⋮
1 2024-05-19 12:34:56 INFO Some informational message
2 2024-05-19 12:35:01 ERROR An error occurred in the application
3 2024-05-19 12:35:10 WARNING A warning message
4 2024-05-19 12:35:15 ERROR Another error occurred

2024-05-19 12:35:01 application
2024-05-19 12:35:15 occurred

...Program finished with exit code 0
Press ENTER to exit console.␣
```

Assignment 7: Create a script that takes a text file and replaces all occurrences of "old\_text" with "new\_text". Use sed to perform this operation and output the result to a new file.

main.bash	file1.txt	:	file2.txt	:
-----------	-----------	---	-----------	---

```
1 input="$1"
2 old="$2"
3 new="$3"
4 out="output_$(basename "$input")"
5 if [ ! -f "$input" ]; then
6     echo "Error: File not found!"
7     exit 1
8 fi
9 sed "s/$old/$new/g" "$input" > "$out"
```

✓ ↗ ⚙️ 📋

```
Error: File not found!

...Program finished with exit code 1
Press ENTER to exit console.
```