Getting Started with Git

By

Vijaya Nandini M

Getting Started with Git

- **Welcome!**
- This is where things really start to take-off since we're going to learn a lot of commands that you will use **all** the time with Git and GitHub.

Getting Started with Git

- **Review:**
    - So far we've learned about version control, setting up Git and GitHub, and how to create Code Repositories.
    - Now it's time to learn how to add code to these repositories, make changes or updates, and push or pull changes from repos to and from local machines.

Getting Started with Git

- **Review:**
    - We're also going to be more precise with terminology than we have been.
    - We've been using the term code **repository** as a catch-all term for where we are developing our code, but really we want to break this down into our **working directory**, **staging area**, and **repository**.

Getting Started with Git

- **Overview:**
  - Learn the logic behind Git Usage and general Workflow.
  - Explore commands that allow us to use this workflow.

Basic Git Usage

Getting Started with Git

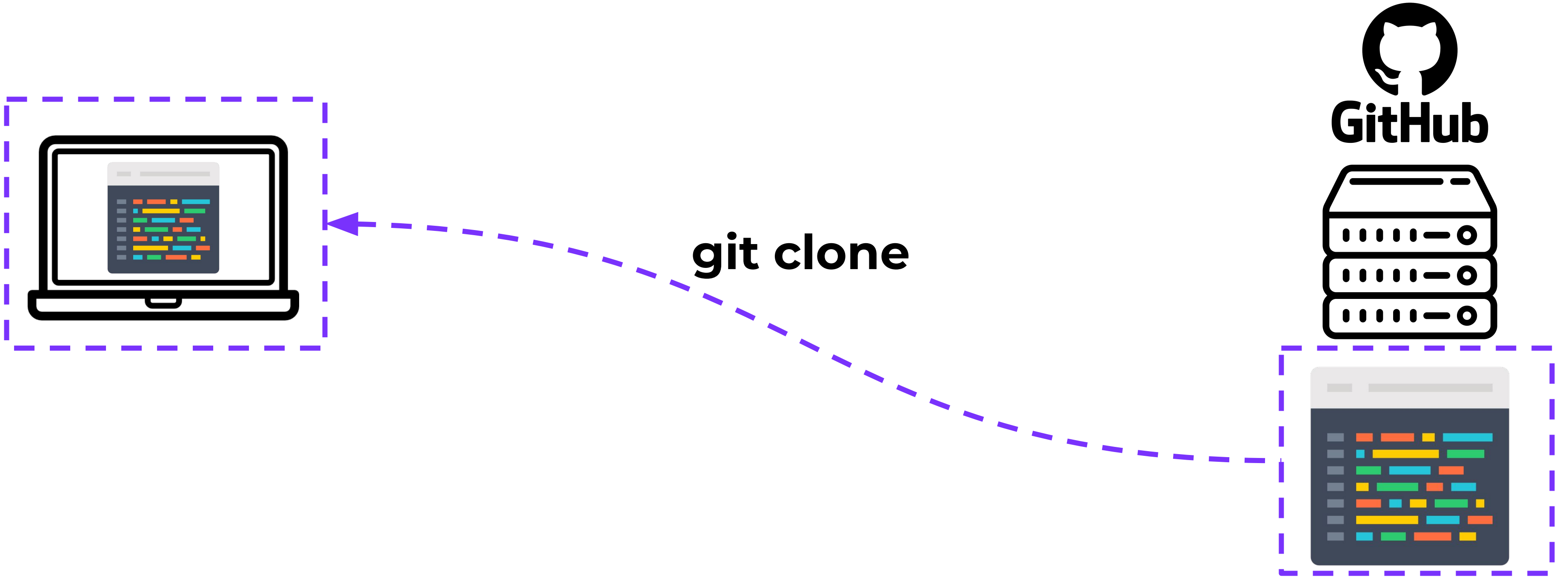
- **Basic Git Usage**
  - Let’s cover the basic cycle of a workflow of using Git and GitHub.
  - This particular basic example will assume just a solo developer and everything working on the same branch.

Getting Started with Git

Getting Started with Git

Create Repo on GitHub

Getting Started with Git



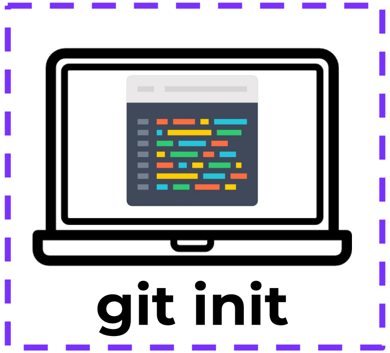
Create Repo on GitHub

Getting Started with Git

Getting Started with Git

git init
GitHub

Getting Started with Git

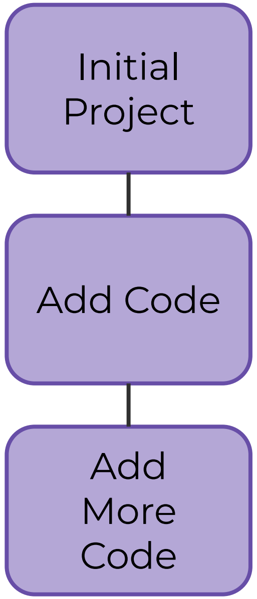


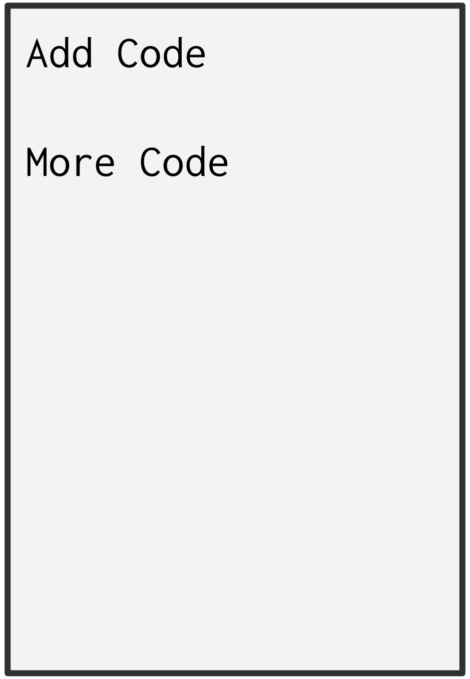
Getting Started with Git

What we need to learn today:

- Git Workflow
- How to tell Git about changes to our code
- How to push changes to GitHub
- How to pull changes from GitHub

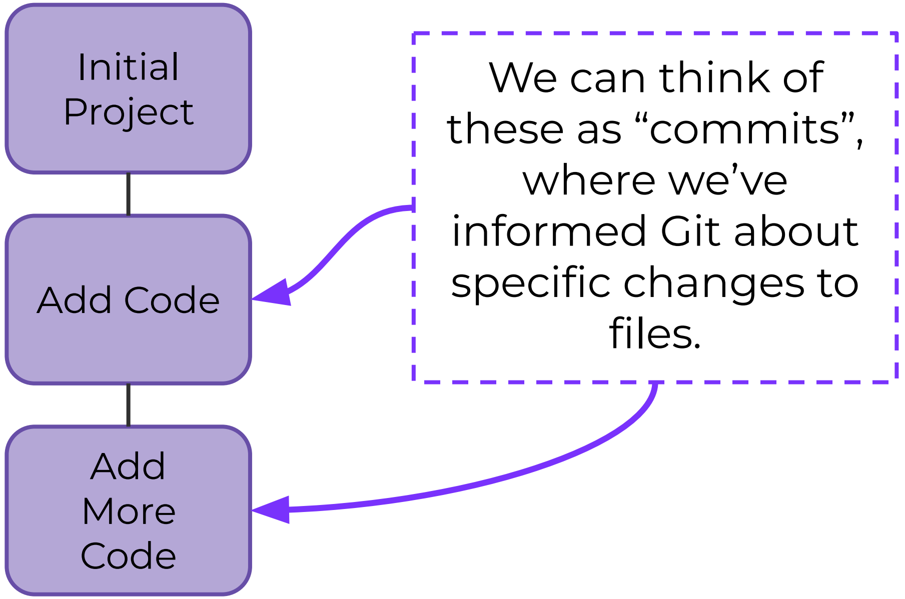
Getting Started with Git





```
Add Code

More Code
```

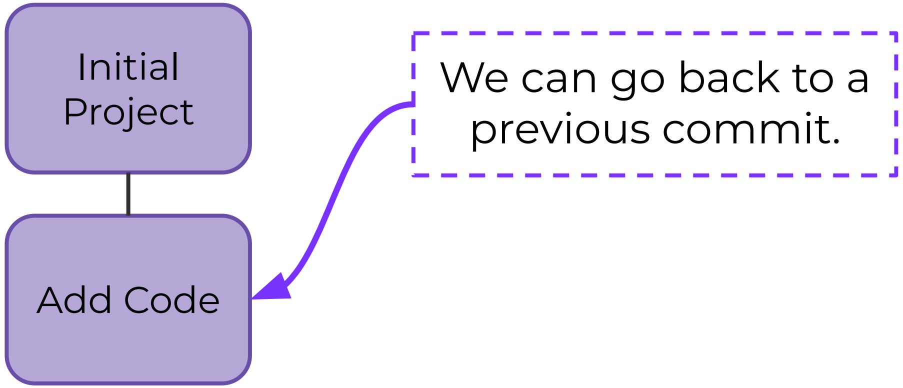
Getting Started with Git



We can think of these as “commits”, where we’ve informed Git about specific changes to files.

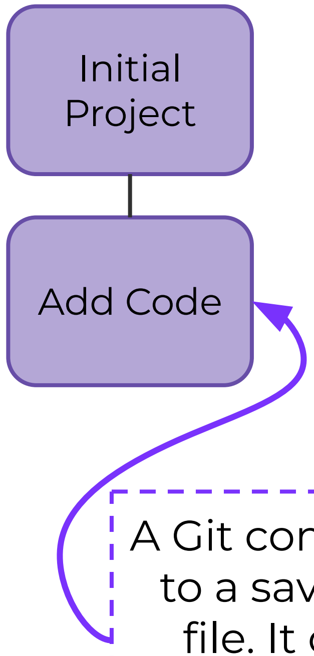
```
Add Code

More Code
```

Getting Started with Git

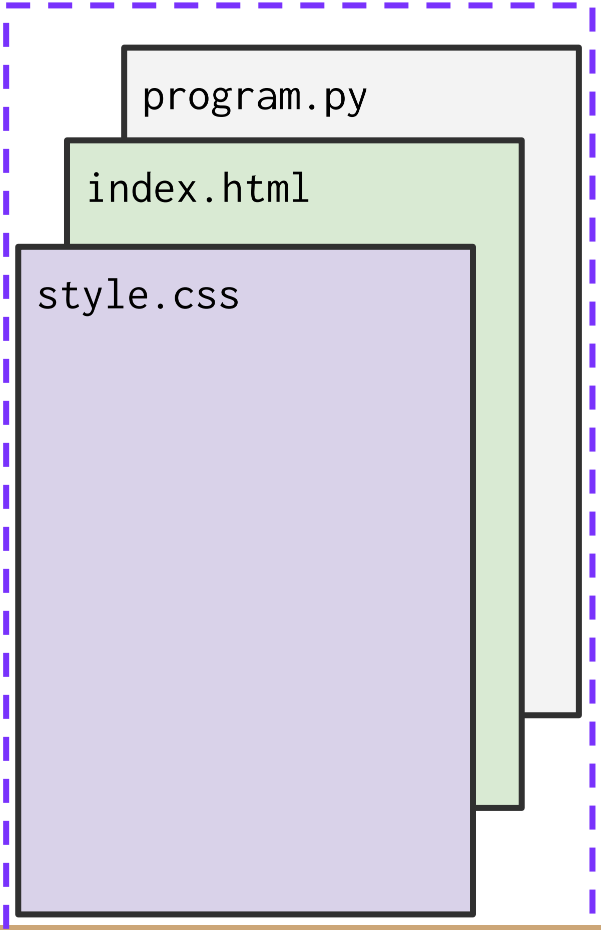


```
Add Code
```

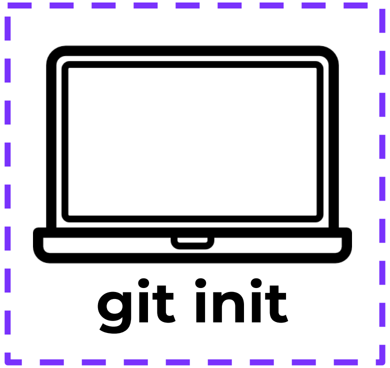
Getting Started with Git



A Git commit doesn't just pertain to a saving changes in a single file. It can constitute specific changes across an entire **working directory**.

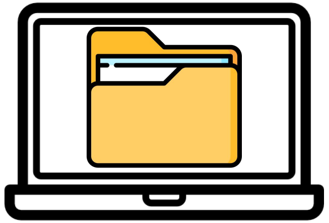


Getting Started with Git



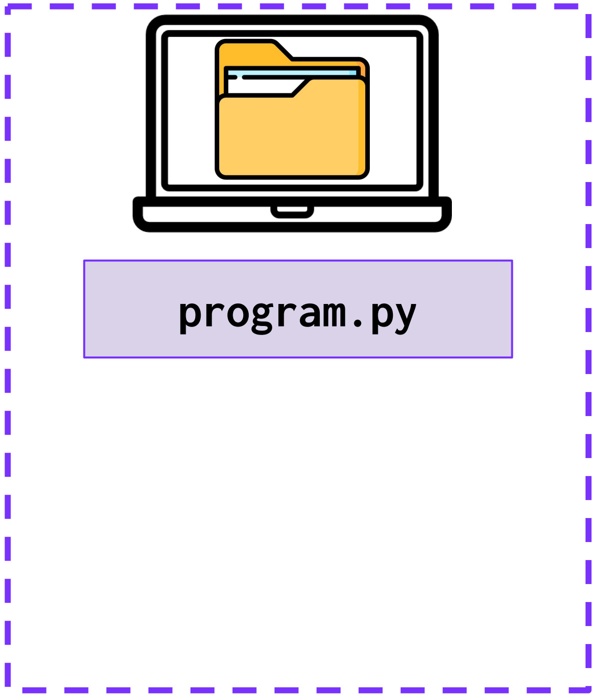
Getting Started with Git

Working Directory



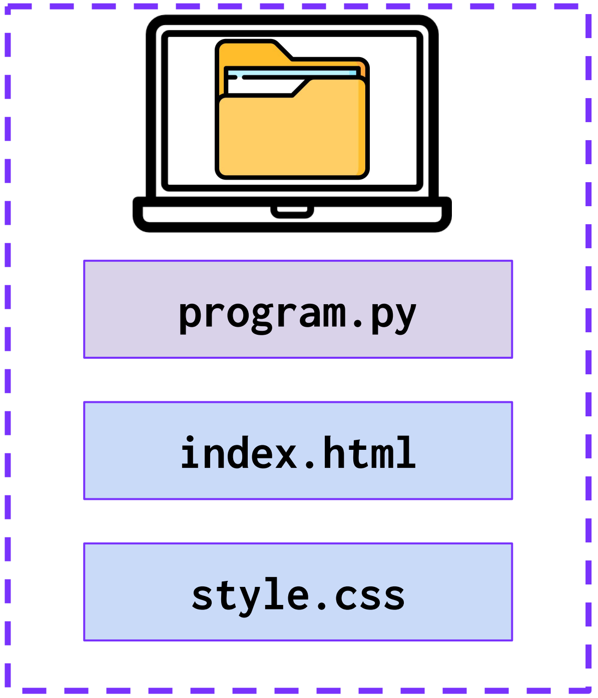
Getting Started with Git

Working Directory



Getting Started with Git

Working Directory



Getting Started with Git

Working Directory

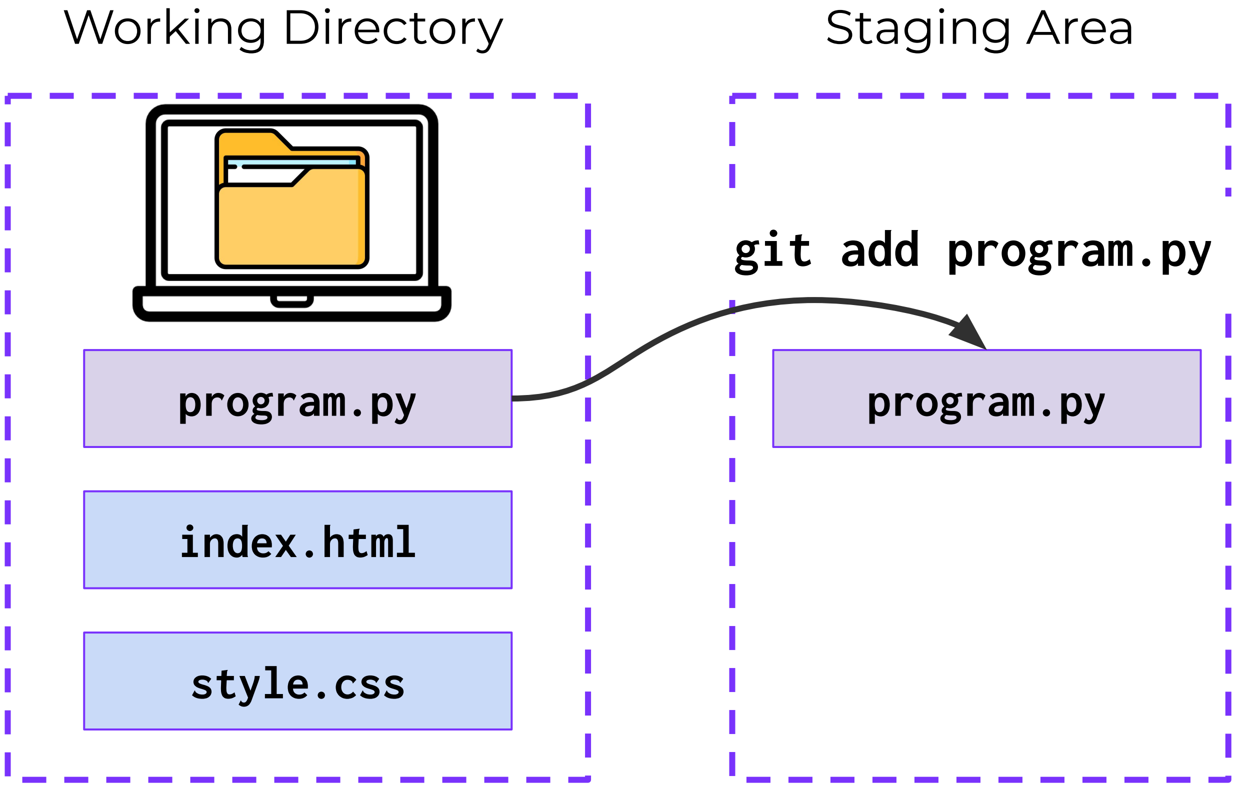
`program.py`

`index.html`

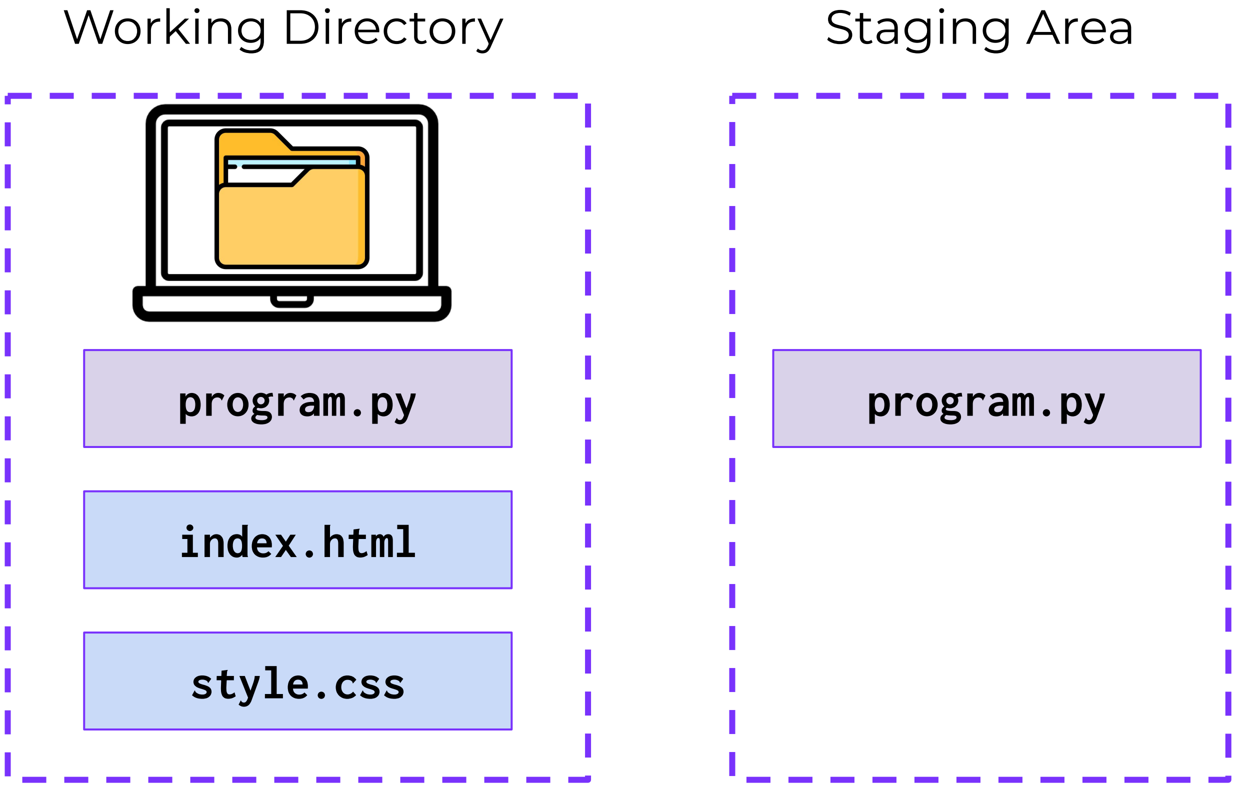
`style.css`

Staging Area

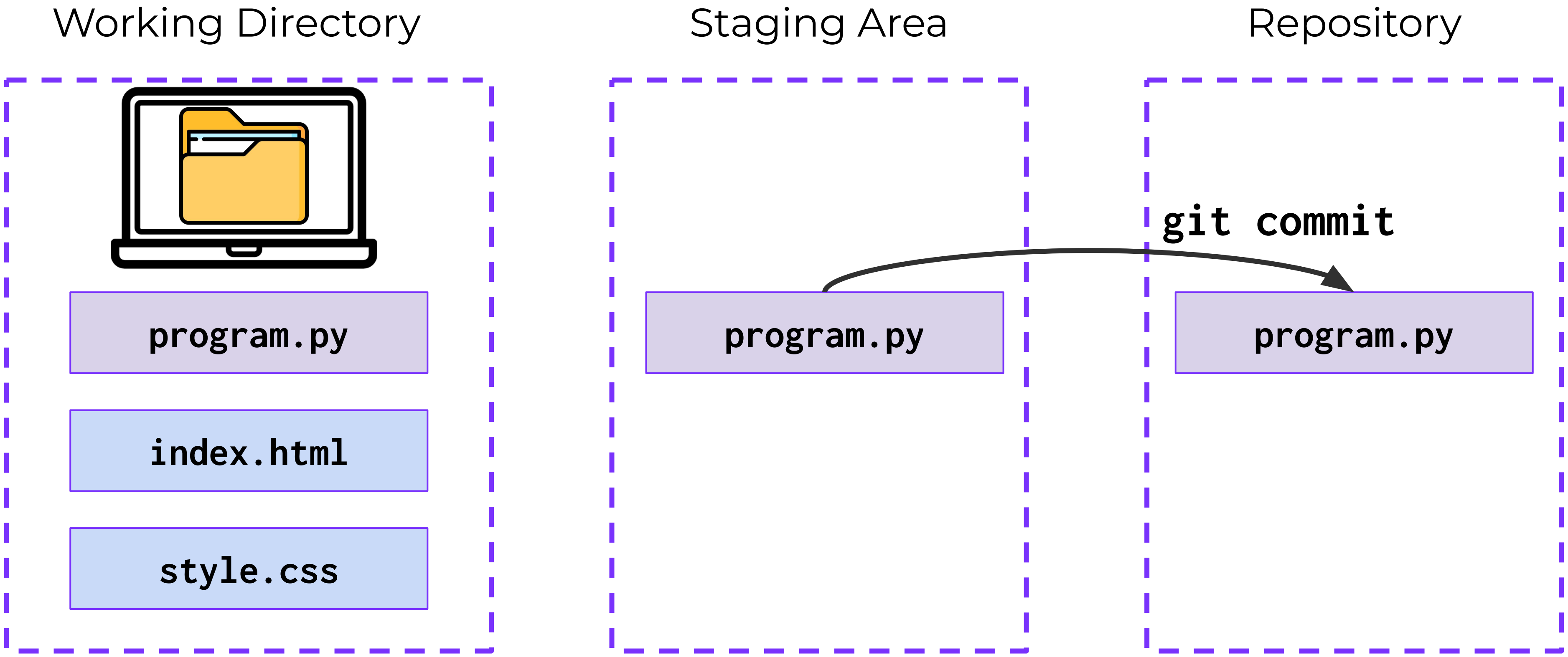
Getting Started with Git



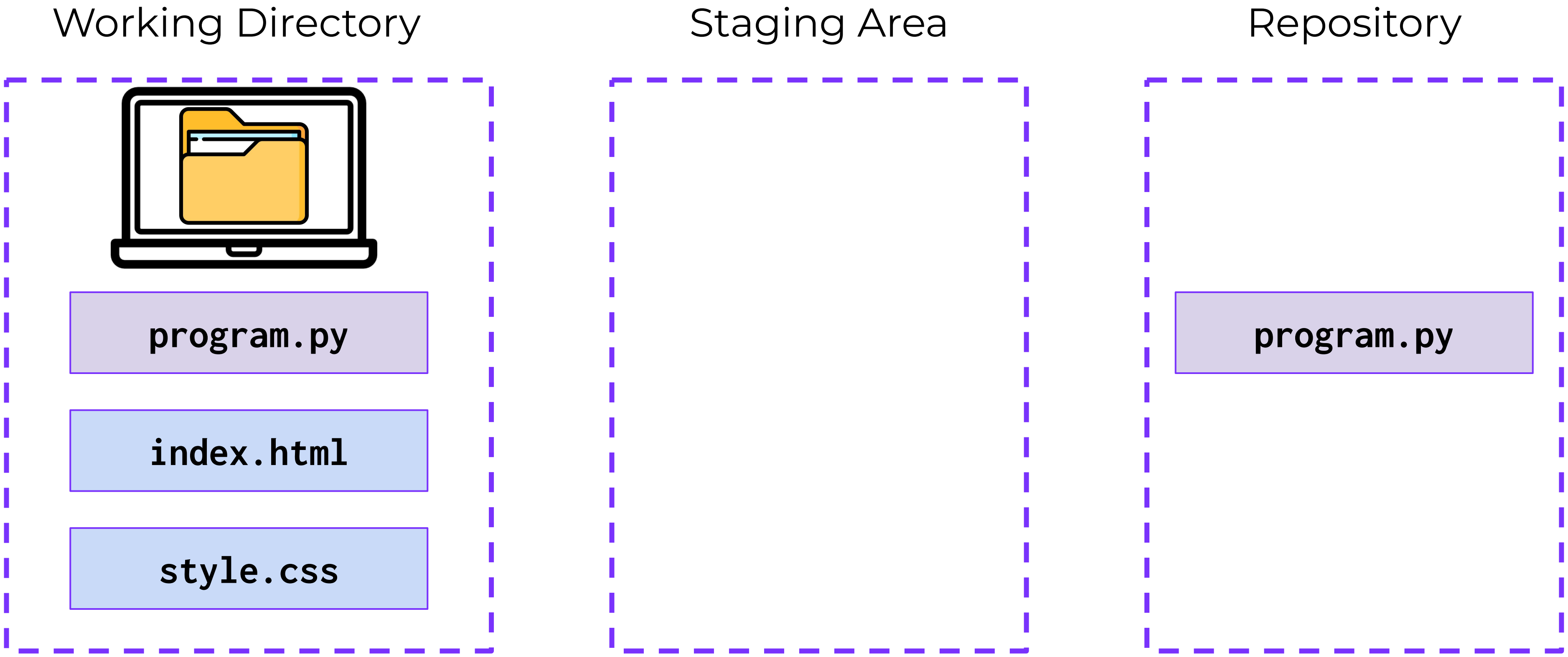
Getting Started with Git



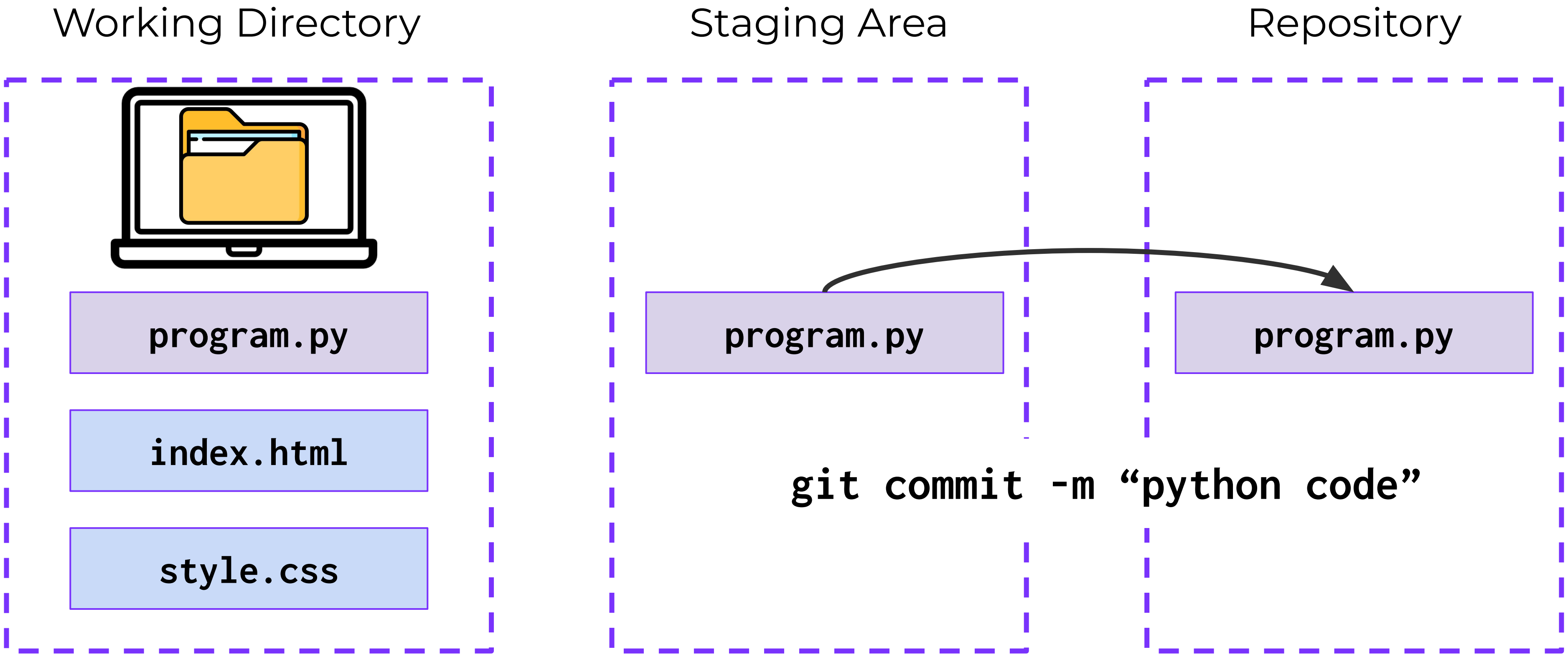
Getting Started with Git



Getting Started with Git

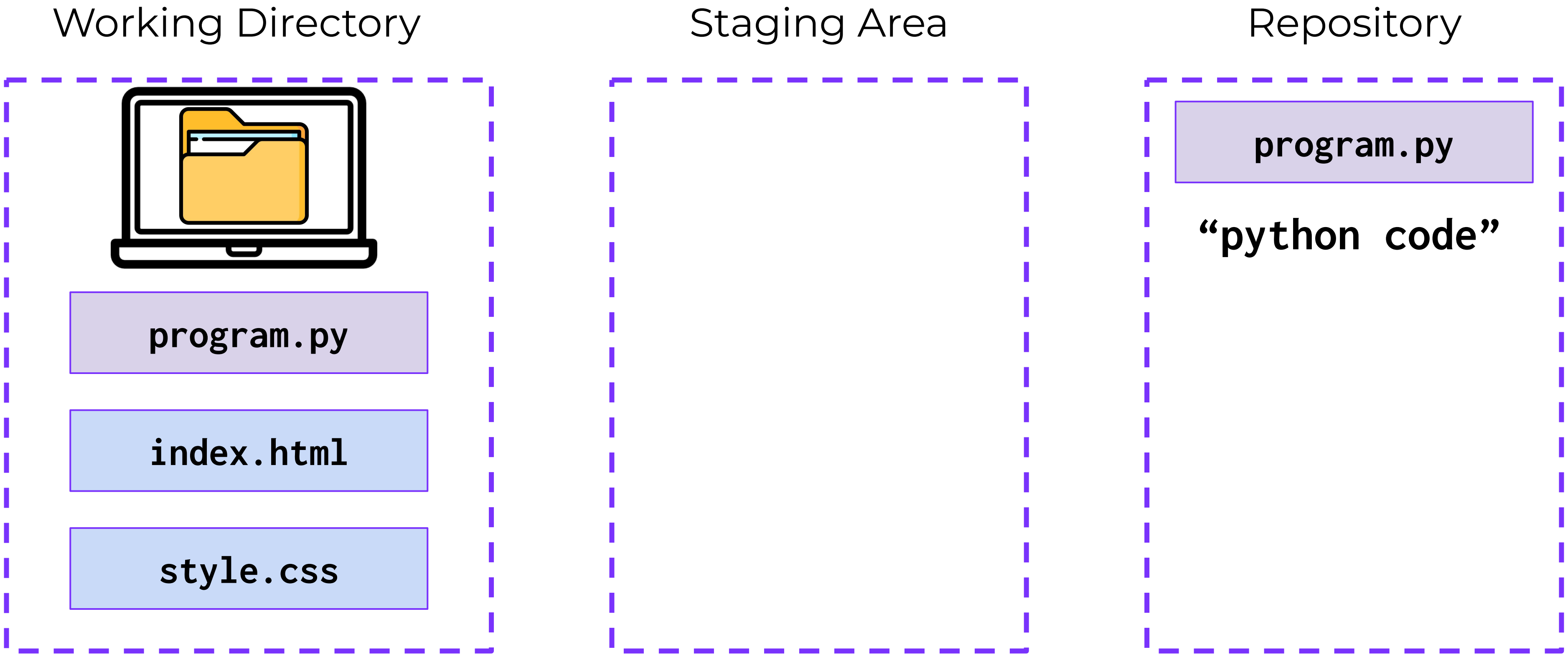


Getting Started with Git

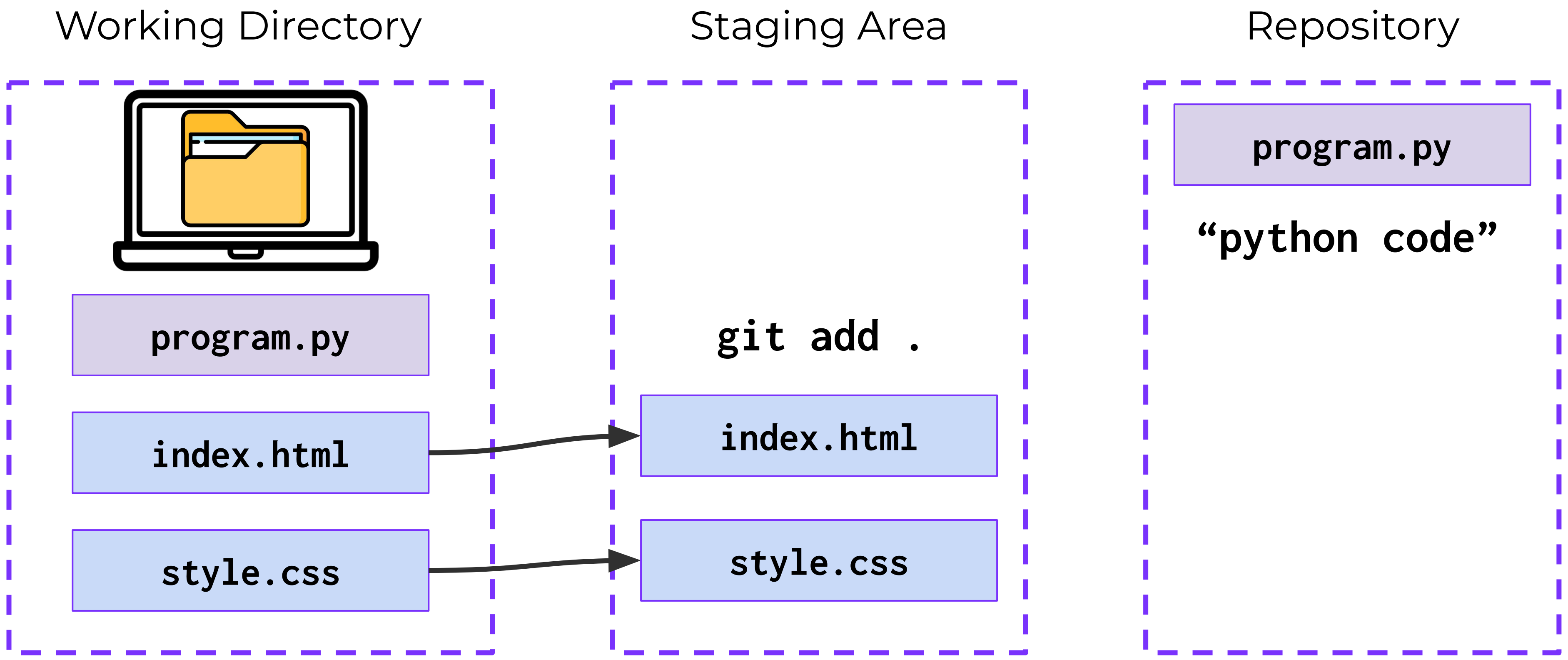


```
git commit -m "python code"
```

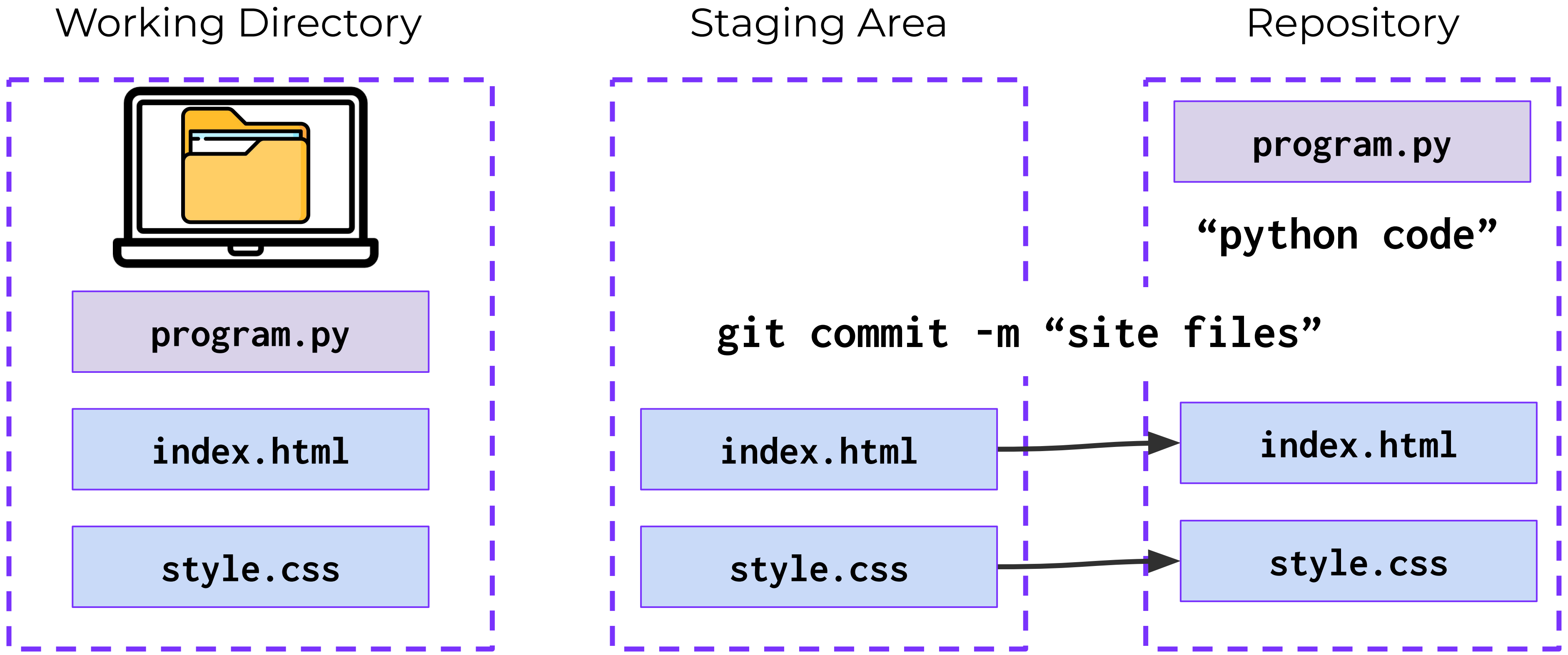
Getting Started with Git



Getting Started with Git

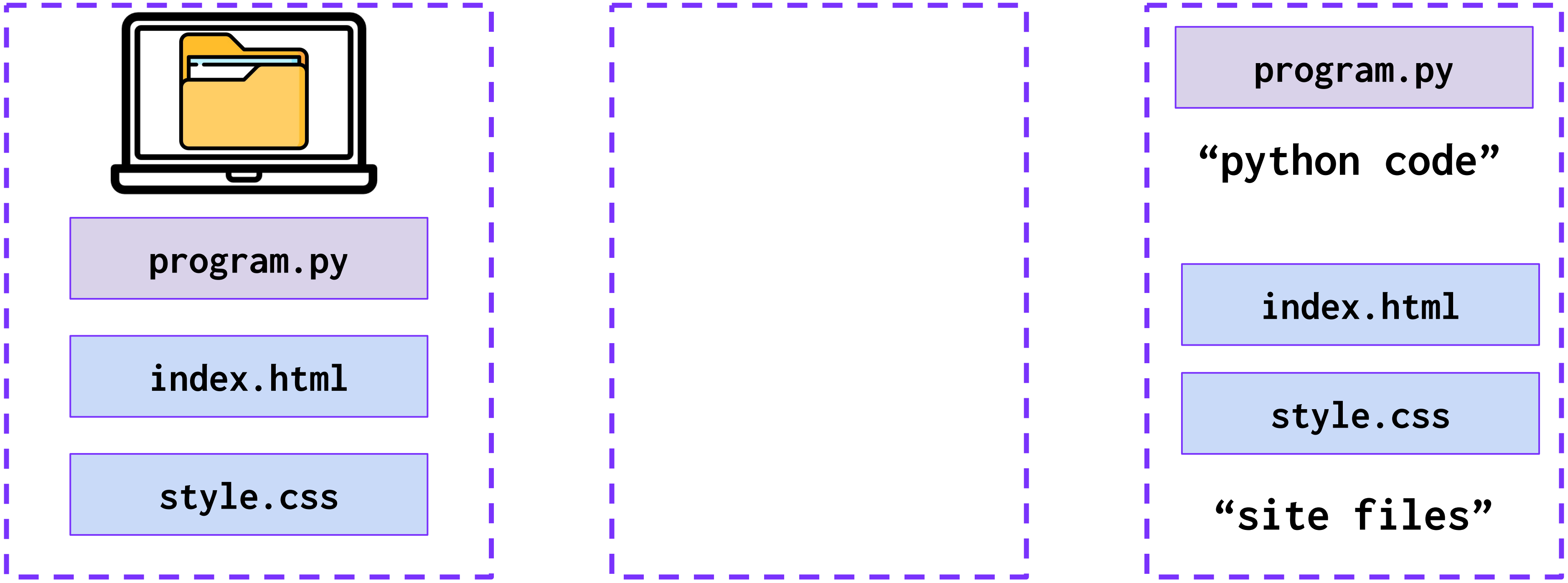


Getting Started with Git

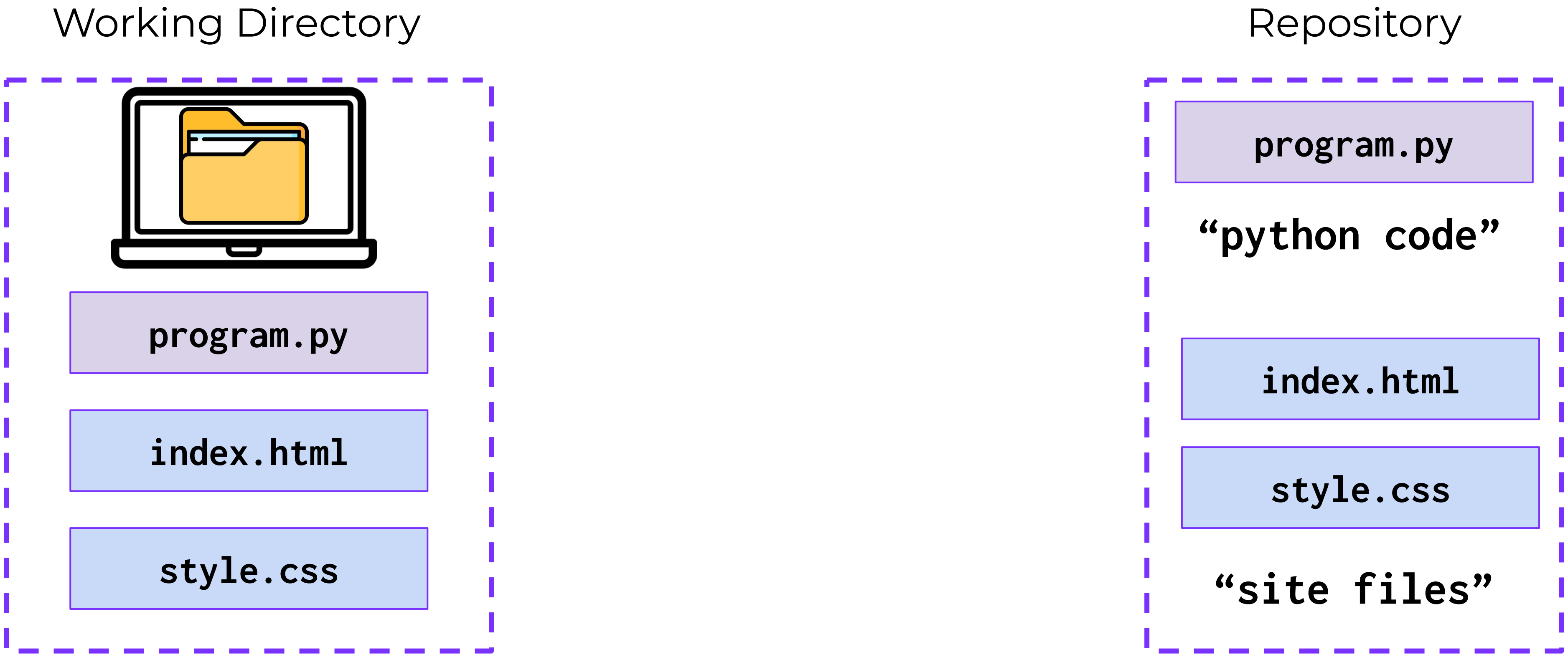


```
git commit -m “site files”
```

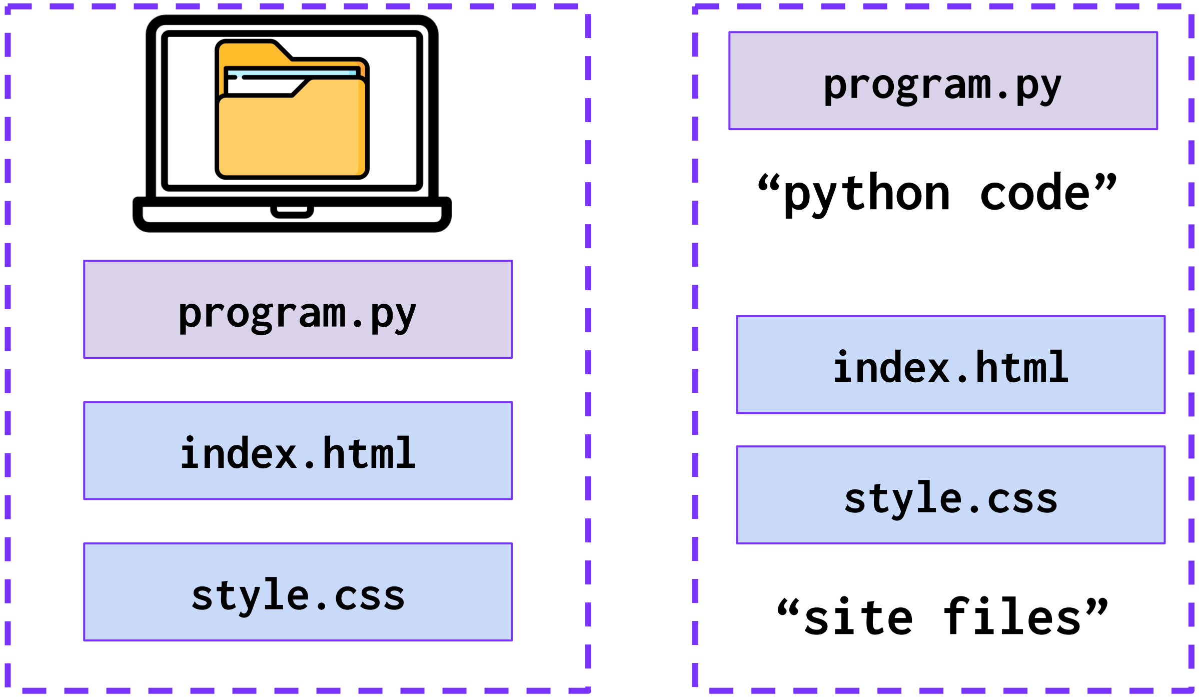
Getting Started with Git



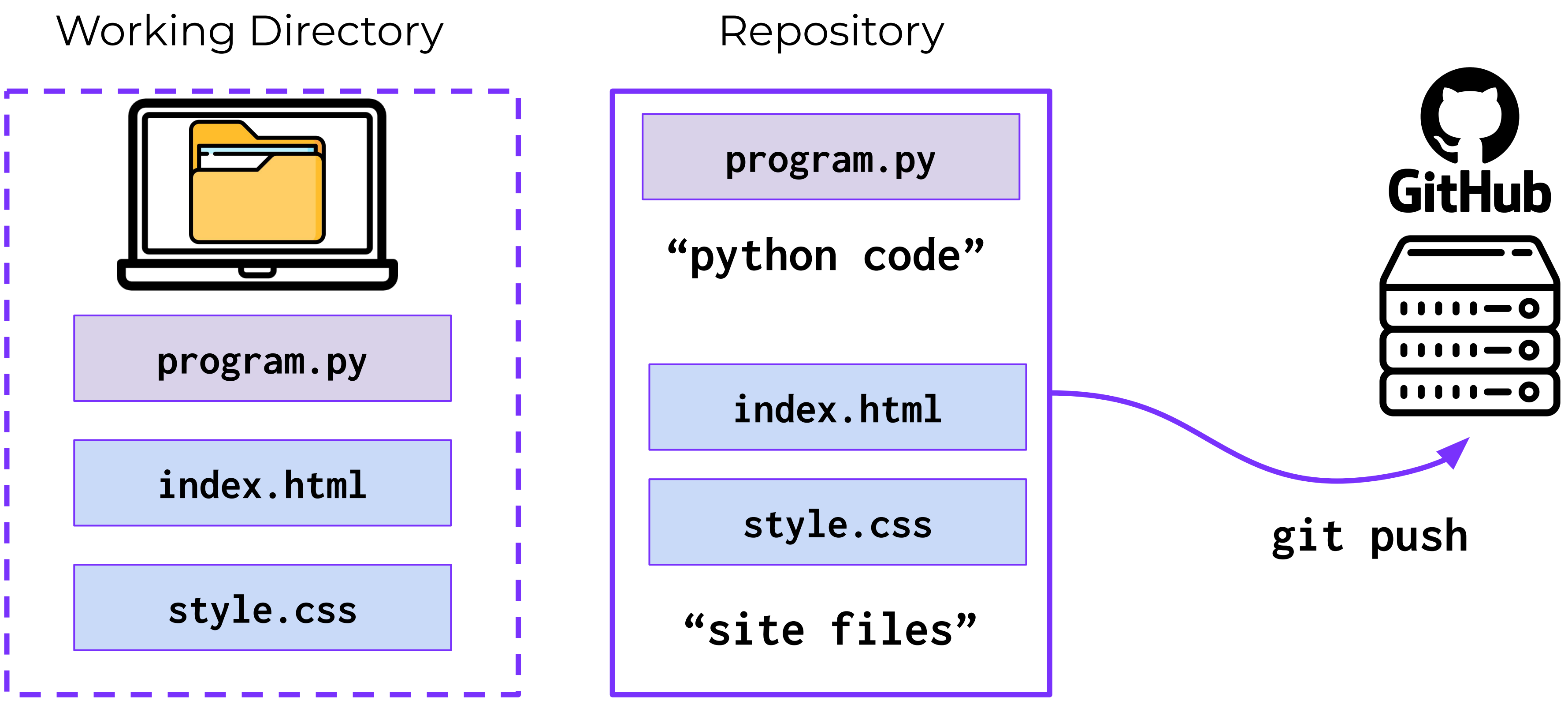
Getting Started with Git



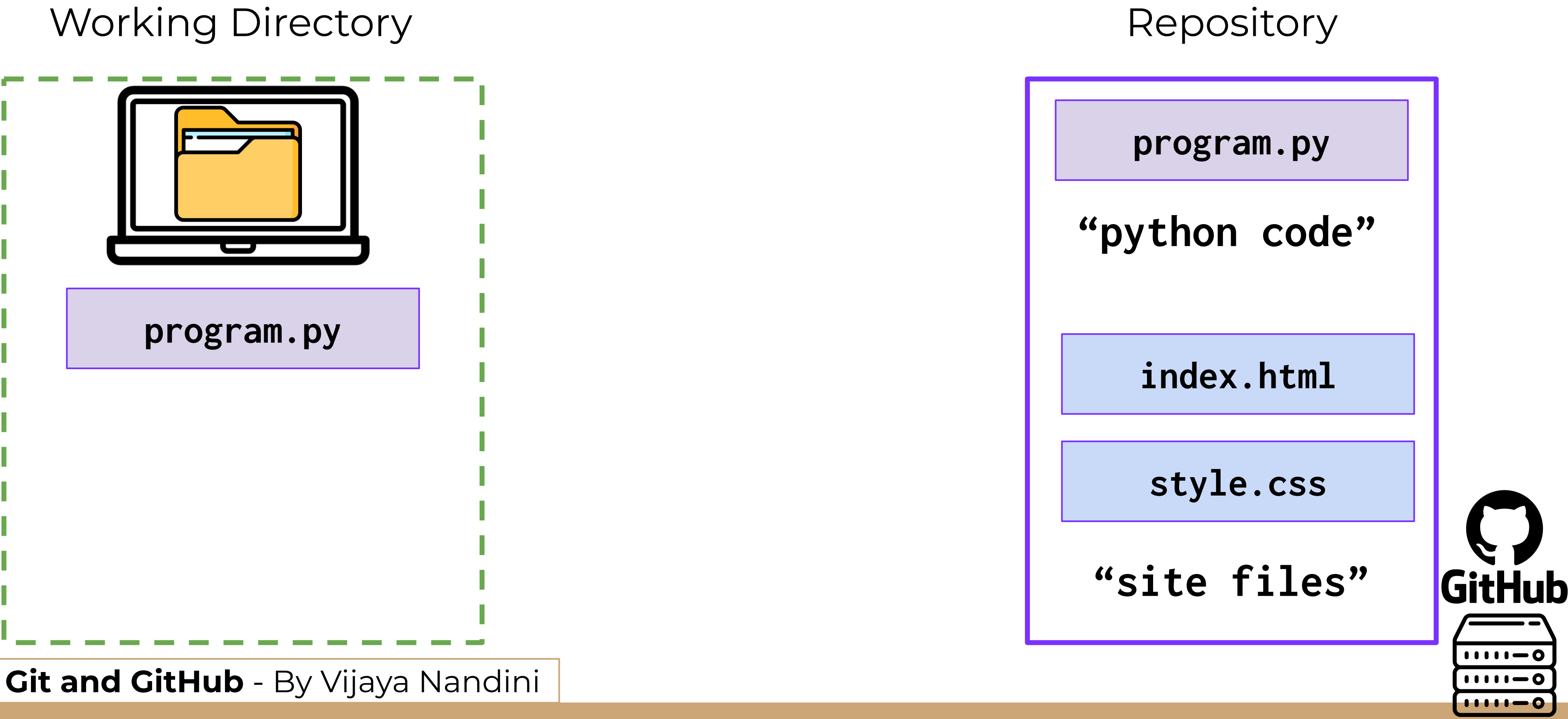
Getting Started with Git



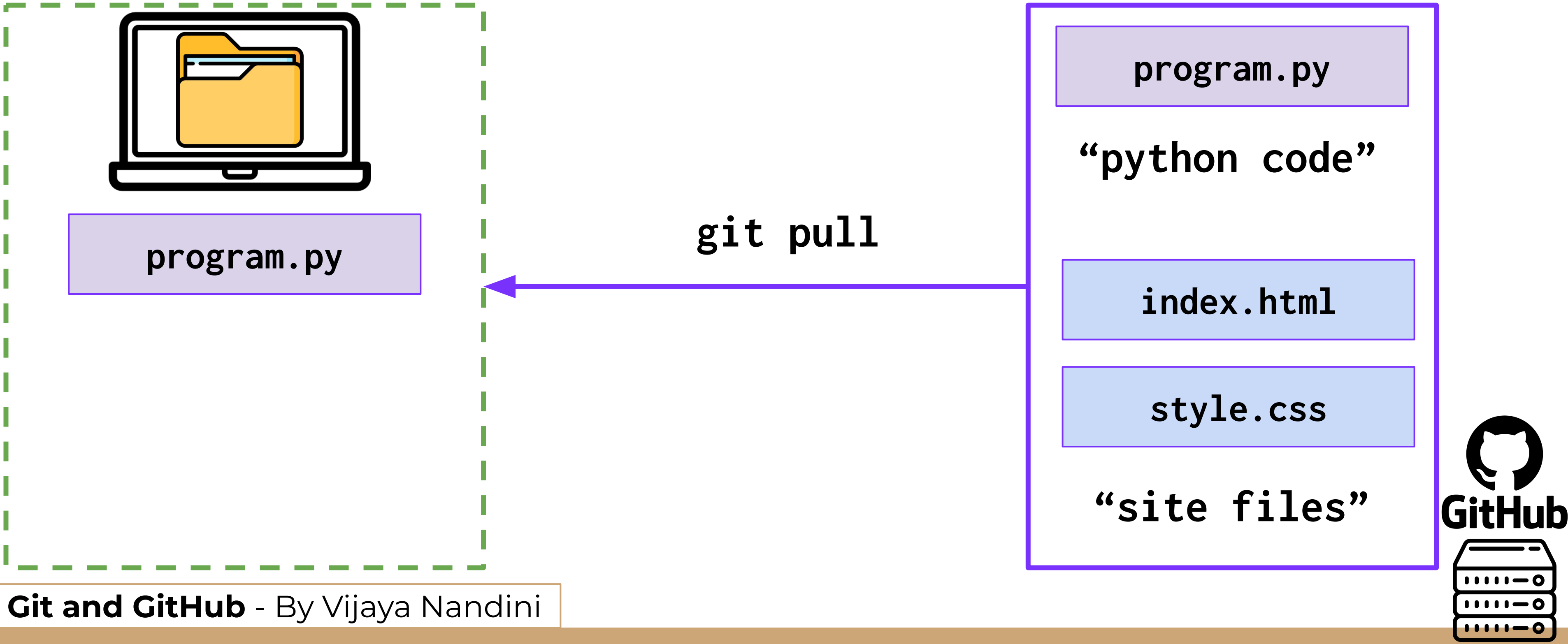
Getting Started with Git



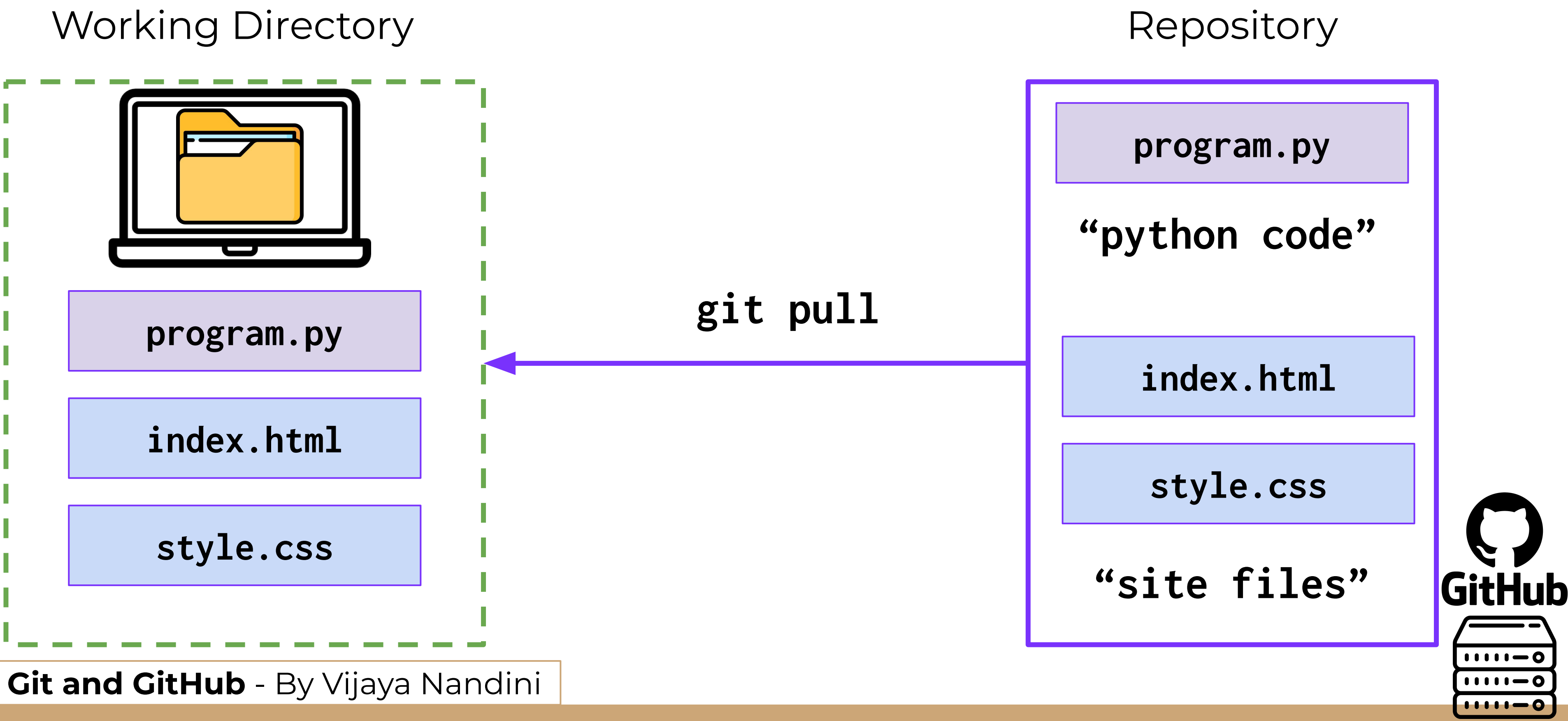
Getting Started with Git



Getting Started with Git



Getting Started with Git



Push and Remote Branches

Getting Started with Git

- We've learned how to create repositories locally and add changes to the staging area and then commit them to the main (master) branch we have locally.
- If you are also using GitHub as a hosting service, we can think of this as a **remote branch**.

Getting Started with Git

- We're still operating under the assumption that the context is just a single solo developer operating on just a single branch, but later on we'll talk about branches in more detail.
- Let's learn how to **push** local code to a **remote branch** on GitHub.

Getting Started with Git

- We can check for remote branches with the command:
  - **git remote -v**
- If you run this command on a cloned repo, you will **view** the URL of the remote branch, like the GitHub URL.
- If there is no connection to a remote branch, then you won't see a URL.

Getting Started with Git

- After we've created a repository locally, we need to create the repository on GitHub.
- Once you've created the repository on GitHub, you will actually see the instructions under: ***"...or push an existing repository from the command line"***

Getting Started with Git

- We tell git we want to add a remote branch using the git remote command syntax:
  - **git remote add name https://url.git**
- By convention, we call this remote branch the **origin** branch.
  - **git remote add origin https://url.git**
- You then replace the .git URL with the .git URL from the repository you created.

Getting Started with Git

- **Important Note:**
  - We won’t use these commands, but just in case you need them in the future:
    - **git remote rename <old> <new>**
    - **git remote remove <name>**

Getting Started with Git

- Once we’ve connected to our remote branch on GitHub, we can **push** our code to the remote branch.
- We tell git to push to the remote main/master branch called origin with the command:
    - **git push -u origin main/master**

Getting Started with Git

- **Important Note:**
  - GitHub has officially changed the naming convention of its **master** branch to **main** branch.
  - You'll see this reflected in the instructions that GitHub provides:
    - **git branch -M main**

Getting Started with Git

- **Git Log**
  - Before we jump into using git fetch and git pull, let's understand **git log**.
  - The **git log** command will show a list of all the commits made to a repository, including the hash, message, and metadata.
  - Think of it as the history of a repo.

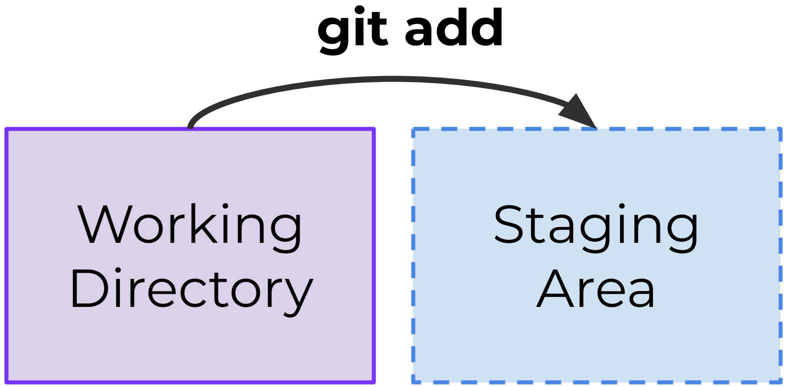
Fetch and Pull

Getting Started with Git

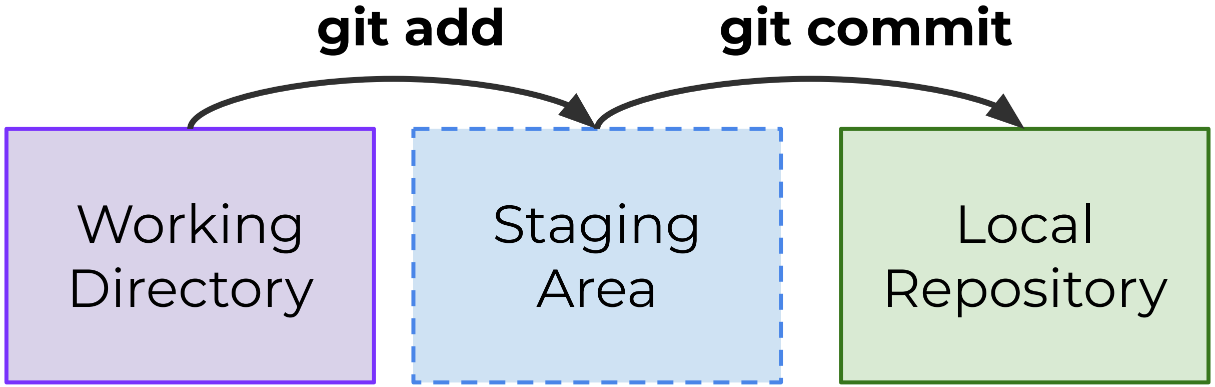
- There are two options of getting repository changes from a remote branch (like the remote branch on GitHub).
    - **git pull**
    - **git fetch**

Getting Started with Git

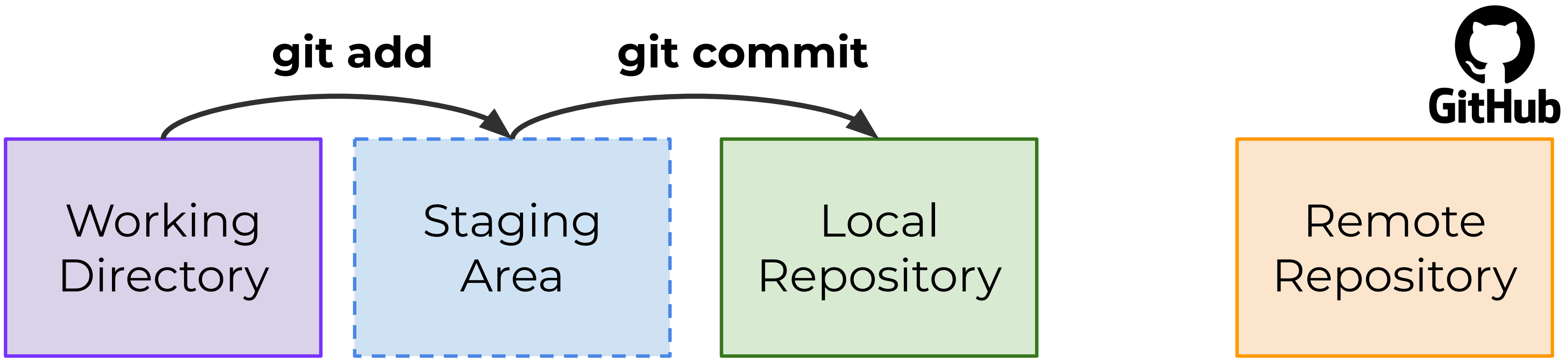
Getting Started with Git



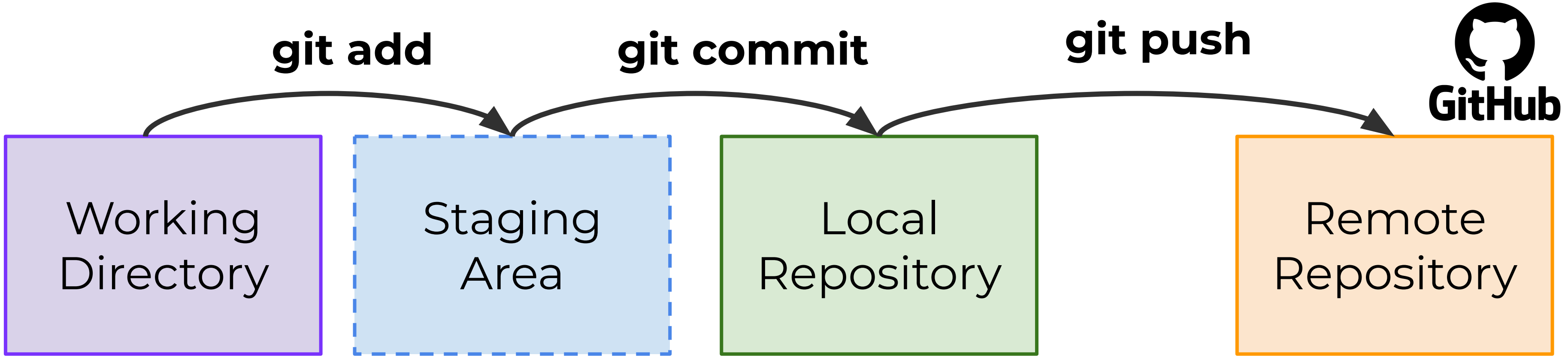
Getting Started with Git



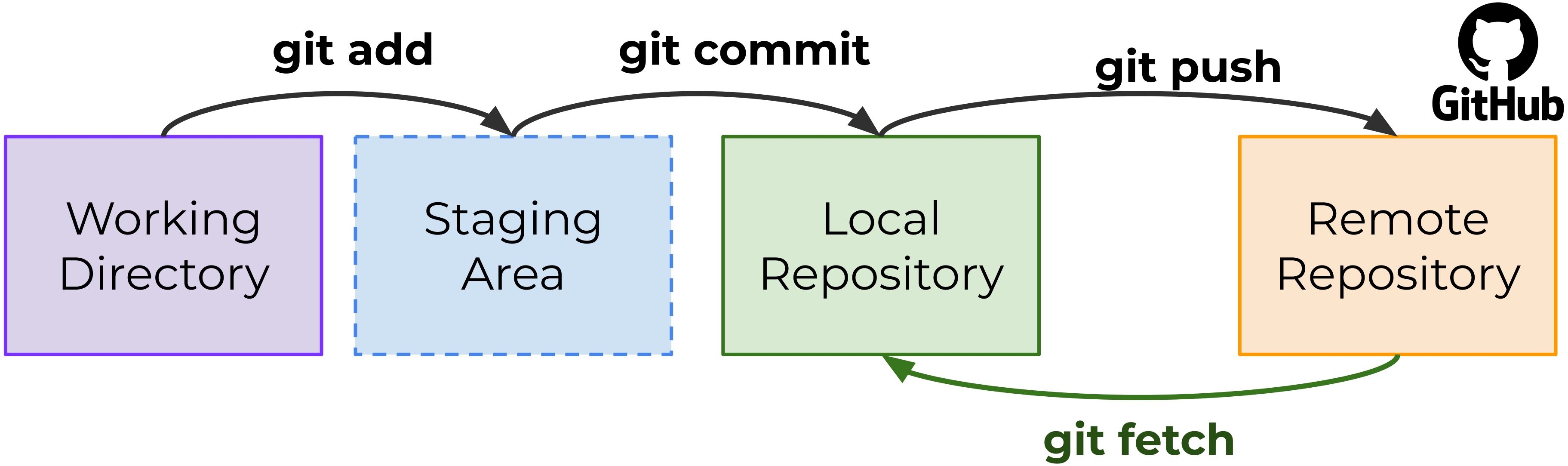
Getting Started with Git



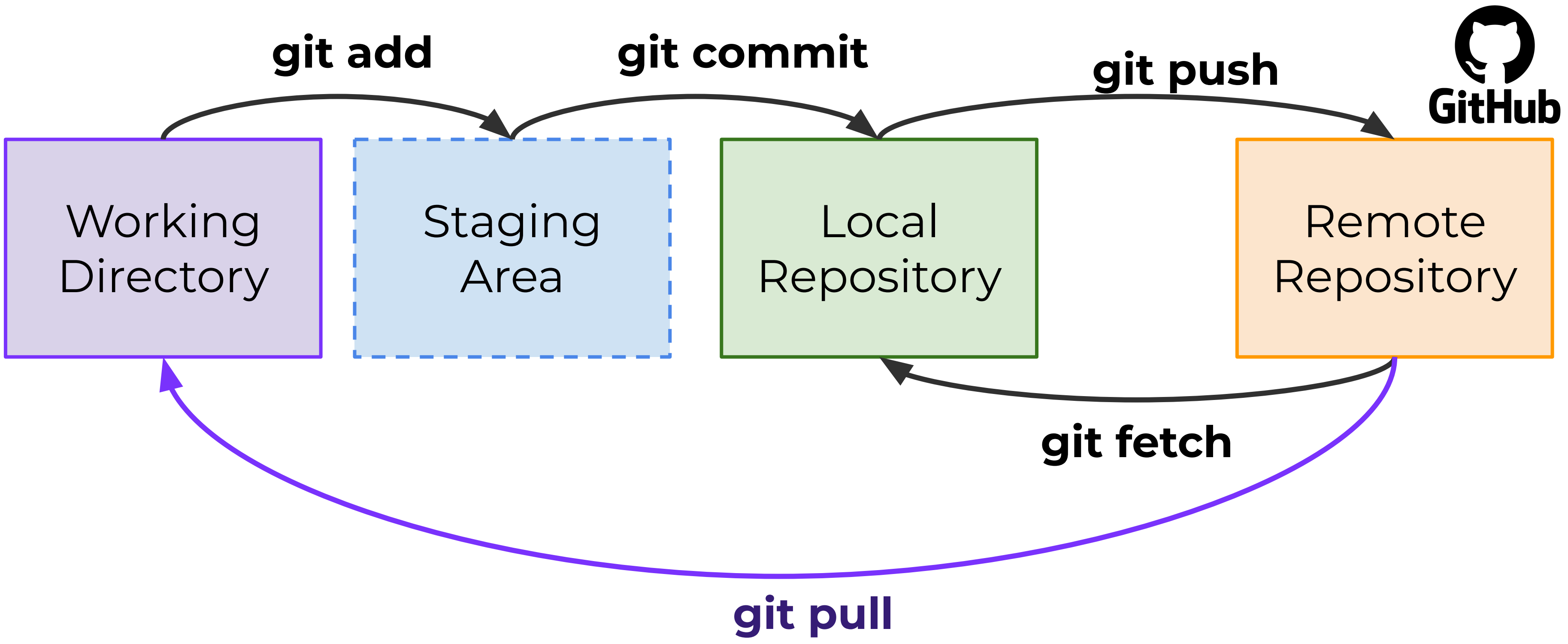
Getting Started with Git



Getting Started with Git



Getting Started with Git



Getting Started with Git

- Using **git fetch** will download changes from the GitHub remote repository, however you will not see those changes be part of the files you have in the working directory.
- Fetch allows you to grab additional work done on the remote master branch, without needing to merge it in your working directory files.

Getting Started with Git

- Using **git fetch** makes sense when you're working with others and want to see what changes they've made but aren't ready to overwrite your own files yet.
- Also if you are simply working by yourself, you may want to just fetch remote changes without overwriting your latest work (later we'll discover branches are a better way of doing this).

Getting Started with Git

- When using fetch, often you'll just use:
  - **git fetch**
- But you can specify to fetch specific branches using:
  - **git fetch <remote> <branch>**
  - **git fetch origin <branch>**

Getting Started with Git

- Using **git pull** makes sense when you want to directly grab changes from the remote branch and directly merge them into your current branch.
- This means you will literally update the files in your working directory to match up and merge with the remote branch.

Getting Started with Git

- If you're a solo developer working on a single master branch, you often skip using a combination of **git fetch** and **git merge** and go straight for a **git pull**.
- We're not going to condone this as the "best practice", but we also want to be realistic of the workflow of a solo developer on a single branch.

Getting Started with Git

- Later we learn about branches and merging changes more carefully or stashing current changes, we’ll have a more nuanced understanding of using **git fetch** vs. **git pull**.
- In general you should pull before pushing to a remote branch, to make sure you’re in sync.

Exercise and Solution

Getting Started with Git

- Let’s test your understanding of what you’ve learned so far with a quick exercise.

Getting Started with Git

- **Tasks:**
  - Create a new local repo
  - Add a text file locally
  - Create the remote repo on GitHub
  - Push the Local repo changes to GitHub
  - Create a new file on GitHub
  - Pull these changes from GitHub to the local repo