

# Shell Scripts: Art of Automation

Diptesh Kanojia  
IITB-Monash Research Academy

e-Yantra Team  
ERTS Lab, IIT Bombay

IIT Bombay  
March 29, 2021



# Agenda for Discussion

## 1 Shell Scripts

- Why Shell Scripts ?
- Creating Shell Scripts
- Art of Repetition
- Storing output



# Boon of Re-usability

We will see the power of shell as programming environment. We would like to store the commands we use frequently and store in a files so that we can re-run the operations by typing single-command. Thus, such a file which stores shell commands are called **shell scripts**.

Consider this scenario:

You have 1000 files (.csv files) inside ~/Desktop/project/data/ containing hospital records. Each line in the file is record of a single patient. You want to perform some analysis on these files. But, you want to analyze the files in the order such that files which contains smallest number of patients are analyzed first.



# Boon of Re-usability

How to sort files based on the number of lines?

Using multiple commands: run inside ~/Desktop/project/data/

```
wc -l *.csv | sort -n
```

## Output

You should see number of lines along with names of files in increasing order.



# Boon of Re-usability

To save the commands ran previously for later use, we open a new file `hospital.sh` and paste `wc -l *.csv | sort -n` inside it and save.

Using `hospital.sh`: run inside `~/Desktop/project/data/`

```
bash hospital.sh
```

## Output

You should see number of lines along with names of files in increasing order.



# Boon of Re-usability

Let's say that we want to support files with other file extensions (e.g. .txt, .tsv, .csv, etc.) by passing the files as the command line argument. To save the commands ran previously for later use, we open a new file `hospital.sh` and paste `wc -l "$@" | sort -n` inside it and save. Assume that instead of 1000 .csv files we have .txt files.

Using `hospital.sh` for .txt files: run inside `~/Desktop/project/data/`  
`bash hospital.sh *.txt`

## Output

You should see number of lines along with names of files in increasing order.

**NOTE:** "\$@" refers to all of a shell script's command-line arguments.



# Repetition using loops

Let's say that we want to support files with other file extensions (e.g. .txt, .tsv, .csv, etc.) by passing the files as the command line argument. `code.py` calculates some statistics for each file. How to write a shell script to perform the operation and display the output on the terminal.

Contents of `hospital.sh` is shown below:

```
# Bash Script to calculate statistics using `code.py`  
for file in "$@"  
do  
    echo $file  
    python code.py $file  
done
```

Figure 1: Contents of `hospital.sh`



# Repetition using loops

Let's say that we want to support files with other file extensions (e.g. .txt, .tsv, .csv, etc.). By passing the files as the command line argument. code.py calculates some statistics for each file. How to write a shell script to perform the operation and display the output on the terminal.

Using hospital.sh for .txt files: run inside ~/Desktop/project/data/  
`bash hospital.sh *.txt`

## Output

You should see the file name along with statistics calculate by code.py for each file displayed on the terminal.





# Saving output to a file

Let's say that we want to support files with other file extensions (e.g. .txt, .tsv, .csv, etc.). By passing the files as the command line argument. `code.py` calculates some statistics for each file. How to write a shell script to perform the operation and append the output into a file called `statistics.txt`. Where each line corresponds to statistics to each file.

Using `hospital.sh` for .txt files: run inside `~/Desktop/project/data/`  

```
bash hospital.sh *.txt >> statistics.txt
```

## Output

You won't see anything on the terminal. Inside `statistics.txt`, you should see the file name along with statistics calculate by `code.py` for each file displayed on the terminal.

**NOTE:** `>>` appends the output for every file to `statistics.txt`



# References

- [Shellscript Tutorial](#)
- [LearnShell.org](#)
- [TLDR pages for Shell commands](#)
- [e-Yantra Homepage](#)



# Thank You!

Author: Diptesh Kanojia  
Contributor: Prashant Sharma

Post your queries at: [resources@e-yantra.org](mailto:resources@e-yantra.org)

