

Data Science, lesson 2

Your Development Environment



Group Exercise: Where were we?

5 minutes



Any examples of supervised and unsupervised learning?

Deliverables:

- Split into three groups.
- Pick a location: coffeeshop, hotel, bank.
- Take some time to discuss and come up with three things you can analyse using supervised or unsupervised learning.



Your Development Environment

Using the Command Line

1. Create folders and files using the command line (`mkdir`, `touch`).
2. Change directories and list directory content (`cd`, `ls`).
3. Check the current working directory (`pwd`).

Learning objectives



Introduction to Git

1. Use and explain common Git commands, including `init`, `add`, `commit`.
2. Distinguish between local and remote repositories.
3. Create, copy, and delete repositories locally or on GitHub.
4. Clone and sync repositories using `push`, `pull`, and `clone`.

Data Science



Using the Command Line

Introduction

Terminology

CLI - command line interface, can also stand for command language interpreter. Sometimes referred to as shell, terminal, bash, prompt.

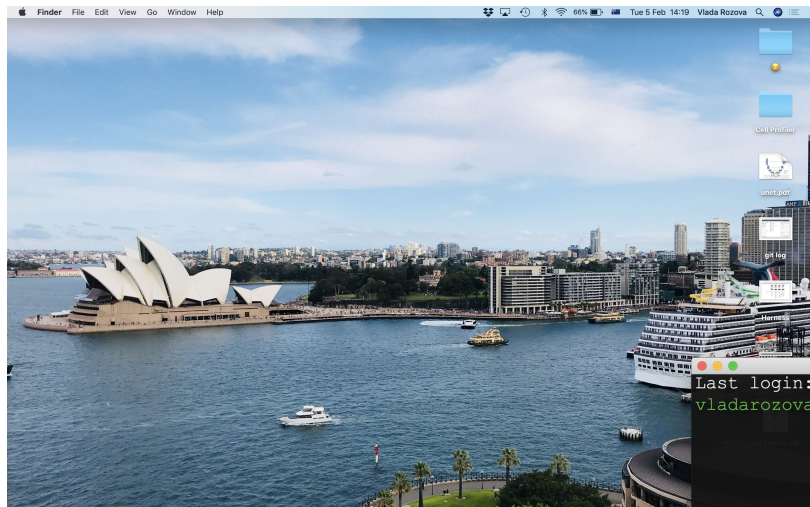
Kernels and Shells

When you turn on your computer, one of the first thing it loads is the **kernel**. It is the core process, that connects to the actual hardware and launches all other processes on your computer.

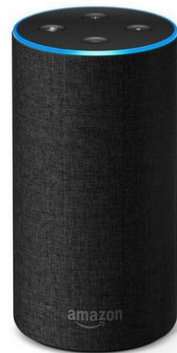
Surrounding the kernel is the **shell**. The shell is the interface to the outside world and provides a layer of protection to the kernel.



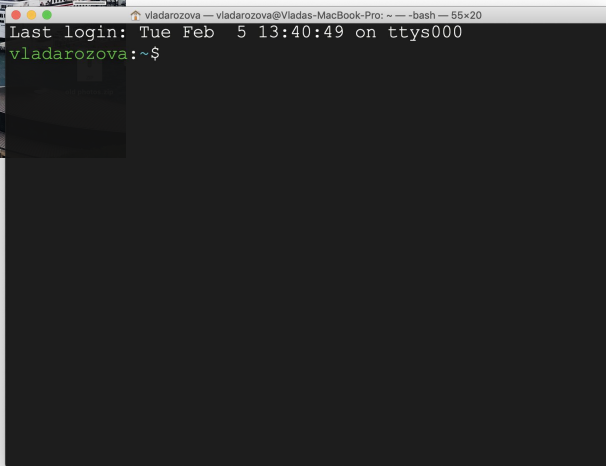
Types of Shells



**GUI or Graphical
User Interface**



**VUI or Voice User
Interface**



**CLI or Command
Line Interface**



Computers Out: Shell Commands

1 hour



During the next hour we will learn some basic shell commands that will allow us to navigate between directories, find, edit and examine files.

But first let's have some fun!

Try this: [Type like a hacker](#)

Now, open your Terminal and click [here](#).



Data Science

Introduction to Git

“

First thing first -- Git is not GitHub.

What is Git?

Git is:

- A distributed version control system.
- A program you run from the command line.

Programmers use Git so that they can keep a history of all changes made to their code. This means that they can roll back changes (or switch to older versions) as far back as when they started using Git in their project.

A code base in Git is referred to as a **repository**, or **repo**, for short.

Git was created by [Linus Torvalds](#), the principal developer of Linux.



git

What is GitHub?

GitHub is:

- A hosting service for Git repositories.
- A web interface to explore Git repositories.
- A social network of programmers.

Some other points to note:

- We all have individual GitHub accounts and store our code bases there.
- You can follow users and star your favorite projects.
- Developers can access code bases on other public accounts.
- GitHub uses Git.



What is GitHub Enterprise?

GitHub Enterprise is:

- A professional application of GitHub
- All repository data is stored on private and/or local machines and networks.

While GitHub is the **public**, 'Social Network' for programming and programmers, Github Enterprise is the **private**, professional application of GitHub.

Because, GitHub and GitHub Enterprise have a similar structure and are based off the git language, interacting with the two is almost identical.



Can you use Git without GitHub?

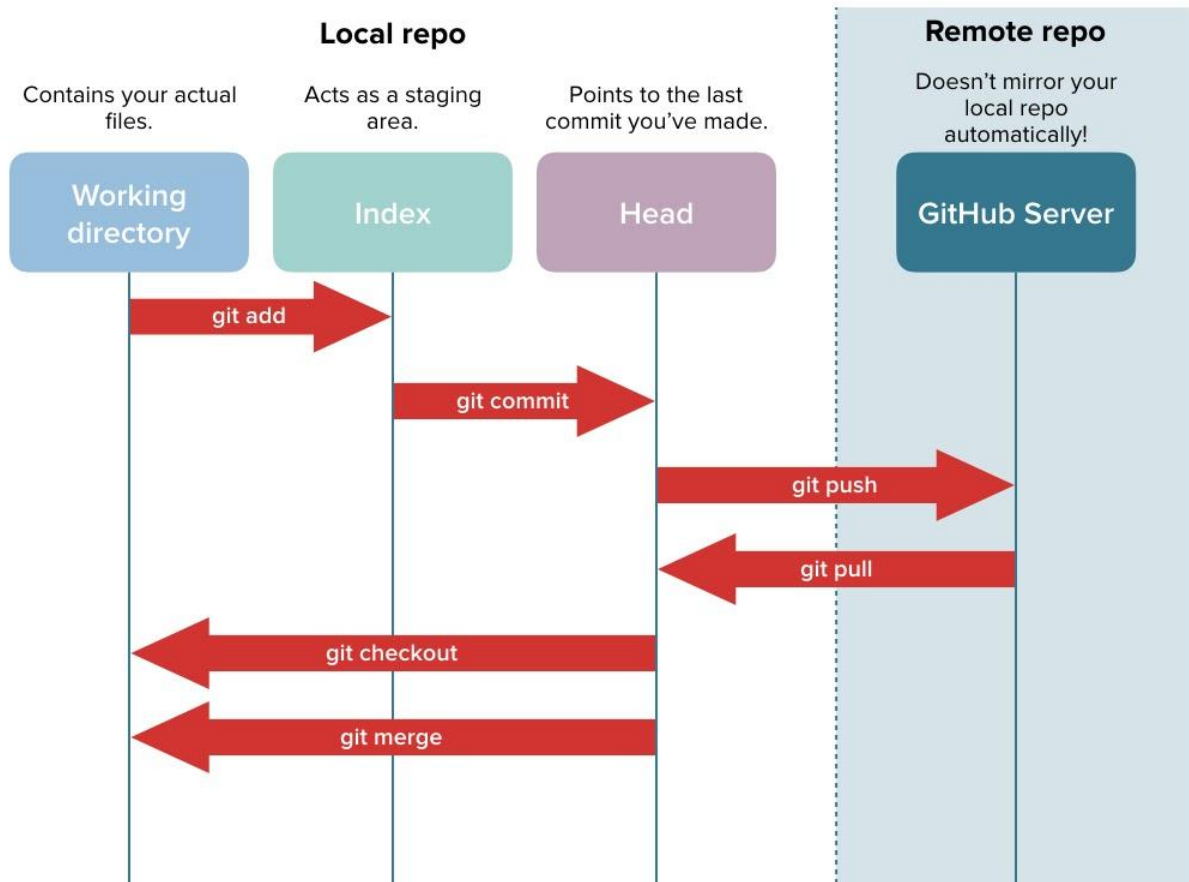
- A. Yes
- B. No.

Types of Repos

A **repository** (“**repo**,” for short) is a project that is under version control.

There are two types of repositories: **local** and **remote**. A local repository is a repo that is stored on your machine, whereas a remote repository is stored online via GitHub or GitHub Enterprise.

It’s common to create a new repo for **each project** you’re working on rather than have several projects lumped into a single repository.





Computers Out: Let's Git into it!

1 hour



Now let's make you feel comfortable
with Git and GitHub!

Open your Terminal and click [here](#).



Data Science



Conclusion



Discussion: Important Takeaways

5 minutes



Let's Debrief

- What did you learn in this lesson?
- What was challenging about this topic?
- What questions do you still have?
- What are some next steps you could take with this information?



Finish That Sentence

What are your biggest takeaways from today?



“Something that really got me thinking is...”

“The best thing I got out of today is...”

“I discovered...”

“I still want to learn about...”

“I was surprised that...”

Ask Me Anything!

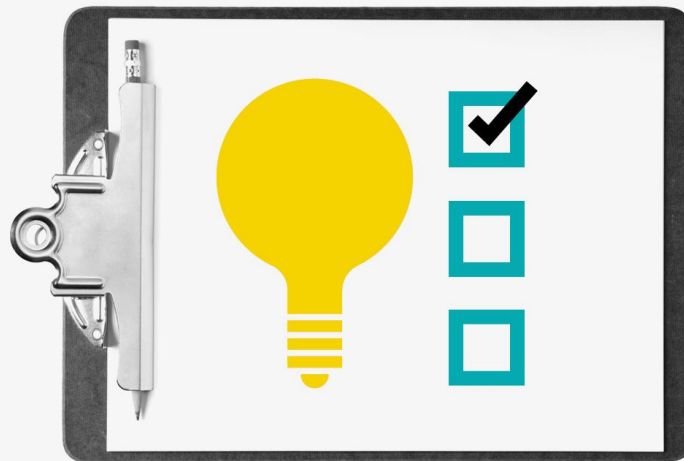


Before Next Class:

- A. Get into habit of using Git and GitHub**
- B. Practice your Python!**

A Few Good References

- [More on how to use GitHub](#)
- [Command Line Cheat Sheet](#)



“

**Don't Forget Your
Exit Tickets!**

See you next time!



Thank you!

