OVERVIEW OF MODERN DATA SCIENCE LANDSCAPE

ERT 474/574

Open-Source Hydro Data Analytics

August 27, 2025

University at Buffalo The State University of New York

Logistics

- Please sign up for a GitHub account and send your usernames to UBLearns (homework #1)
- Difficulties in signing up for the **GitHub Student Developer Pack**?
    - Has anyone successfully signed up?
    - If so, please show your experience in today’s lab session!

Learning objectives

- Coding languages?
  - Python? Matlab? Fortran? C++?
  - Where are different languages used in hydrologic models?
  - What is the best practice to learn the coding languages?
- GitHub

Coding languages

Python

Matlab

Fortran

C++

More language?

Coding languages

Python

- Python is the most widely used high-productivity language in Scientific Computing.

Matlab

Fortran

C++

More language?

Coding languages

Python

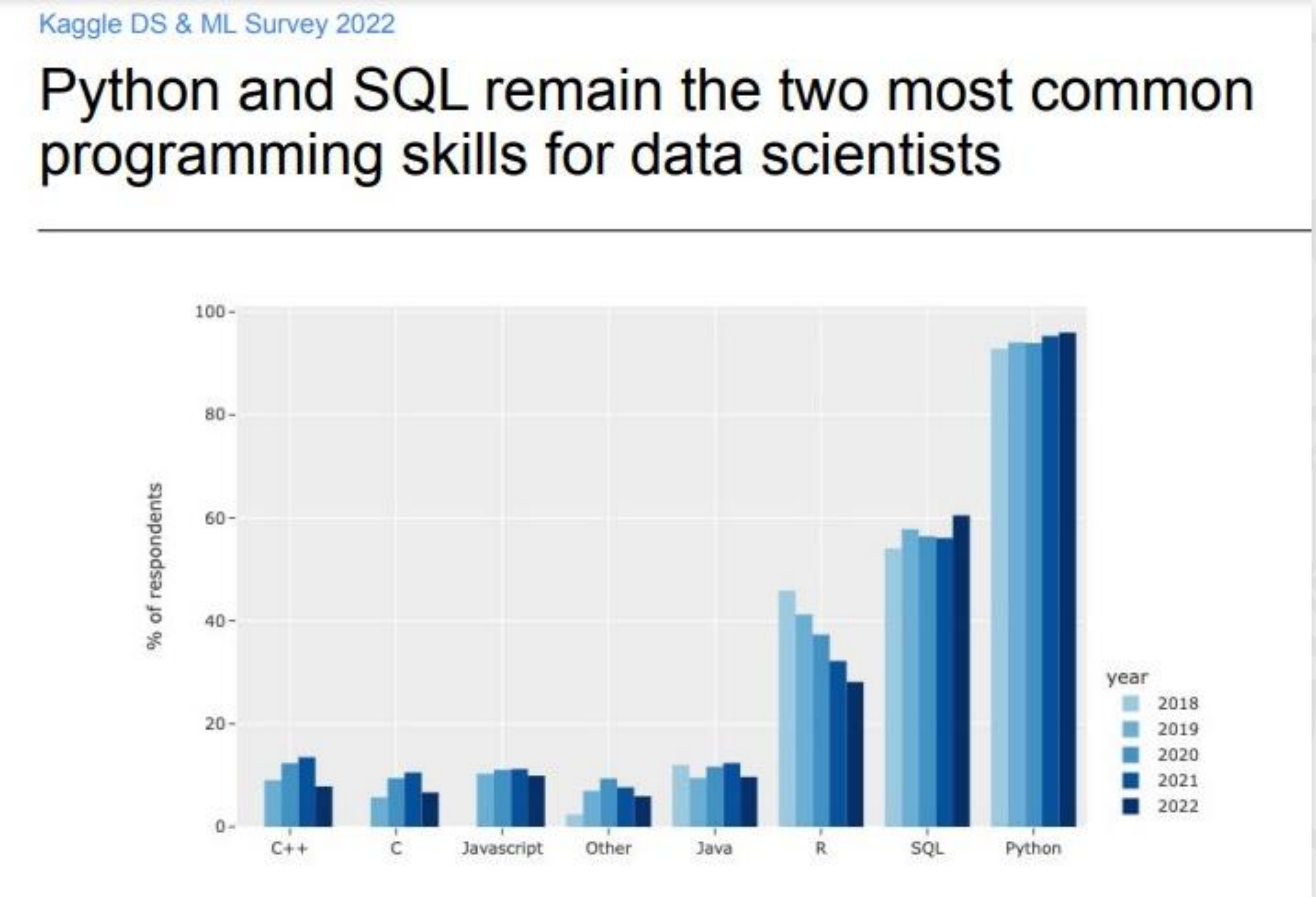
- Python is th
in Scientific

Matlab

Fortran

C++

More language?



Kaggle DS & ML Survey 2022

Python and SQL remain the two most common programming skills for data scientists

Data source: Kaggle DS & ML Survey
Plot credit: https://studyopedia.com/data-science/programming-languages-for-data-science/#google_vignette

Coding languages

Python

Matlab

Fortran

C++

More language?

- Python is the most widely used high-productivity language in Scientific Computing.
- Its very **simple syntax** and broad library support make it ideal for quickly building scalable applications.
- The language does not natively support the type of data structures and other features needed for fast computation, so *few hydrologic models are written in Python.*
- Python is great for data analysis so we will mainly use Python in this class!

Coding languages

Python

Matlab

Fortran

C++

More language?

- Matlab is one of the oldest high-productivity languages and has been the defacto standard for fast numerical prototyping before Python.
- It is still heavily used in many numerical applications, given its excellent toolbox and huge amount of legacy code that exists.

Coding languages

Python

Matlab

Fortran

C++

More language?

- Matlab is one of the oldest high-productivity languages and has been the defacto standard for fast numerical prototyping before Python.
- It is still heavily used in many numerical applications, given its excellent toolbox and huge amount of legacy code that exists.
- **Paywall!!!** While Matlab has quite favorable licenses for academic use, it is expensive for commercial use, and if possible Python as open-source alternative is preferable for new projects.

Coding languages

Python

Matlab

Fortran

C++

More language?

- Fortran is one of the dinosaurs of scientific computing. Fortran originated in the 1950s and its most recent incarnation is Fortran 2018.
- Fortran is still actively used for a lot of HPC code, especially when it comes to legacy applications.
- Many Global Climate Models and hydrologic models were written in Fortran!

Coding languages

Python

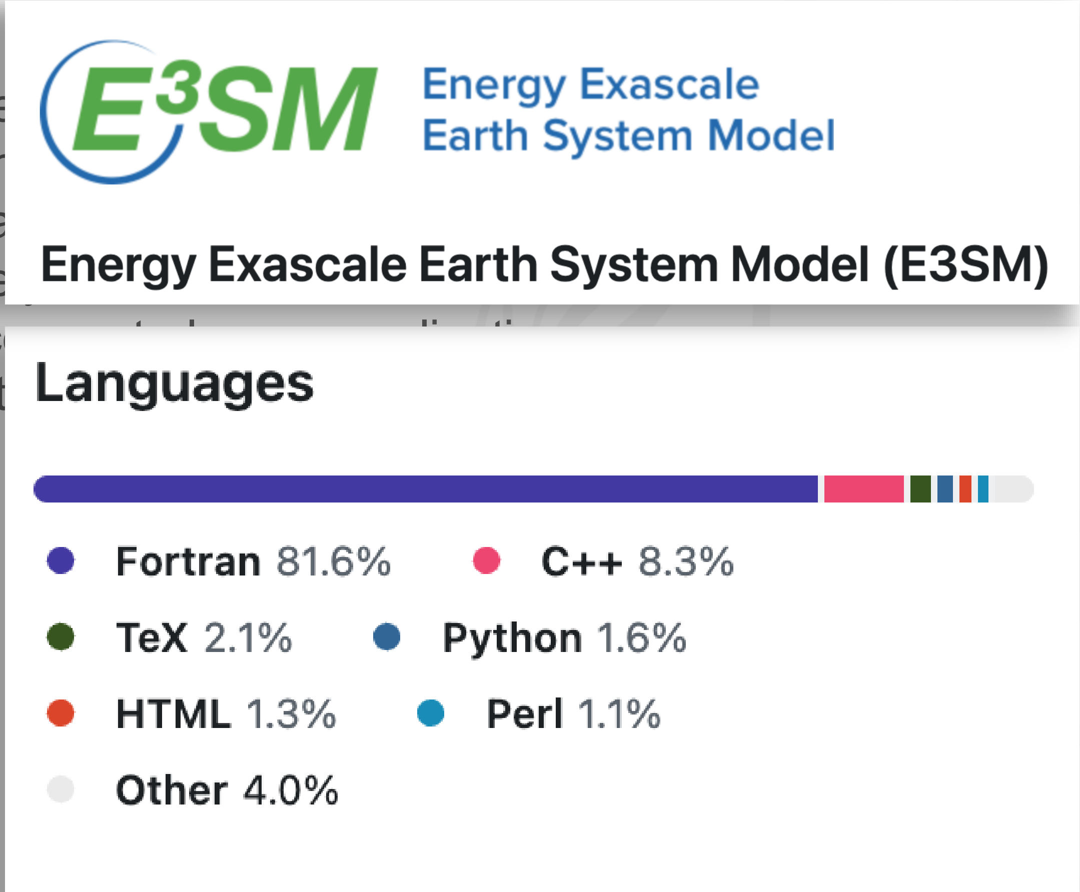
Matlab

Fortran

C++

More language?

- Fortran is one of the
Fortran originated ir
incarnation is Fortra
- Fortran is still active
especially when it c
- Many Global Climat
written in Fortran!



Energy Exascale Earth System Model

Energy Exascale Earth System Model (E3SM)

Languages

● **Fortran** 81.6%	● **C++** 8.3%
● **TeX** 2.1%	● **Python** 1.6%
● **HTML** 1.3%	● **Perl** 1.1%
● **Other** 4.0%	

Source: https://github.com/E3SM-Project/E3SM (accessed on Aug 26th 2024)

Coding languages

Python

Matlab

Fortran

C++

More language?

- C++ is the default language of Scientific Computing.
- It is mature, has a huge ecosystem and most modern heterogeneous compute environments (Cuda/Sycl, etc.) are developed for C++.
- Similar to Fortran, it requires compilation, and it is not as user-friendly as Python or Matlab, so it has a higher learning curves.

Examples

- Please write a function to add up two variables, **a** and **b,** **c**alculate the sum of ***a*** and ***b,*** and assign the value to the variable ***result***

How will you write this function in different languages?

Some other tools used to build this course

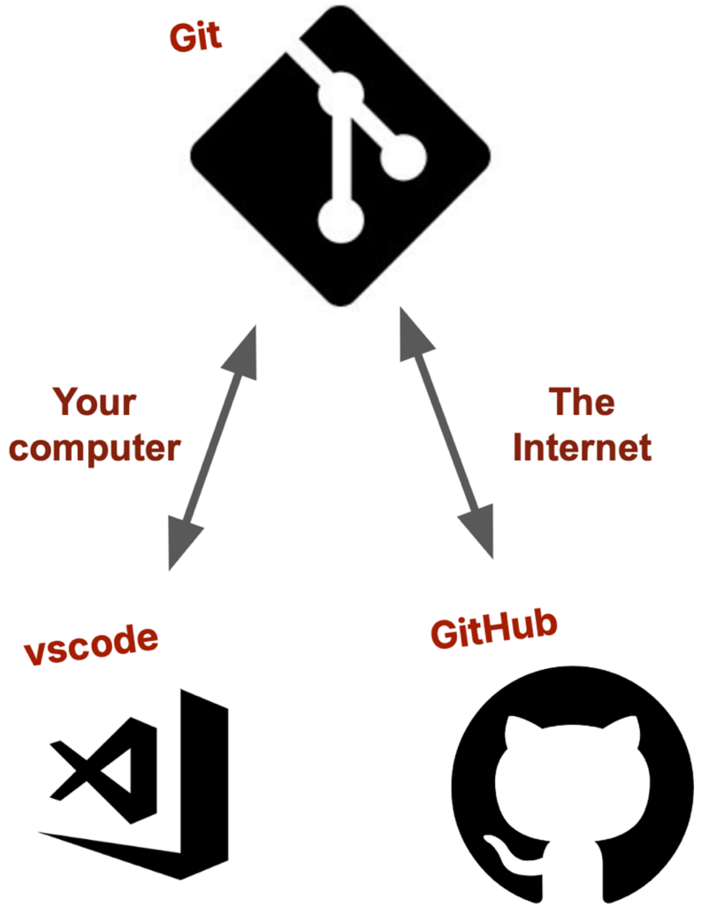
- We will be doing all of our work in **Python**
  - High-level, dynamic, and prominent programming language with massive community support
- We use **uv/pip** for environment management
  - Provides access to community-developed Python packages – more on this towards the end of the semester
- Under the hood, we use **Jupyter** as an interactive computing framework
  - Make it easy to run “snippets” of code and view their outputs. The standard framework for a lot of exploratory scientific computing
- **VSCode** will provide the user interface and code editing experience
  - A modern, extensible, and popular code editing application

GitHub

- GitHub is a web platform that enables software developers to manage and share their code
- Also has great integrations for running automation, building websites, and increasing AI integration
- We will use it for course centralization and homework management via GitHub Education.

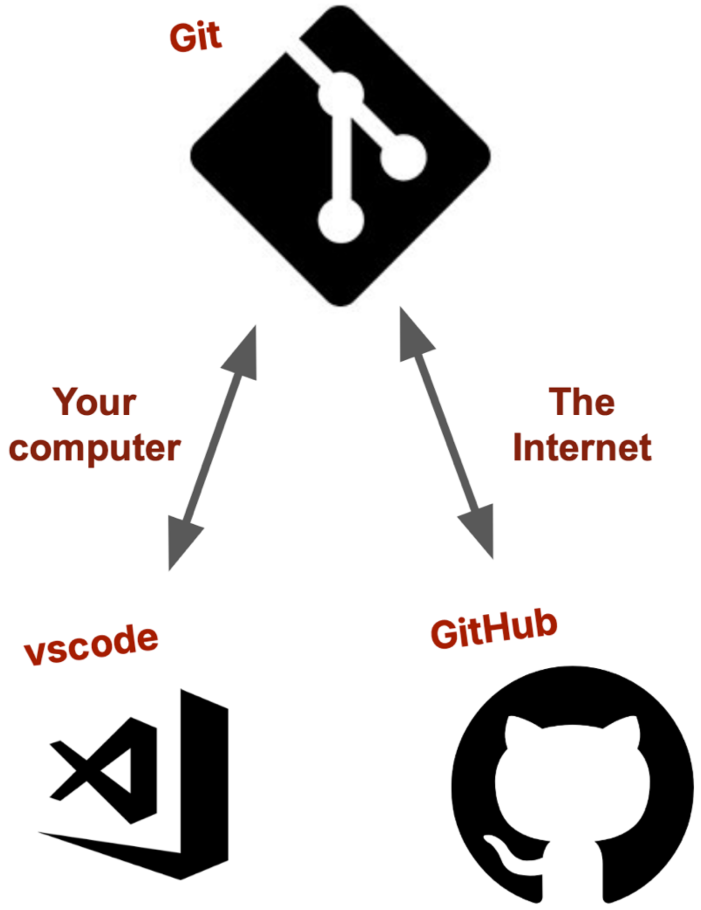
Let’s talk about git

- Git is a version control system. It allows teams to work collaboratively on the same pieces of code (like track changes for MS Word but much more sophisticated)



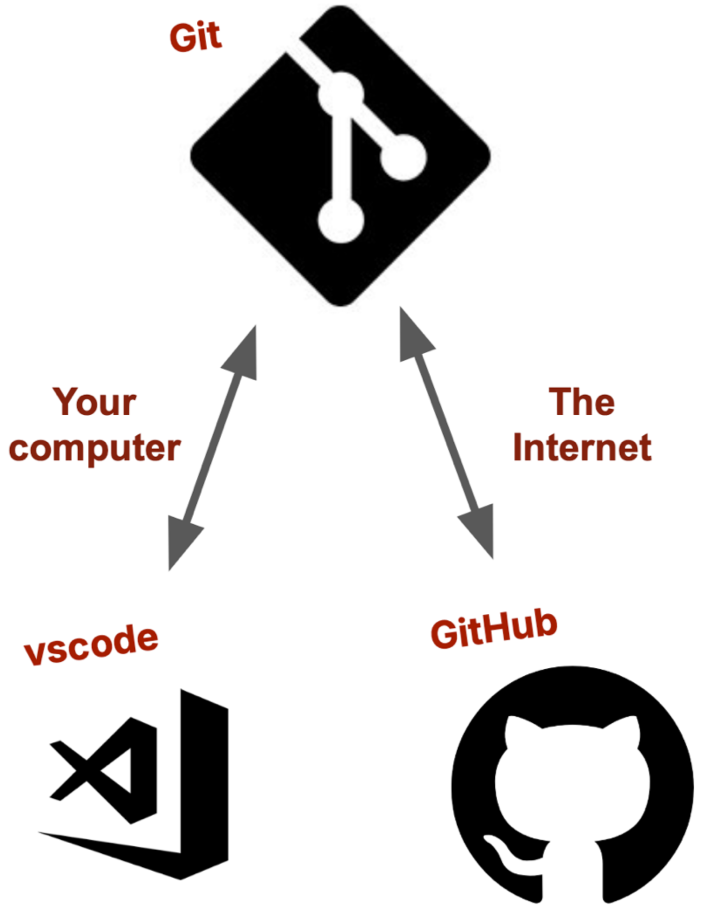
Let’s talk about git

- Git is a version control system. It allows teams to work collaboratively on the same pieces of code (like track changes for MS Word but much more sophisticated)
- GitHub is a website that is essentially a social platform for code, built around git as a version control system



Let’s talk about git

- Git is a version control system. It allows teams to work collaboratively on the same pieces of code (like track changes for MS Word but much more sophisticated)
- GitHub is a website that is essentially a social platform for code, built around git as a version control system
- You will mostly interact with git inside VSCode and use it to communicate with GitHub.



The first thing to know about git: Jargons

Jargons	Description
repository	The overall “container” for your work (synonym: repo) [noun]
clone	Make a copy of the repository on your local machine [verb]
commit	A snapshot in time of your work, or to save changes [noun/verb]
checkout	How to get to a particular snapshot [verb]
branch	A label to a commit, usually denoting a separate stream of work [noun]
merge	Combine changes from different branches [verb]
remote	A computer somewhere else with a repo on it (GitHub) [noun]
local	The computer you are working on (laptop) [noun]
fetch	Download information about history from a remote [verb]
pull	Do a fetch, but also update the code status with a checkout [verb]
push	Sync changes on your local to a remote [verb]
conflict	Occurs when two commits contradict each other [noun?]

Basic concepts in git: Nodes and trees

All repos start from an initial commit

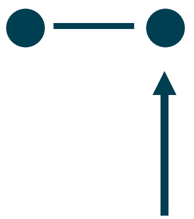
You are here

```
# Your codefile
print('hello world')
```

Basic concepts in git: Nodes and trees

All repos start from an initial commit

As you do work, you can make changes and add commits



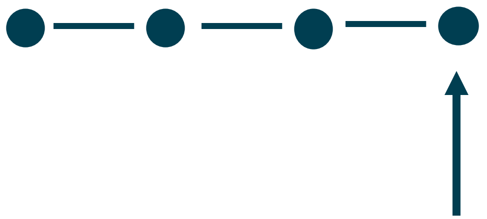
You are here

```
# Your codefile
name = ‘OS-Hydro’
print(‘hello world’)
```

Basic concepts in git: Nodes and trees

All repos start from an initial commit

As you do work, you can make changes and add commits



You are here

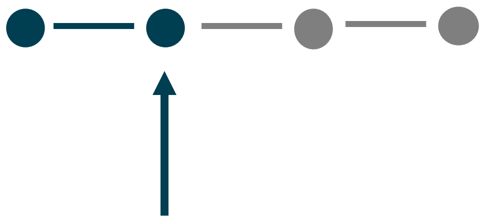
```
# Your codefile
import numpy as np
name = ‘OS-Hydro’
print(f‘hello {name}’)

x = np.arange(50)
y = np.sin(x / np.pi)

# more fun stuff here
```

Basic concepts in git: Nodes and trees

All repos start from an initial commit

As you do work, you can make changes and add commits

You can also move back and forward

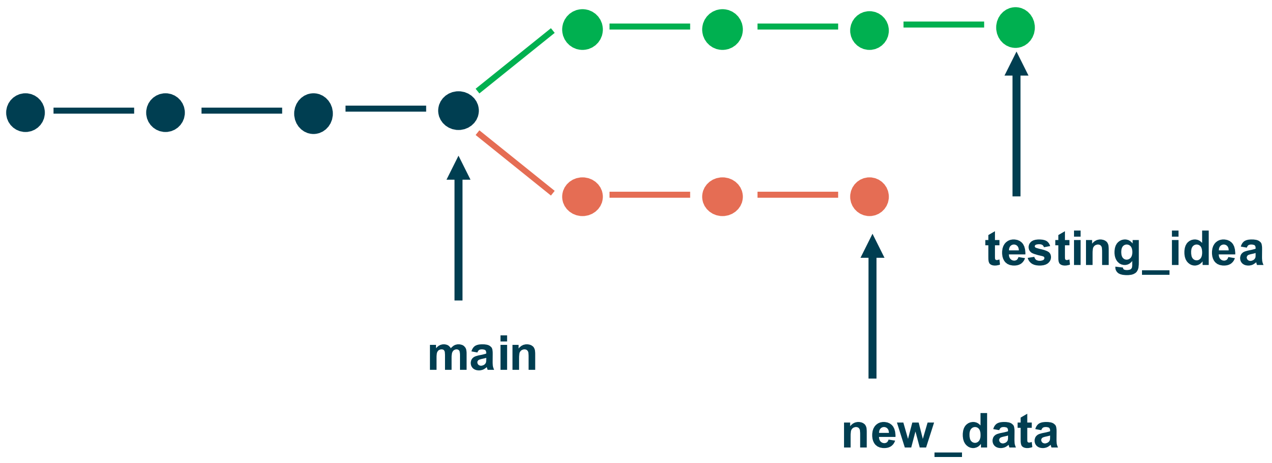


You are here

```
# Your codefile
name = ‘OS-Hydro’
print(‘hello world’)
```

Basic concepts in git: Nodes and trees

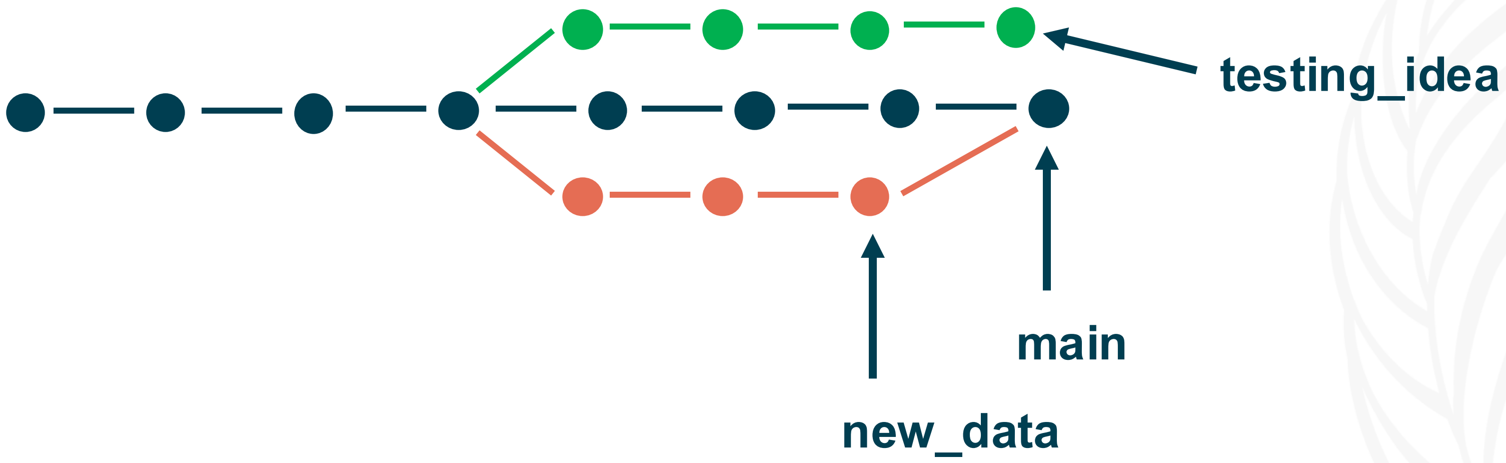
You can also start splits call branches



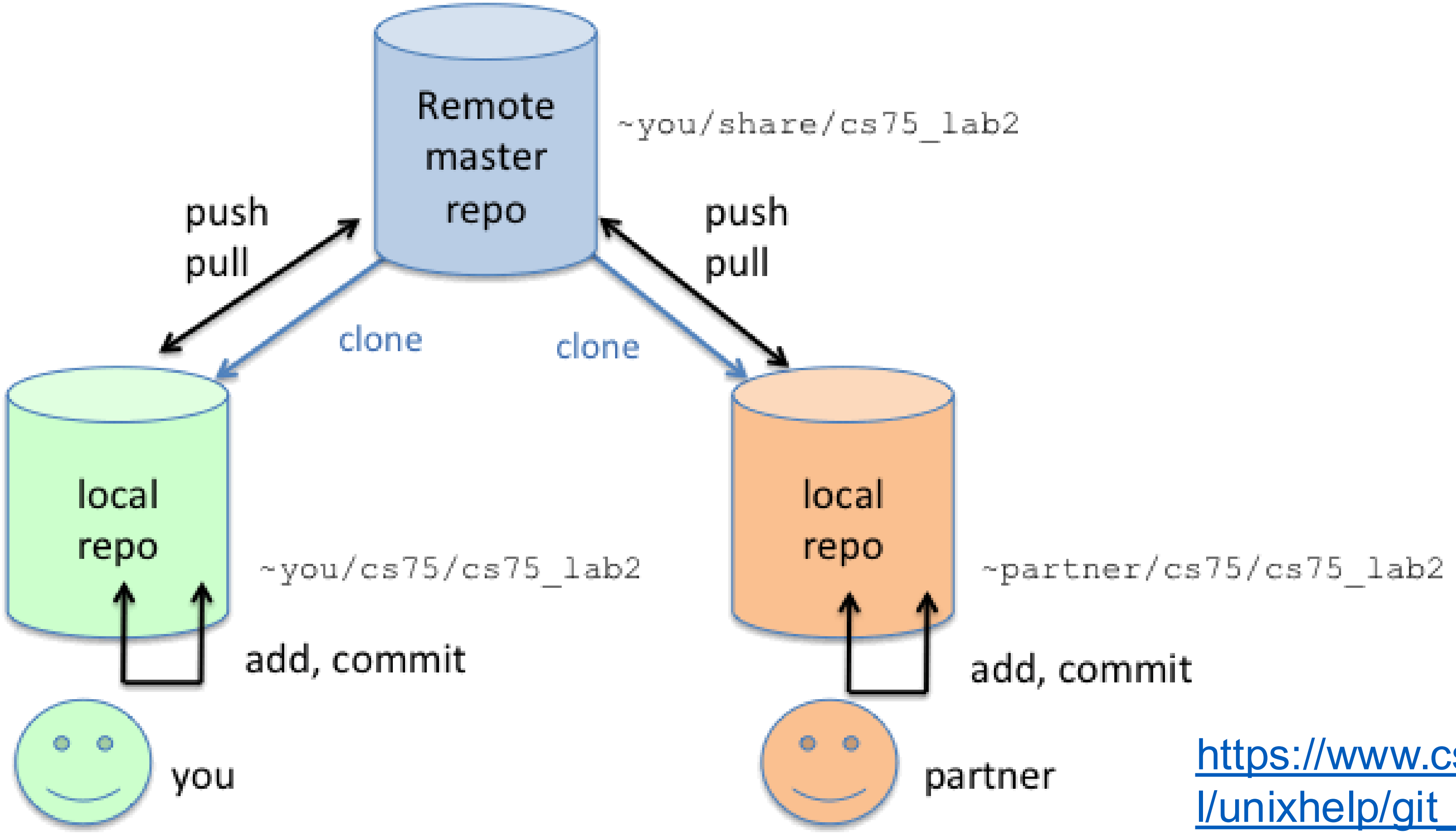
Basic concepts in git: Nodes and trees

You can also start splits call branches

And when you finish an idea you can merge them together



Basic concepts in git: Local and remote

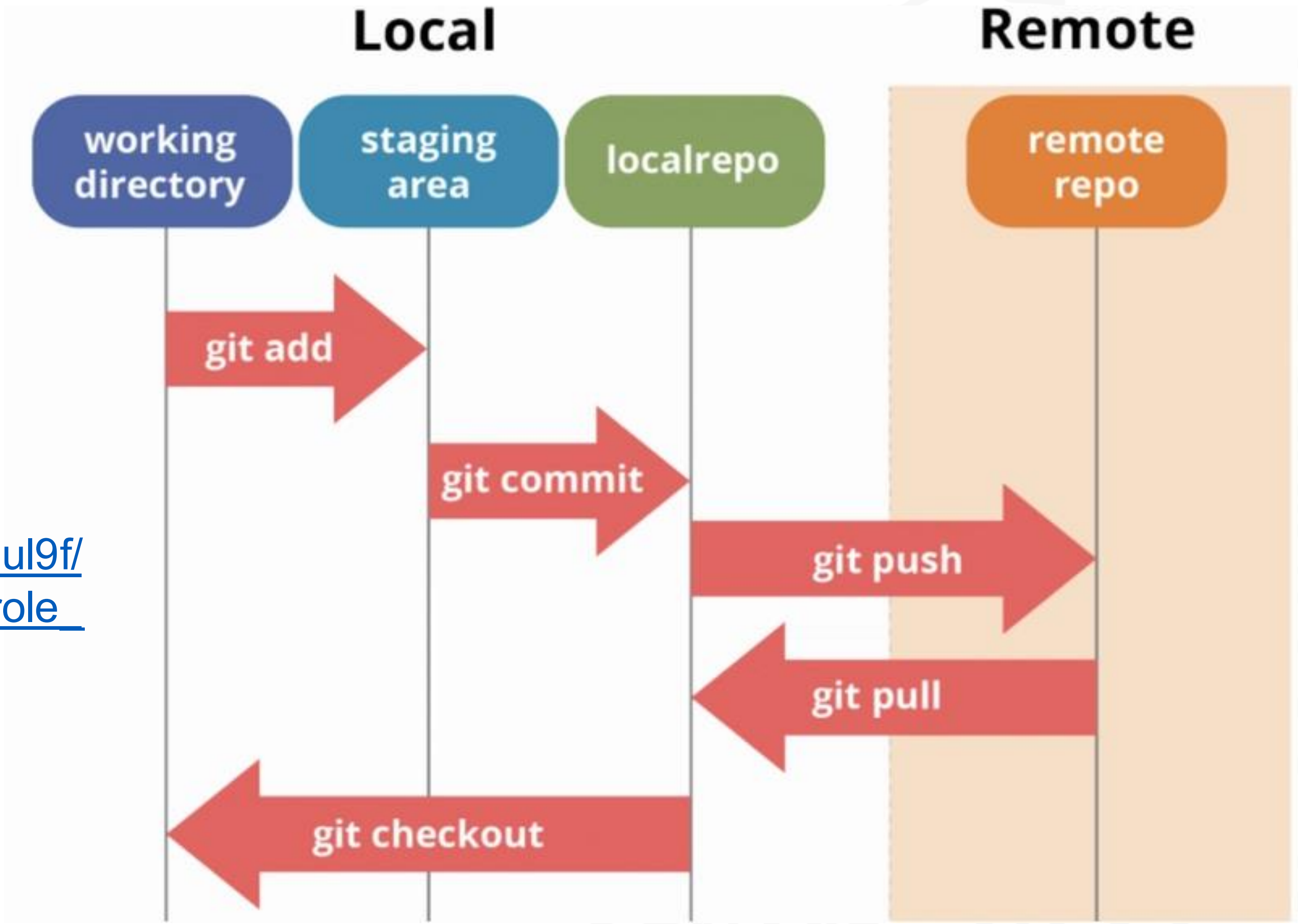


https://www.cs.swarthmore.edu/~newhall/unixhelp/git_create.php

GitHub workflow

Basics:
https://dev.to/mollynem/git-github--workflow-fundamentals-5496

Advanced:
https://www.reddit.com/r/git/comments/99ul9f/git_workflow_diagram_showcasing_the_role_of/



To get familiar with the setup, we will practice GitHub in today’s lab