

# Scripting\_Task

## Task01:

- Create a shell script to print the HTTP error code of guvi.in & print, the success/failure message based on the error code response

```
vineeth@vineeth-pc:~/Dev/devops-playground$ mkdir task04-scripting-task
vineeth@vineeth-pc:~/Dev/devops-playground$ cd task04-scripting-task/
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ vim scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls -la
total 12
drwxrwxr-x 2 vineeth vineeth 4096 May 13 08:54 .
drwxrwxr-x 7 vineeth vineeth 4096 May 13 08:46 ..
-rw-rw-r-- 1 vineeth vineeth 398 May 13 08:54 scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ chmod 764 scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls -la
total 12
drwxrwxr-x 2 vineeth vineeth 4096 May 13 08:54 .
drwxrwxr-x 7 vineeth vineeth 4096 May 13 08:46 ..
-rwxrw-r-- 1 vineeth vineeth 398 May 13 08:54 scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ./scripting-task.sh
HTTP status code: 301
./scripting-task.sh: line 13: conditional binary operator expected
./scripting-task.sh: line 13: syntax error near `200'
./scripting-task.sh: line 13: `if [[ "$HTTP_CODE"-ge 200 && "HTTP_CODE" -lt 300 ]]; then'
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ vim scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ./scripting-task.sh
HTTP status code: 301
Failure: Website returned an error.
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls
scripting-task.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$
```

HTTP\_CODE=\$(curl -o /dev/null -s -w "%{http\_code}" <https://guvi.in>):

- Uses `curl` to make a silent request to the site.
- `-o /dev/null`: don't save the body.
- `-s`: silent mode (no progress bar).
- `-w "%{http_code}"`: write only the HTTP response code.
- `$(...)`: captures the output into variable `HTTP_CODE`.

echo "HTTP Status Code: \$HTTP\_CODE":

- Displays the status code.

if [[ "\$HTTP\_CODE" -ge 200 && "\$HTTP\_CODE" -lt 300 ]]; then:

- Checks if it's a 200 code (success).
- Otherwise prints(failure).

## **Task02:**

- Given a file, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome"

```
vineeth@vineeth-pc: ~/Dev/devops-playground/task04-scripting-task
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls
scripting-task01.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ vim sample.txt
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ vim scripting-task02.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls -la scripting-task02.sh
-rw-rw-r-- 1 vineeth vineeth 338 May 13 09:11 scripting-task02.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ chmod 764 scripting-task02.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ls -la scripting-task02.sh
-rwxrw-r-- 1 vineeth vineeth 338 May 13 09:11 scripting-task02.sh
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ ./scripting-task02.sh
Replacements done from line 5 in 'sample.txt'.
vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ cat scripting-task02.sh
#!/bin/bash

# Input file
FILE="sample.txt"

# Replace all occurrence of the word "give" with "learning"
awk 'NR<5 {print; next} /welcome/ {gsub("give", "learning")} {print}' "$FILE" > temp.txt

# Replace original file with updated content
mv temp.txt "$FILE"

# Confirmation message
echo "Replacements done from line 5 in '$FILE'."

vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$ cat sample.txt
Hello there, welcome to the tutorial.
You should give it a try.
This line has nothing to change.
We hope you give your best.
welcome to the platform, learning it your all.
Another welcome message without the word.
This line has the word learning but no welcome.
Final line to learning a proper welcome and learning again.

vineeth@vineeth-pc:~/Dev/devops-playground/task04-scripting-task$
```

FILE="sample.txt":

- Set a sample file to update the content

```
awk 'NR<5 {print; next} /welcome/ {gsub("give", "learning")} {print}' "$FILE" > temp.txt
```

- `awk` for text processing
- `NR<5 {print; next}` → For lines 1 to 4, print as-is and skip further checks.
- `/welcome/ {gsub("give", "learning")}` → If the line contains "welcome", replace all "give" with "learning".
- `{print}` → Print every line (modified or not).
- Output goes to `temp.txt`.

```
mv temp.txt "$FILE"
```

- Replace the original file with the modified one.