DATA 3401
Python for Data Science

Unix Shell & Version Control System

Dr. Max (Masoud) Rostami

Department of Science / College of Science
Data Science Program / College of Science
The University of Texas Arlington, Texas


display:block;position:absol
opacity:1;*top:-2px;*left:-5px;
top:-4px\0/;left:-6px\0/;rig
inline-box;display:inline-block;fo
display:block;list-style:none;
inline-block;line-height:27px;padd
cursor:pointer;display:block;text-de
relative;z-index:1000}.gbts{*displ
padding-right:9px}#gbz .gbzt

Unix shell

Objectives:

- Recognize the importance of the Shell terminal for a Data Scientists.
- Operate with a Shell terminal using multiple commands.
- Practice various commands to perform different operations like navigating directories, files organization, and ….

Origins and Development:

1. Unix was developed at **Bell Labs** in the late 1960s by a team led by Ken Thompson and Dennis Ritchie. It was initially designed to meet the needs of multitasking, multi-user computing.
2. Unix was built around a set of principles, including simplicity, modularity, and the idea of treating everything as a file. This design philosophy made Unix highly flexible and scalable.

Command-line shell offers several advantages over graphical user interfaces (GUIs).

1.Increased Efficiency:

2.Flexibility and Power:

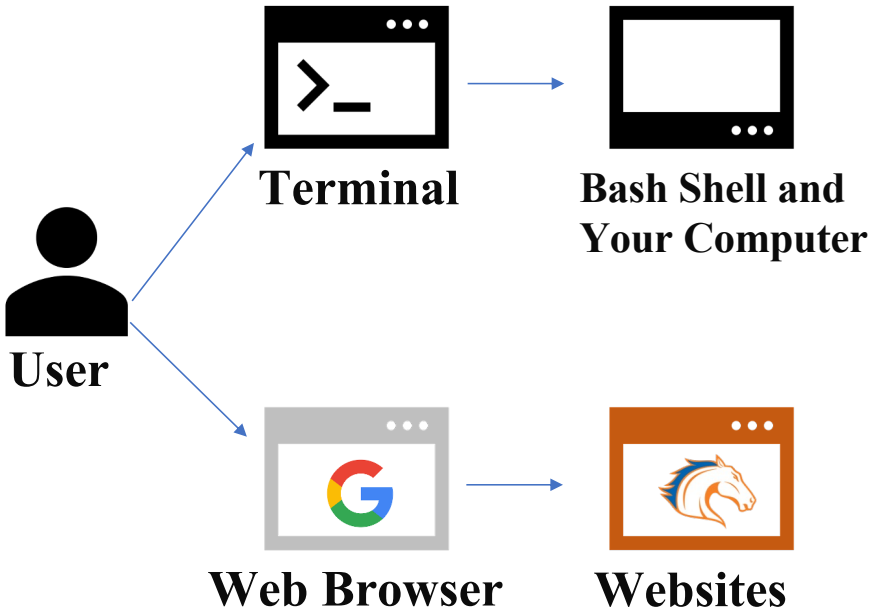
3.Automation and Scripting:

4.Remote System Management:

5.Resource Efficiency:

6.Reproducibility and Version Control:

Terminal


Terminal
Bash Shell and
Your Computer
User
Web Browser
Websites

Absolute and Relative paths

Absolute Path

An absolute path is the complete, exact location of a file or a directory, starting from the root directory.

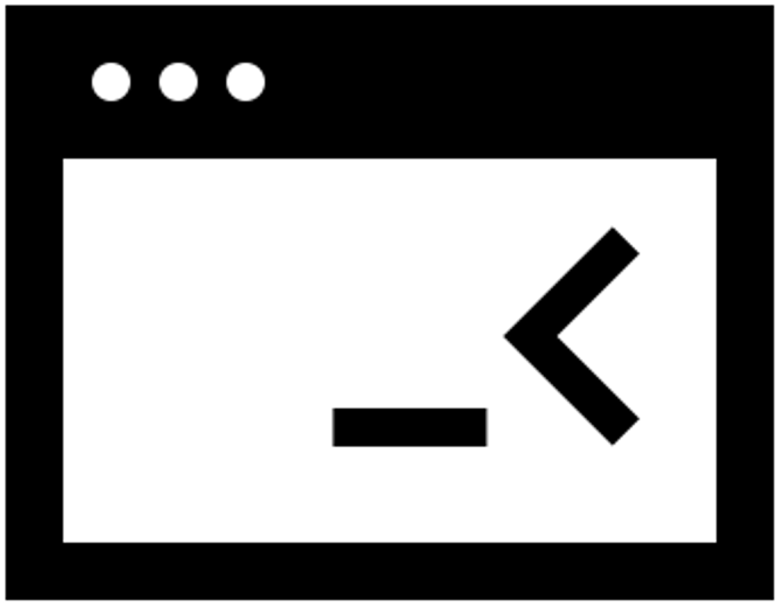
```
/home/username/documents/example.txt
```

Relative Path

A relative path, on the other hand, is defined in relation to the current working directory. It's like giving directions from your current location. If you're already in the "/home/username" directory, you can access the "example.txt" file using the relat

```bash
documents/example.txt
```

Let's Start



Windows: Installing Git Bash

(Windows-only! Mac and Linux users, skip this part)

https://git-scm.com/download/win


git
bash

Echo (echo):

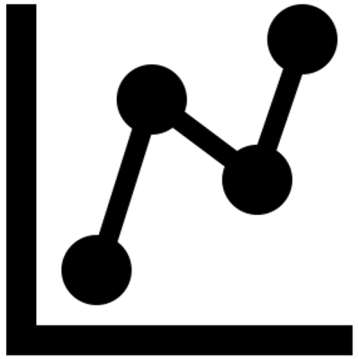
This command is used to display or print a line of text or string.

```
$ echo "Hello, World!"
```

File Operations

Managing your directories and file structure in your computer or servers is an important skill you need to use in the data science pipelines.

Here, I will show you how to use the ***Shell*** with multiple commands to navigate and organize your files.



1. Navigating directories

pwd (Print Working Directory):

Prints the current working directory.

```shell
$ pwd
```

cd (Change Directory):

Changes the current working directory.

Go to a specific directory: To go to a specific directory, you can specify its path. For instance, to go to the Desktop directory from your home directory, you can do:

Go to Home directory: By default, cd without any arguments will take you to your home directory.

```
cd
```

Go to the home directory with a specific path: To go to a specific directory from your home directory no matter where you currently are, you can prefix the path with ~.

```
cd ~
```

Go back to the previous directory: If you want to go back to the previous directory you were in, you can use - as an argument to cd.

```
cd -
```

Go to the parent directory: If you want to go to the parent directory of your current location, you can use .. as an argument to cd.

```
cd ..
```

ls (List):

Lists all the files and directories in the current directory.

Some commands have parameters or options to help you get more information or change the default behavior of those commands.

ls -l: Displays long format listing, which includes file/directory permissions, number of links, owner, group, size, and time of last modification.

```
ls -l
```

ls -a: Lists all files, including hidden files (those whose names start with . in Unix-like operating systems).

```
ls -a
```

ls -t: Lists files sorted by time and date.

```
ls -t
```

Organizing your files

With the Shell, you can use commands to organize your files into directories, move files, copy or remove the files.

mkdir(Make Directory):

This command is used to create a new directory.

```
$ mkdir NewDirectory
```

rmdir (Remove Directory):

This command is used to delete a directory.

```
$ rmdir NewDirectory
```