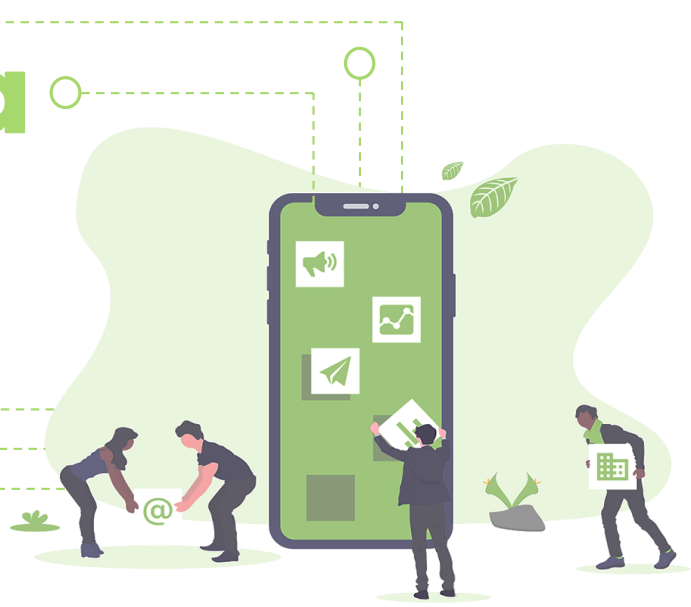
The Data Science Track



1

2. The Command Line Interface (CLI)

2

What is the Command Line Interface?

Nearly every computer comes with a CLI

- Windows: Git Bash
- Mac/Linux: Terminal

3

What can the CLI do?

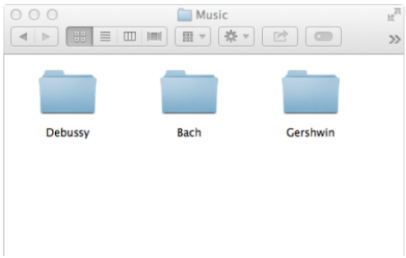
The CLI can help you:

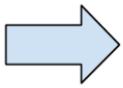
- Navigate folders
- Create files, folders, and programs
- Edit files, folders, and programs
- Run computer programs

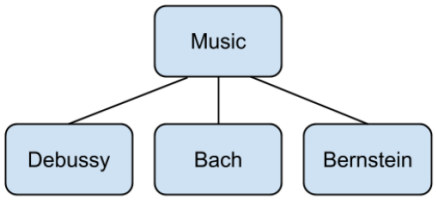
4

Basics of Directories

- "Directory" is just another name for folder
- Directories on your computer are organized like a tree
- Directories can be inside other directories
- We can navigate directories using the CLI



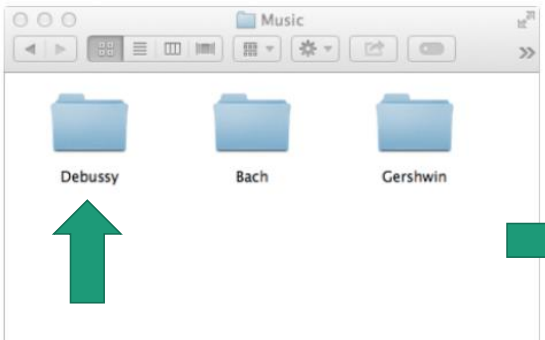


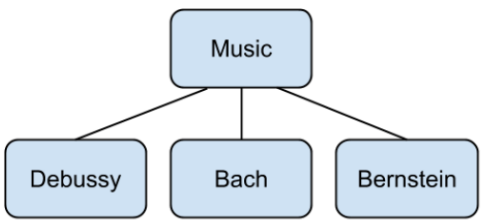


5

Basics of Directories

- The "Debussy" directory is contained inside of the "Music" directory.

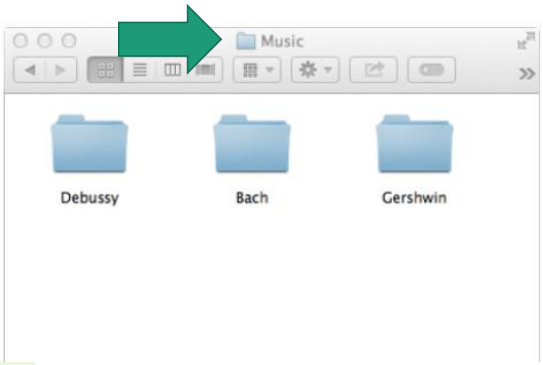


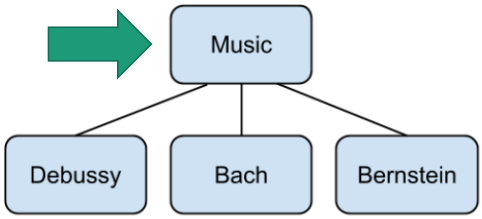


6

Basics of Directories

- The directory "up" from "Debussy" directory is the "Music" directory.

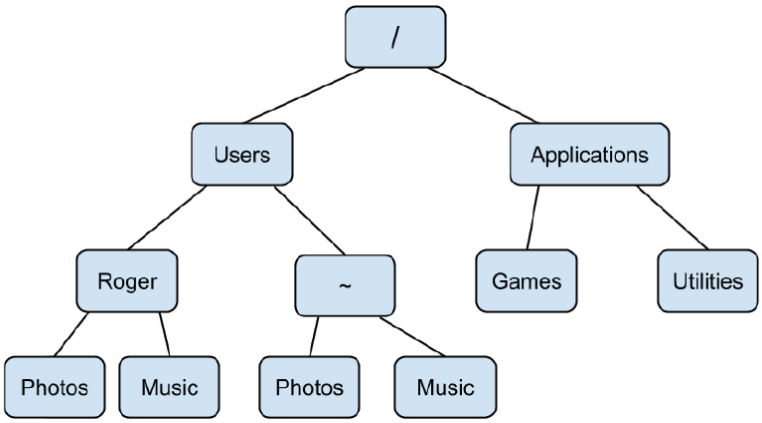




7

Your computer's directory structure

- The directory structure on your computer looks something like this:



8

Your computer's directory structure

Tree view of directories in Windows

Method

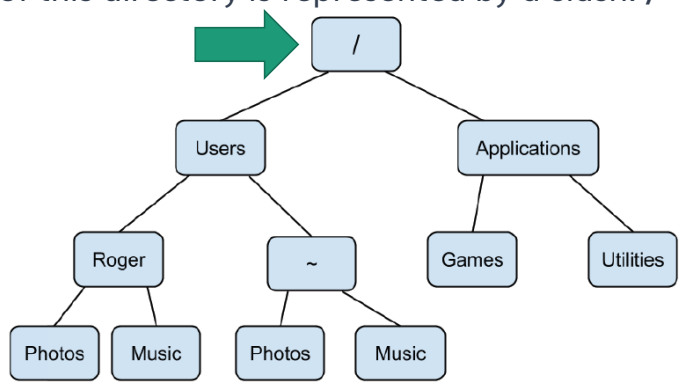
For Windows 8.1 or Windows 10, follow these steps:

1. Navigate into the folder in file explorer.
2. Press Shift, right-click mouse, and select "Open command window here".
3. Type `tree /f /a > tree.txt` and press Enter.
4. Open the new `tree.txt` file in your favourite text editor/viewer.

9

Special directories: root

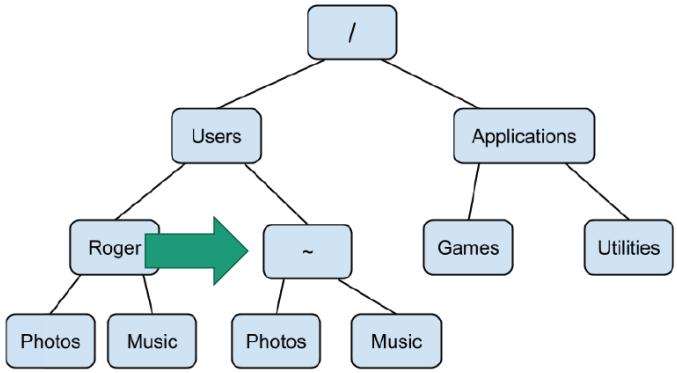
- The directory at the top of the tree is called the **root** directory
- The **root** directory contains all other directories
- The name of this directory is represented by a slash: /



10

Special directories: home

- Your **home** directory is represented by a tilde: **~**
- Your **home** directory usually contains most of your personal files



Navigating directories with the CLI

Windows Users:

- Open start menu
- Search for Git Bash
- Open Git Bash

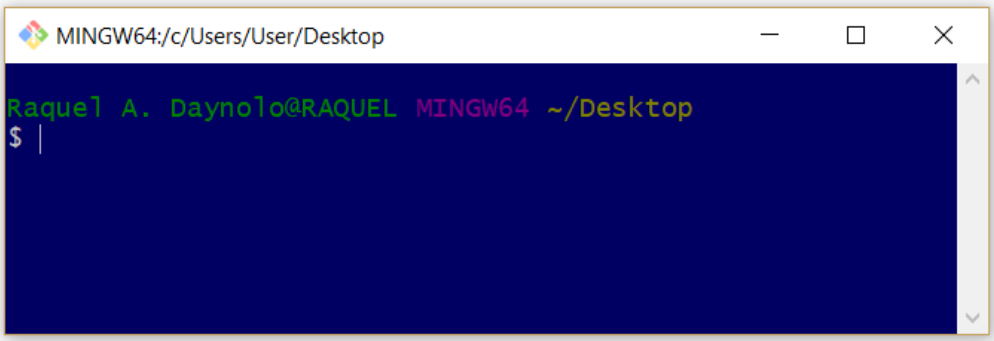
Navigating directories with the CLI

MAC Users:

- Open Spotlight
- Search Terminal
- Open Terminal

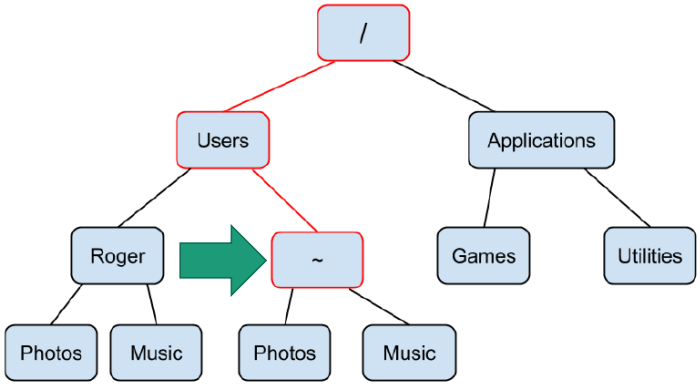
CLI Basics

- When you open your CLI you will see your prompt, which looks something like this:



CLI Basics

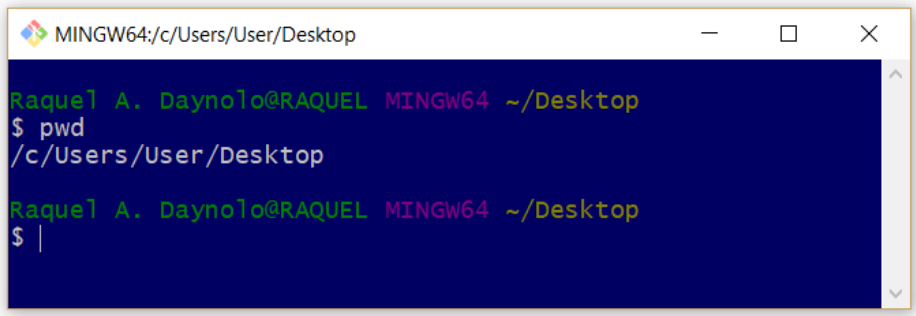
- You can imagine tracing all of the directories from your root directory to the directory you're currently in.
- This is called the "**path**" to your **working directory**.



15

CLI Basics

- In your CLI prompt, type `pwd` and press enter.
- This will display the path to your working directory.
- As you can see, we get the prompt back after entering a command.



16

CLI Commands

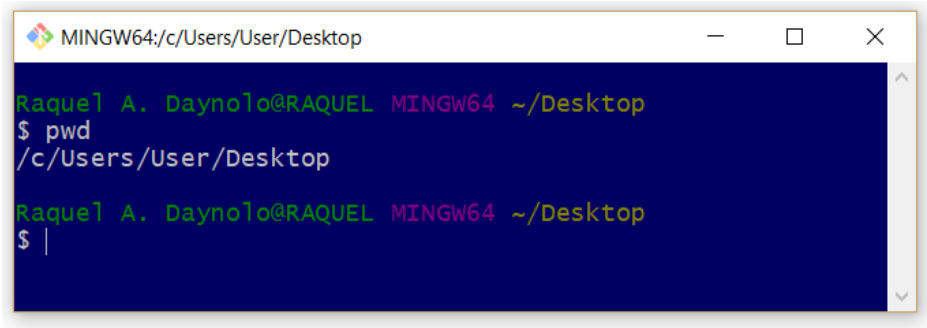
- You use the CLI prompt by typing in a command and pressing enter.
- `pwd` can be used at any time to display the path to your working directory
- `pwd` is an abbreviation for "print working directory"

CLI Commands

- CLI commands follow this format: **command flags arguments**
- **command** is the CLI command which does a specific task
- flags are options we give to the command to trigger certain behaviors, preceded by a dash (–)
- **arguments** can be what the **command** is going to modify, or other options for the **command**
- Depending on the **command**, there can be zero or more **flags** and **arguments**
- For example, `pwd` is a **command** that requires no **flags** or **arguments**

CLI Commands

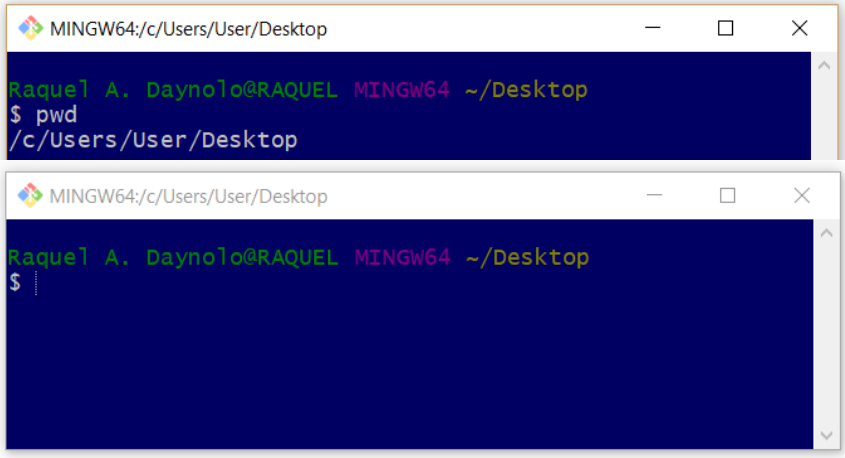
- `pwd` displays the path to the current working directory



19

CLI Commands

- `clear` will clear out the commands in your current CLI window

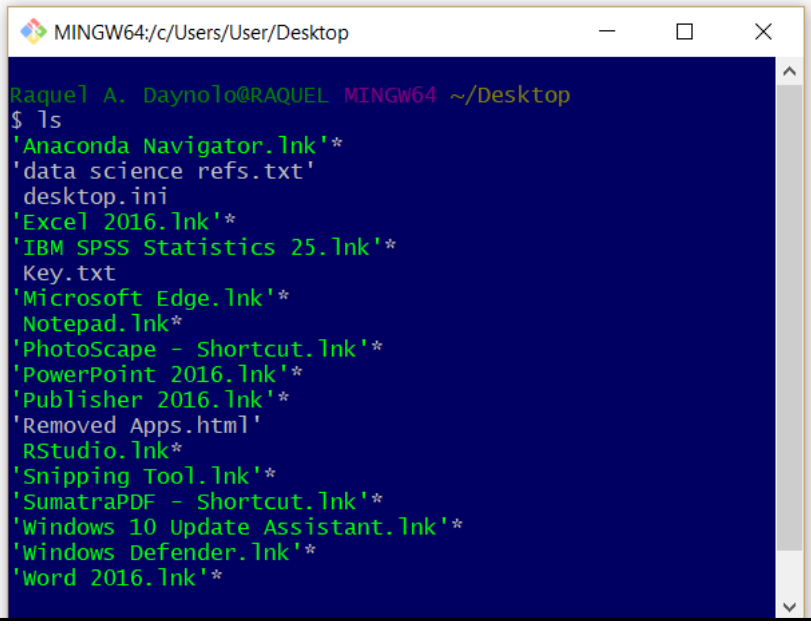


20

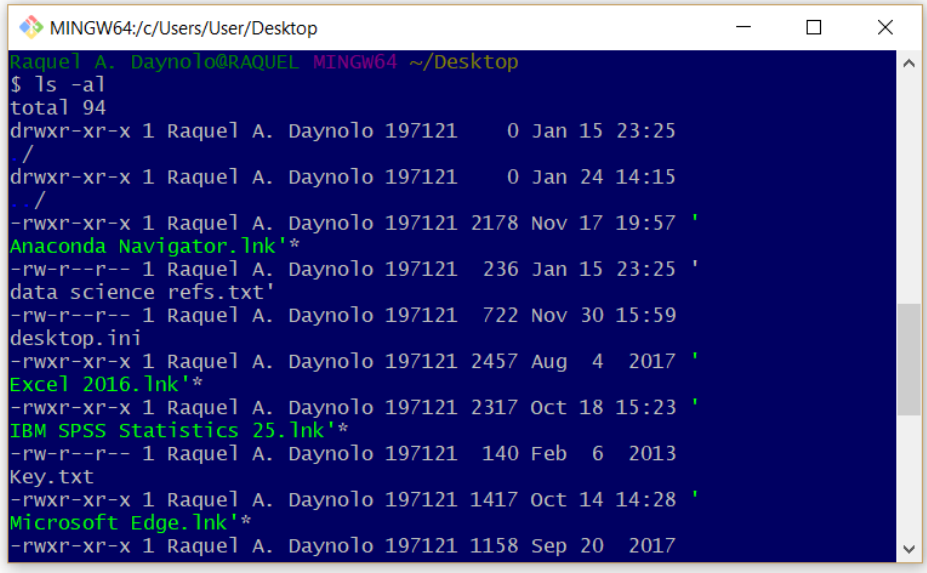
CLI Commands

- `ls` lists files and folders in the current directory
- `ls -a` lists hidden files and unhidden files and folders
- `ls -al` lists details for hidden and unhidden files and folders
- Notice that `-a` and `-l` are flags (they're preceded by a -)
- They can be combined into a flag: `-al`

CLI Commands



CLI Commands

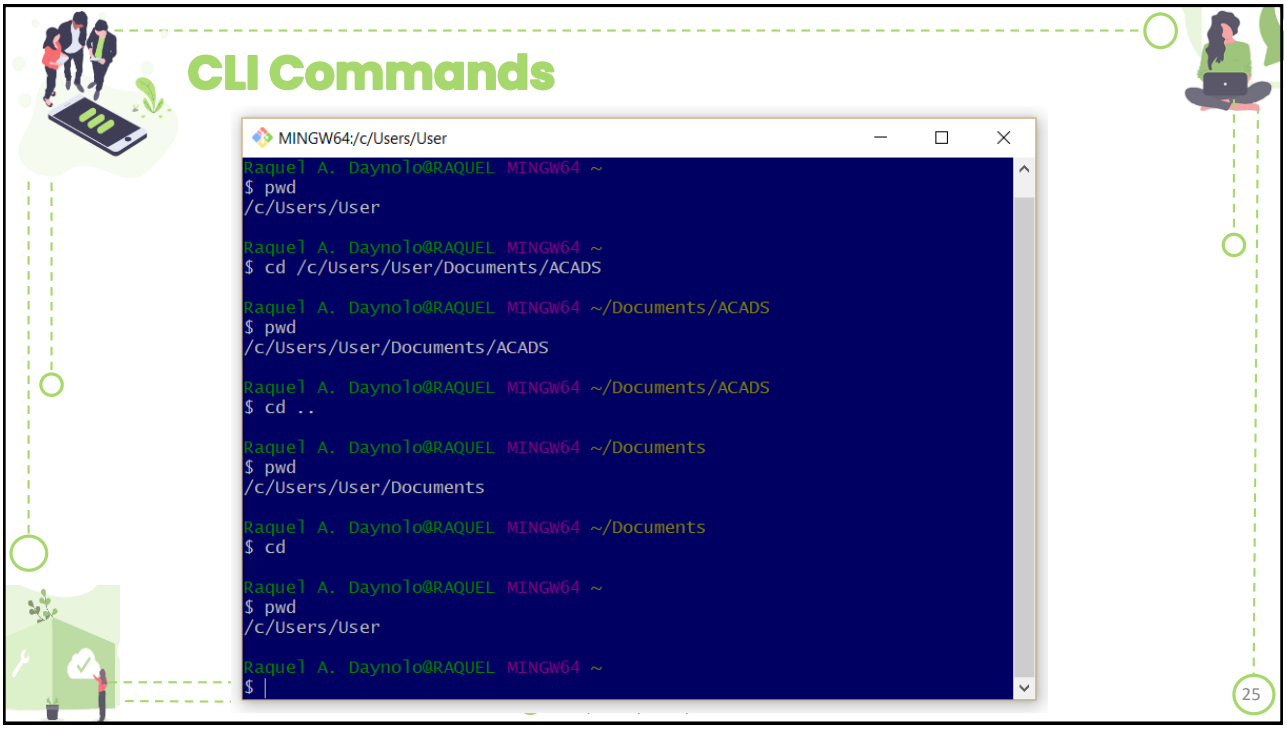


23

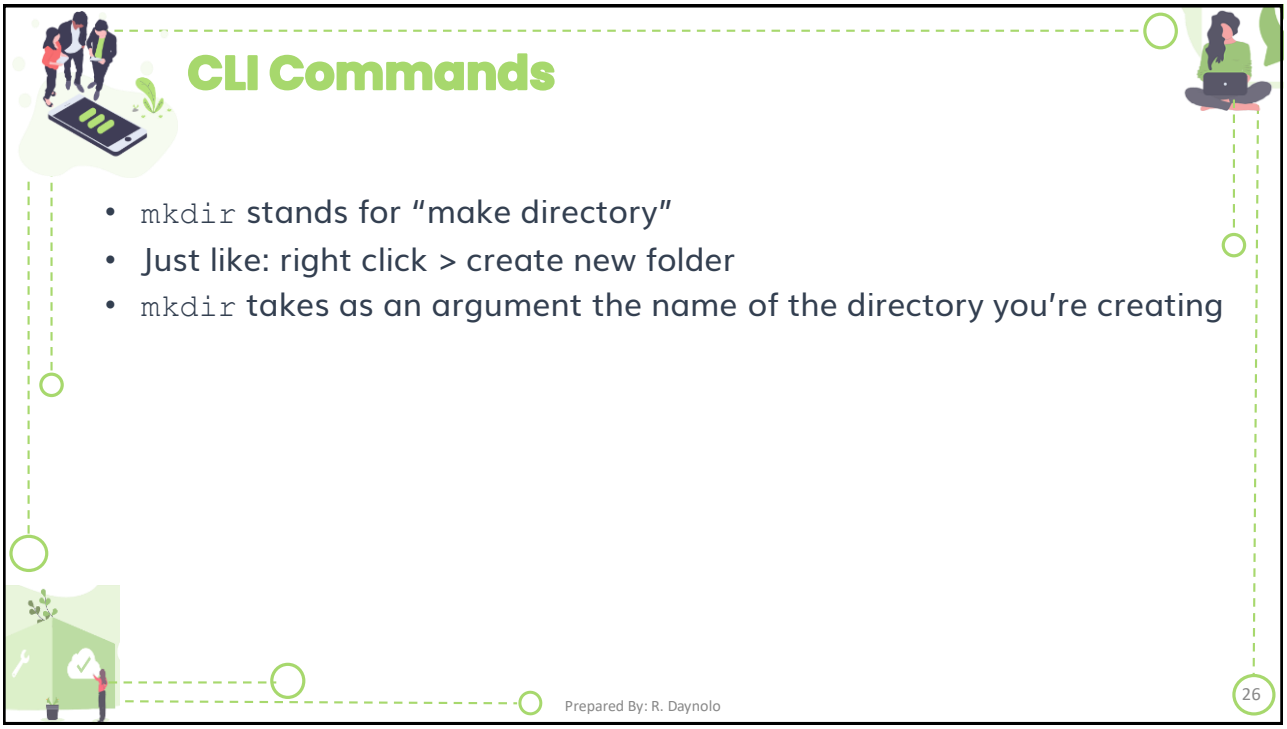
CLI Commands

- `cd` stands for “change directory”
- `cd` takes as an argument the directory you want to visit
- `cd` with no argument takes you to you home directory
- `cd ..` allows you to change directory to one level above your current directory

24

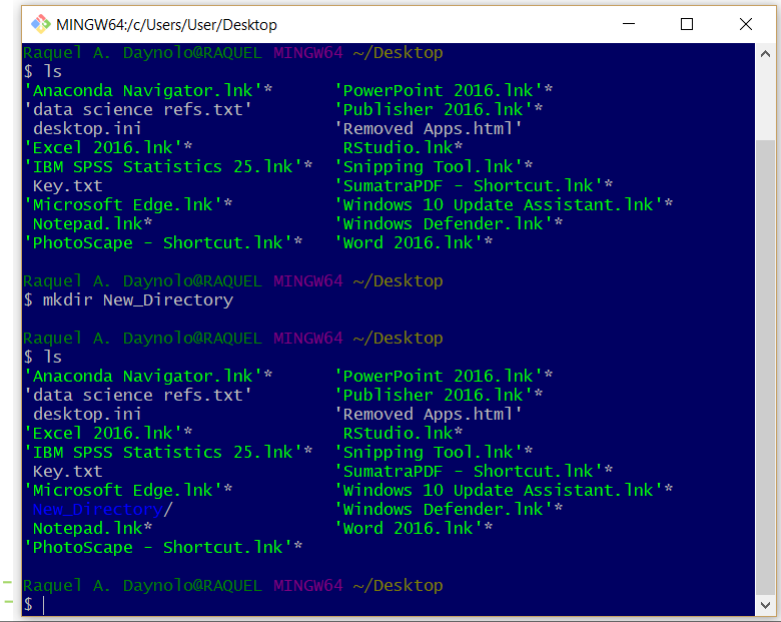

CLI Commands
MINGW64:/c/Users/User
Raquel A. Daynolo@RAQUEL MINGW64 ~
$ pwd
/c/Users/User
Raquel A. Daynolo@RAQUEL MINGW64 ~
$ cd /c/Users/User/Documents/ACADS
Raquel A. Daynolo@RAQUEL MINGW64 ~/Documents/ACADS
$ pwd
/c/Users/User/Documents/ACADS
Raquel A. Daynolo@RAQUEL MINGW64 ~/Documents/ACADS
$ cd ..
Raquel A. Daynolo@RAQUEL MINGW64 ~/Documents
$ pwd
/c/Users/User/Documents
Raquel A. Daynolo@RAQUEL MINGW64 ~/Documents
$ cd
Raquel A. Daynolo@RAQUEL MINGW64 ~
$ pwd
/c/Users/User
Raquel A. Daynolo@RAQUEL MINGW64 ~
$
25

25


CLI Commands
• mkdir stands for “make directory”
• Just like: right click > create new folder
• mkdir takes as an argument the name of the directory you’re creating
Prepared By: R. Daynolo
26

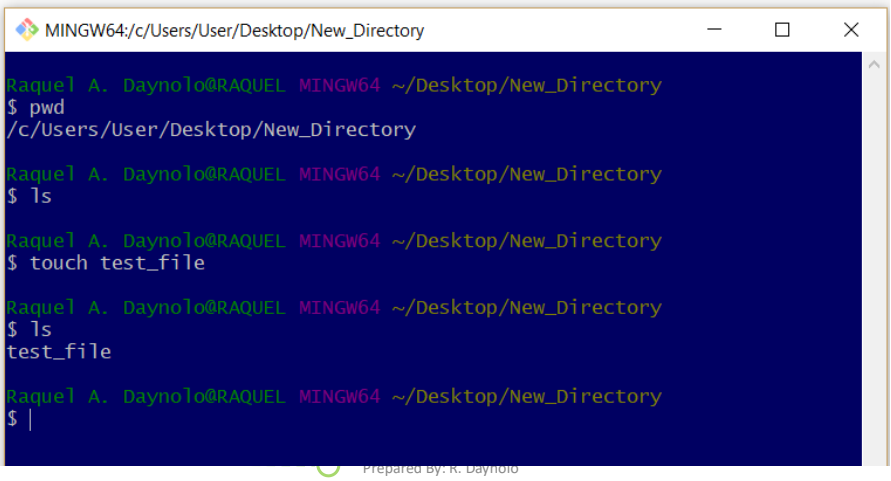
26

CLI Commands



CLI Commands

- `touch` creates an empty file



CLI Commands

- `cp` stands for "copy"
- `cp` takes as its first argument a file, and as its second argument the path to where you want the file to be copied

CLI Commands

MINGW64:/c/Users/User/Desktop

```
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/New_Directory
$ cp test_file /c/Users/User/Desktop

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/New_Directory
$ cd ..

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ ls
'Anaconda Navigator.lnk'*       'PowerPoint 2016.lnk'*
'data science refs.txt'         'Publisher 2016.lnk'*
 desktop.ini                    'Removed Apps.html'
'Excel 2016.lnk'*                RStudio.lnk*
'IBM SPSS Statistics 25.lnk'*   'Snipping Tool.lnk'*
 Key.txt                        'SumatraPDF - Shortcut.lnk'*
'Microsoft Edge.lnk'*            test_file
 New_Directory/                 'Windows 10 Update Assistant.lnk'*
 Notepad.lnk*                   'Windows Defender.lnk'*
'PhotoScape - Shortcut.lnk'*    'Word 2016.lnk'*

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$
```

CLI Commands

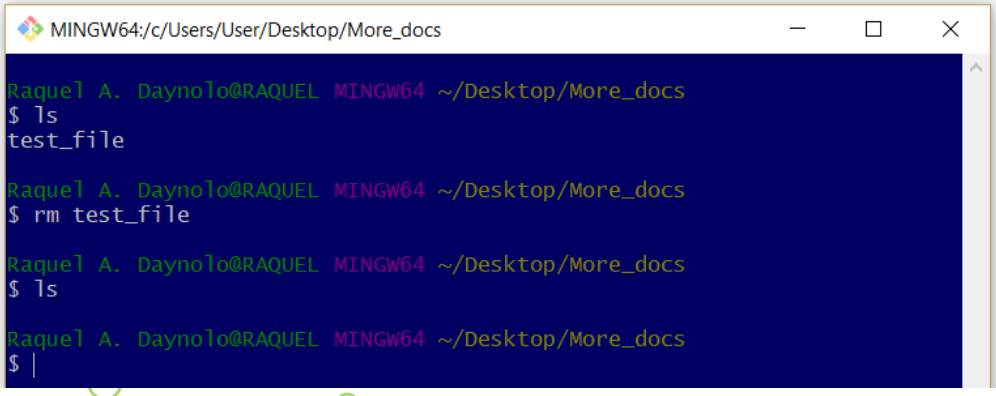
- `cp` can also be used for copying the contents of directories, but you must use the `-r` flag
- The line : `cp -r New_Directory/. More_docs/` copies the contents of **New_Directory** into **More_docs**

CLI Commands

```
MINGW64:/c/Users/User/Desktop/More_docs

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ mkdir More_docs

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ cp -r New_Directory/. More_docs/

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ cd /c/Users/User/Desktop/More_docs

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ ls
test_file

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$
```

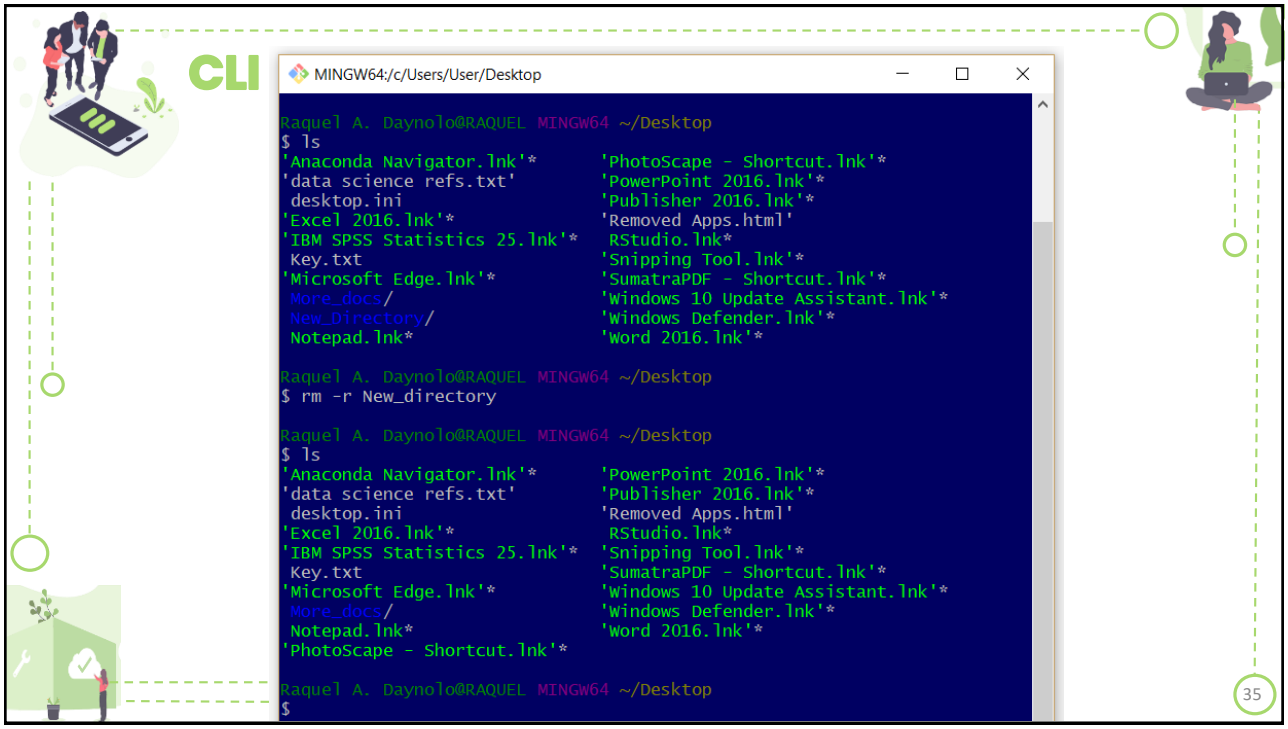
CLI Commands

- `rm` stands for “remove”
- `rm` takes the name of a file you wish to remove as its argument

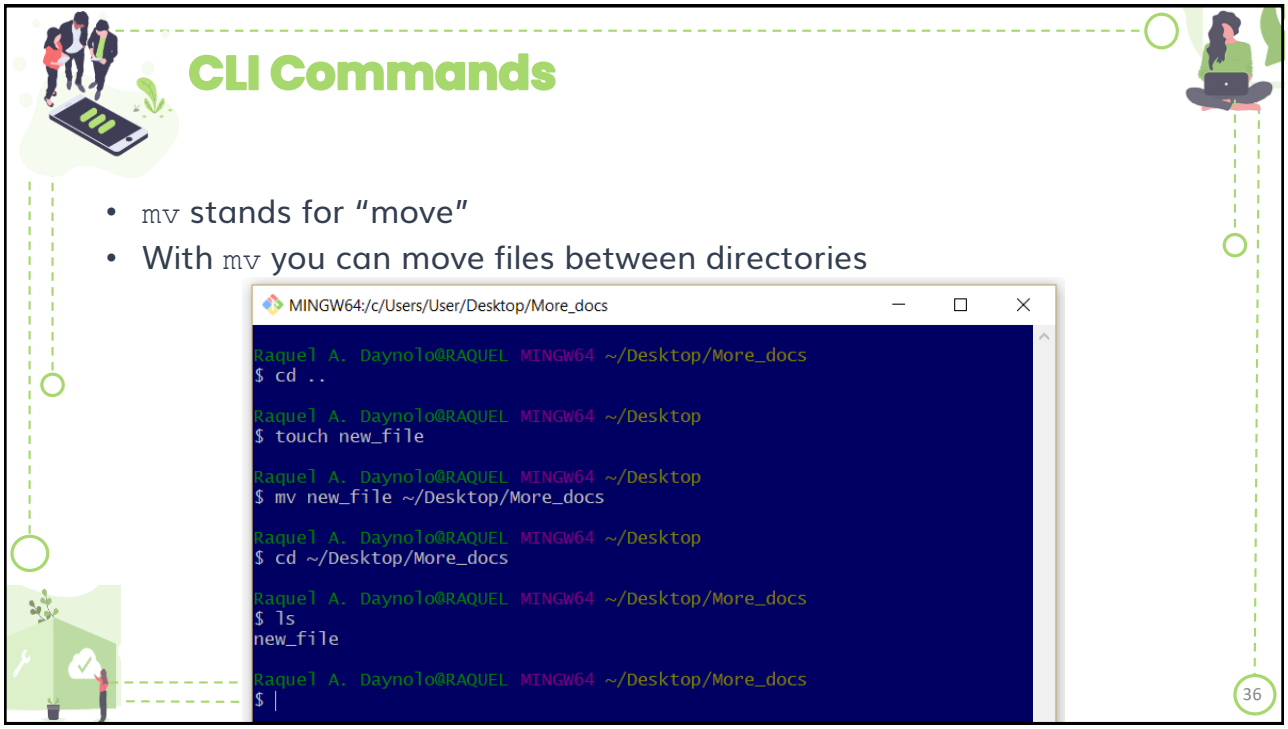


CLI Commands

- `rm` can also be used to delete entire directories and their contents by using the `-r` flag
- Be careful when you do this, there is no way to undo an `rm`


CLI
MINGW64:/c/Users/User/Desktop
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ ls
'Anaconda Navigator.lnk'*
'data science refs.txt'
desktop.ini
'Excel 2016.lnk'*
'IBM SPSS Statistics 25.lnk'*
Key.txt
'Microsoft Edge.lnk'*
More_docs/
New_Directory/
Notepad.lnk*
'PhotoScape - Shortcut.lnk'*
'PowerPoint 2016.lnk'*
'Publisher 2016.lnk'*
'Removed Apps.html'
RStudio.lnk*
'Snipping Tool.lnk'*
'SumatraPDF - Shortcut.lnk'*
'Windows 10 Update Assistant.lnk'*
'Windows Defender.lnk'*
'Word 2016.lnk'*
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ rm -r New_directory
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ ls
'Anaconda Navigator.lnk'*
'data science refs.txt'
desktop.ini
'Excel 2016.lnk'*
'IBM SPSS Statistics 25.lnk'*
Key.txt
'Microsoft Edge.lnk'*
More_docs/
Notepad.lnk*
'PhotoScape - Shortcut.lnk'*
'PowerPoint 2016.lnk'*
'Publisher 2016.lnk'*
'Removed Apps.html'
RStudio.lnk*
'Snipping Tool.lnk'*
'SumatraPDF - Shortcut.lnk'*
'Windows 10 Update Assistant.lnk'*
'Windows Defender.lnk'*
'Word 2016.lnk'*
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$
35

35


CLI Commands
• mv stands for “move”
• With mv you can move files between directories
MINGW64:/c/Users/User/Desktop/More_docs
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ cd ..
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ touch new_file
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ mv new_file ~/Desktop/More_docs
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop
$ cd ~/Desktop/More_docs
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ ls
new_file
Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$
36

36

CLI Commands

- You can also use `mv` to rename files

```
MINGW64:/c/Users/User/Desktop/More_docs

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ ls
new_file

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ mv new_file renamed_file

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ ls
renamed_file

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$
```

37

CLI Commands

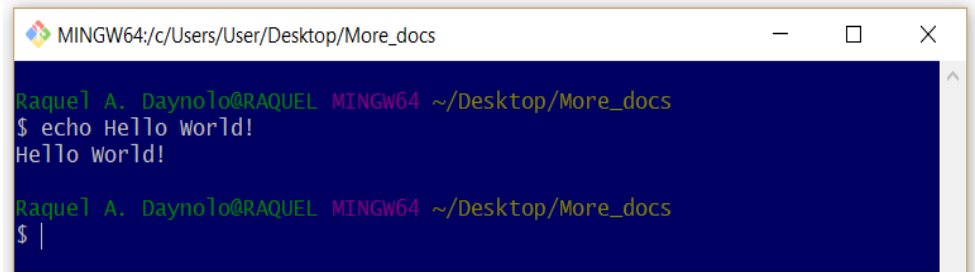
- `echo` will print whatever arguments you provide

```
MINGW64:/c/Users/User/Desktop/More_docs

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$ echo Hello World!
Hello World!

Raquel A. Daynolo@RAQUEL MINGW64 ~/Desktop/More_docs
$
```

38

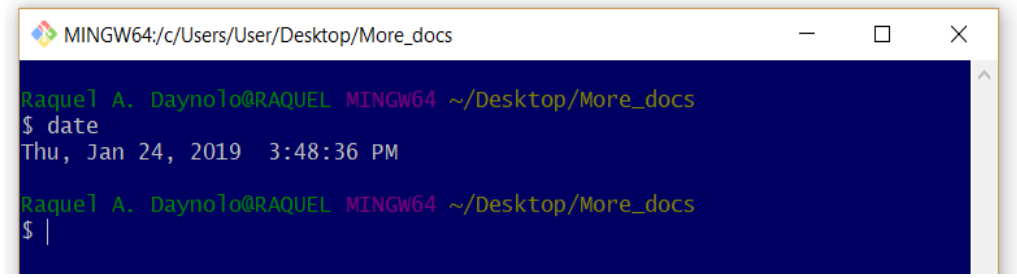
CLI Commands

- `echo` will print whatever arguments you provide



CLI Commands

- `date` will print today's date



Summary of Commands

- pwd
- clear
- ls
- cd
- mkdir
- touch
- cp
- rm
- mv
- echo
- date

3. Introduction to Git

What is Version Control?

"Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later."

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

43

What is Version Control?

- Many of us constantly create something, save it, change it, then save it again
- Version (or revision) control is a means of managing this process in a reliable and efficient way
- Especially important when collaborating with others

44

What is Git?

"Git is a free and open source distributed version control system (VCS) designed to handle everything from small to very large projects with speed and efficiency."

https://git-scm.com/

45

What is Git?

- Created by the same people who developed Linux
- The most popular implementation of version control today
- Everything is store in local repositories on your computer
- Operated from the command line

https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git

46

What is Git?

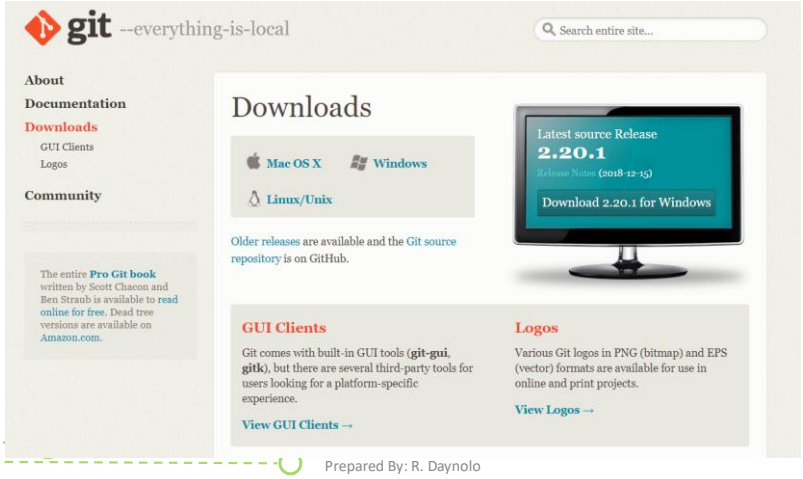
Concepts of Git

- ✓ Keeps track of code history
- ✓ Takes "snapshots" of you files
- ✓ You decide when to take a snapshot by making a "commit"
- ✓ You can visit any snapshot at any time
- ✓ You can stage files before committing

47

Download Git

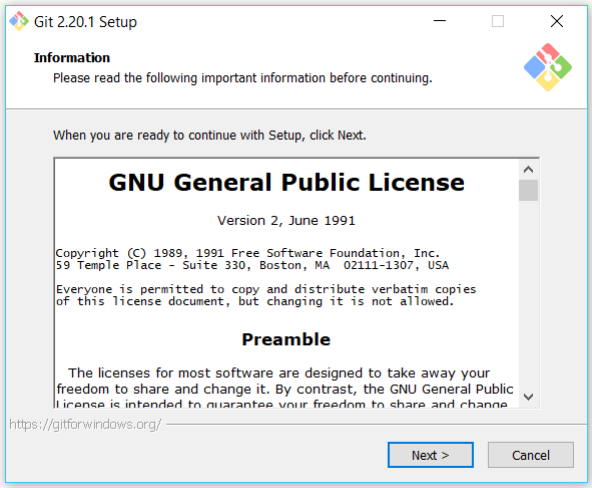
https://git-scm.com/downloads



48

Install Git

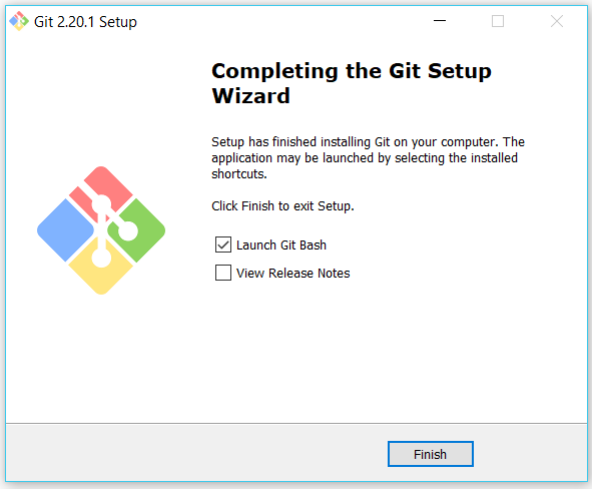
- Once the file is done downloading, open it to begin Git installation



49

Install Git

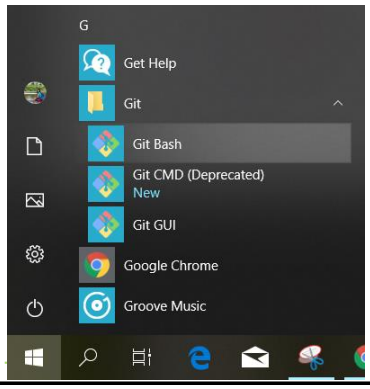
- Unless you really know what you're doing, just go with the default options at each step of the installation



50

Open Git Bash

- Find the program called Git Bash, which is the command line environment for interacting with Git
- It should be located in the directory into which Git was installed (or, for Windows users, in the Start Menu)



51

51

Open Git Bash

- Once Git Bash opens, you'll see a short welcome message followed by the name of your computer and a dollar sign on the next line
- The dollar sign means that it's your turn to type a command

52

Configure Username and Email

- Each commit to a Git repository will be "tagged" with the username of the person who made the commit
- Enter the following commands in Git Bash, one at a time, to set vour username and email:

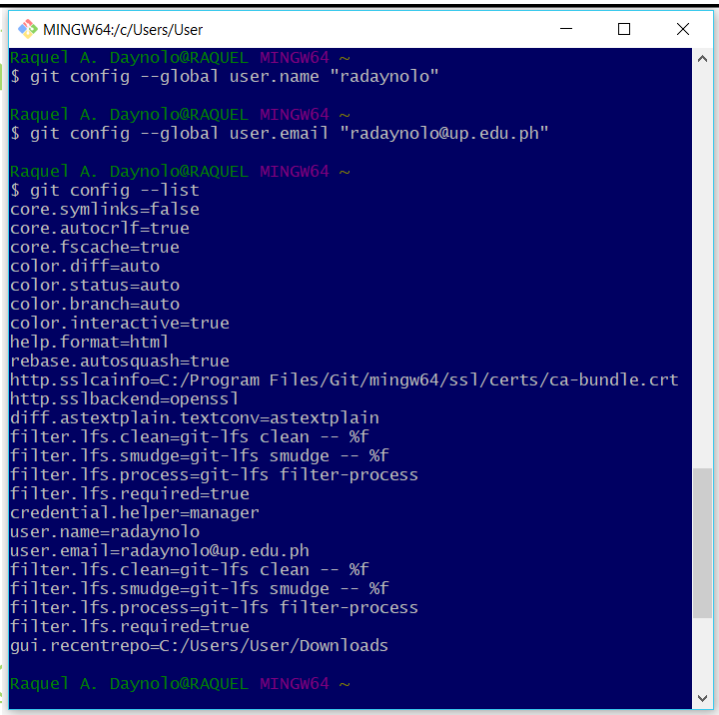
```
$ git config --global user.name "Your Name Here"
$ git config --global user.email "your_email@example.com"
```

- Now type the following to confirm your changes

```
$ git config --list
```

Con

- S



3. Introduction to GitHub

55

What is GitHub?

"GitHub is a web-based hosting service for software development projects that use the Git revision control system."

https://en.wikipedia.org/wiki/GitHub

56

The "Git" in GitHub

- Git is a version control system. When developers create something (an app, for example), they make constant changes to the code, releasing new versions up to and after the first official release.

The "Hub" in GitHub

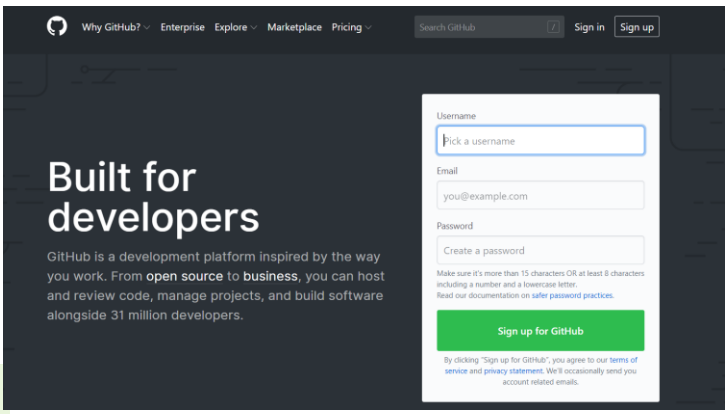
- Git is a command-line tool, but the center around which all things involving Git revolve is the hub—GitHub.com—where developers store their projects and network with like minded people.

What is GitHub?

- Allows users to “push” and “pull” their local repositories to and from remote repositories on the web
- Provides users with a homepage that displays their public repositories
- Users’ repositories are backed up on the GitHub server in case something happens to the local copies
- Social aspect allows users to follow one another and share projects

Set Up a GitHub Account

- Go to the GitHub homepage at https://github.com/



- Enter a username, email, and password and click “Sign up for GitHub”
- NOTE: You should use the same email address that you used when setting up Git.

Your GitHub Profile

- Your profile is where all of your activity on GitHub is displayed