# git and GitHub

# What is Git?

A free and open source distributed version control system.

Records changes to a file or set of files over time so that you can recall specific versions later.

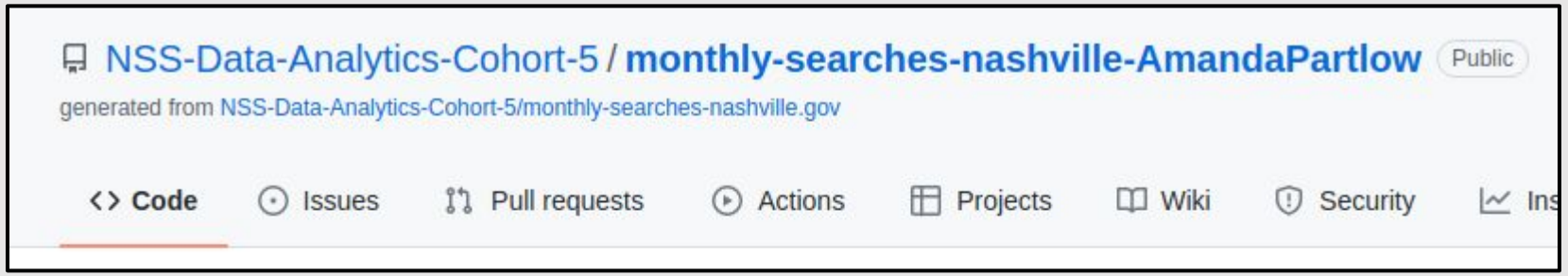
Files and changes to those files are stored in a **repository**.

# What is GitHub?

A hosting service for Git repositories.

Also includes features for collaboration and project management.

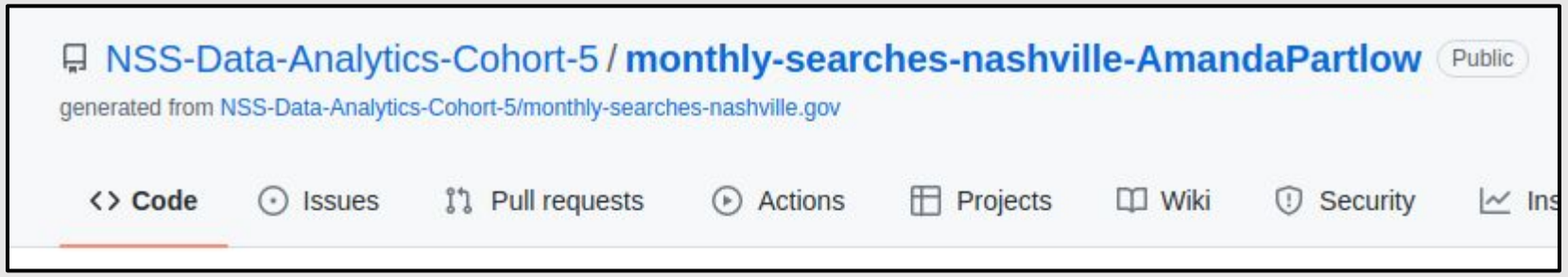
Repository on GitHub




Internet

Your Computer

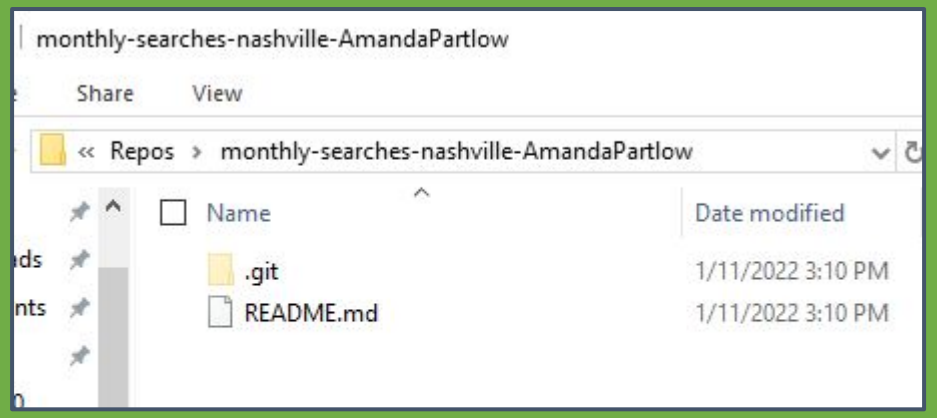
## Repository on GitHub




Internet

Clone

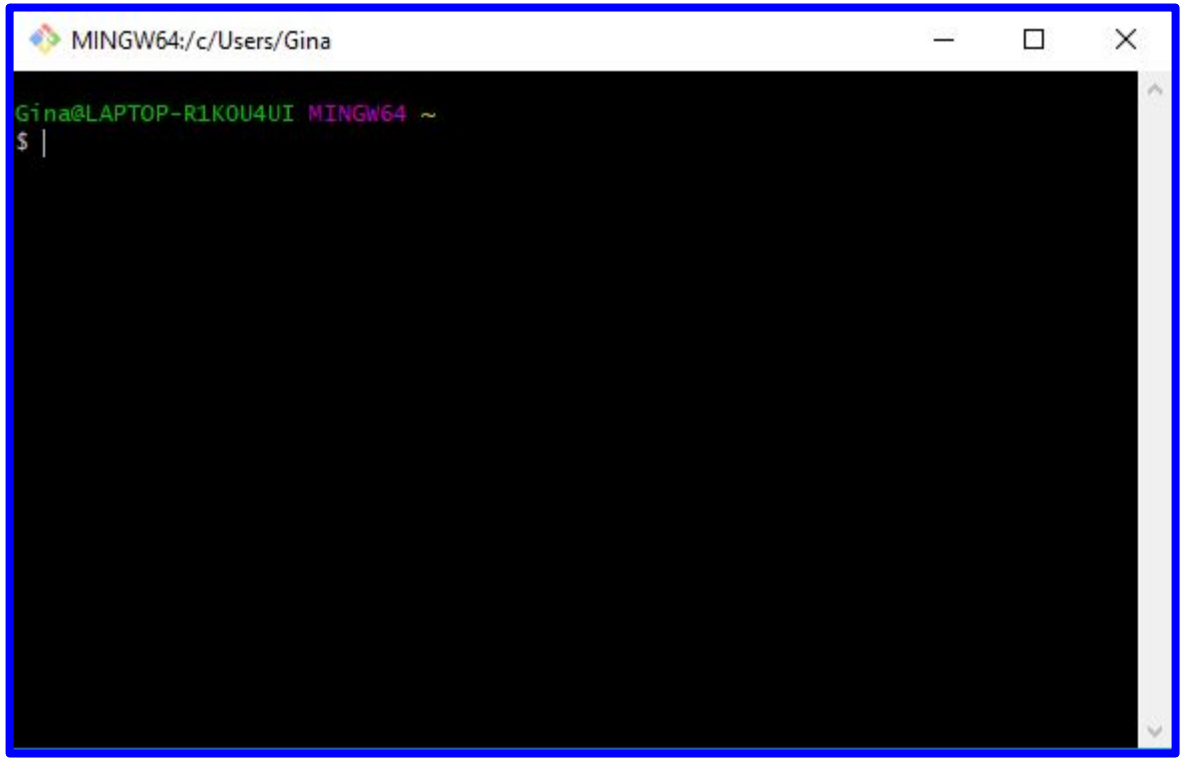
## Your Computer




Local Copy of Repository

# Setup

- Since we will be working in Windows for our Excel projects, everyone should be working on their Windows machine.
- Install git (if not already done) (https://git-scm.com/downloads)
- After installing, open Git Bash
- If Git Bash didn't install automatically with git, you can get it here: https://gitforwindows.org/



# Command Line Introduction

**pwd**

"print working directory" - tells you where you are located

**ls**

Lists the contents of the current directory

**mkdir <directory name>**

Creates a new directory

**cd <directory name>**

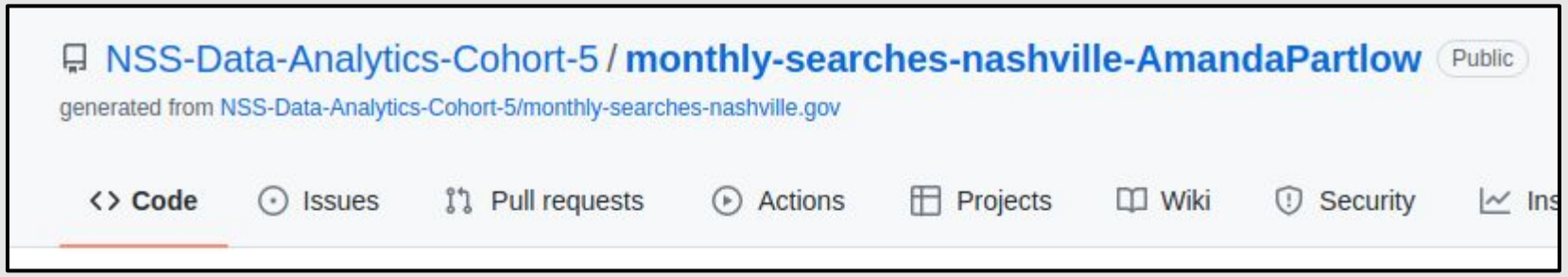
Changes the directory to the one specified

**cd ..**

Takes you up on level in the directory structure

Use these commands to navigate to where you want to save your class files. Create a new directory here named nss-data-analytics (or similar).

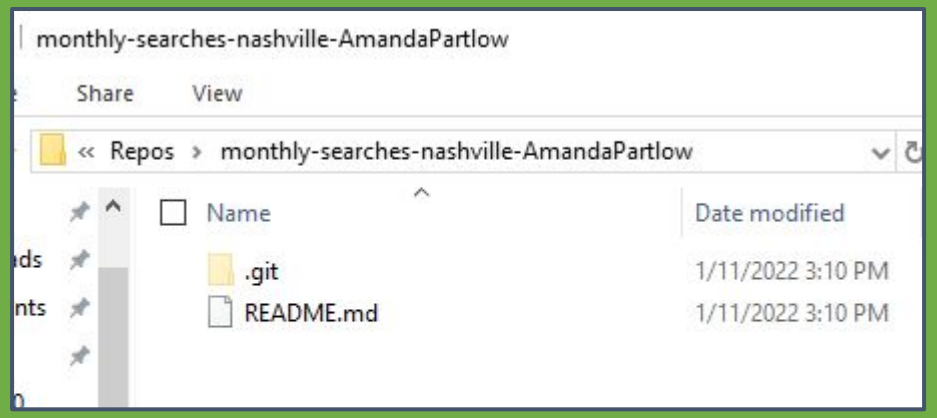
## Repository on GitHub




Internet

Clone

## Your Computer




Local Copy of Repository

# Repository Setup

Navigate in your browser to the repository that was set up when you accepted the assignment. Locate the “Clone or download” button in the upper right:


Clone your remote repository to create a local repo by copying the clone url from github.com and then running from Git Bash:

```
git clone <url to repository that you copied>
```

## Setting your Git username for *every* repository on your computer

1. Open Terminal. **or GitBash if you are on Windows**
2. Set a Git username:

```
$ git config --global user.name "Mona Lisa"
```

3. Confirm that you have set the Git username correctly:

```
$ git config --global user.name
> Mona Lisa
```

Your username is the account name. In the example below it would be **MaryLvV**. It is below your actual name and in the URL.

## Setting your commit email address in Git

You can use the `git config` command to change the email address you associate with your Git commits. The new email address you set will be visible in any future commits you push to GitHub from the command line. Any commits you made prior to changing your commit email address are still associated with your previous email address.

### Setting your email address for every repository on your computer

1. Open Terminal. **or GitBash if you are on Windows**
2. Set an email address in Git. You can use your GitHub-provided `no-reply` email address or any email address.

```
$ git config --global user.email "email@example.com"
```

3. Confirm that you have set the email address correctly in Git:

```
$ git config --global user.email
email@example.com
```

# Adding and Committing Changes

After finishing a unit of work (eg. finishing one of the project questions, end of class meeting), you should save and **commit** your changes.

1. First, add your changes to the **staging area**

```
git add <files>
```

2. Then commit those changes with a *descriptive* message (eg. “Completed question 3”):

```
git commit -m '<message>'
```

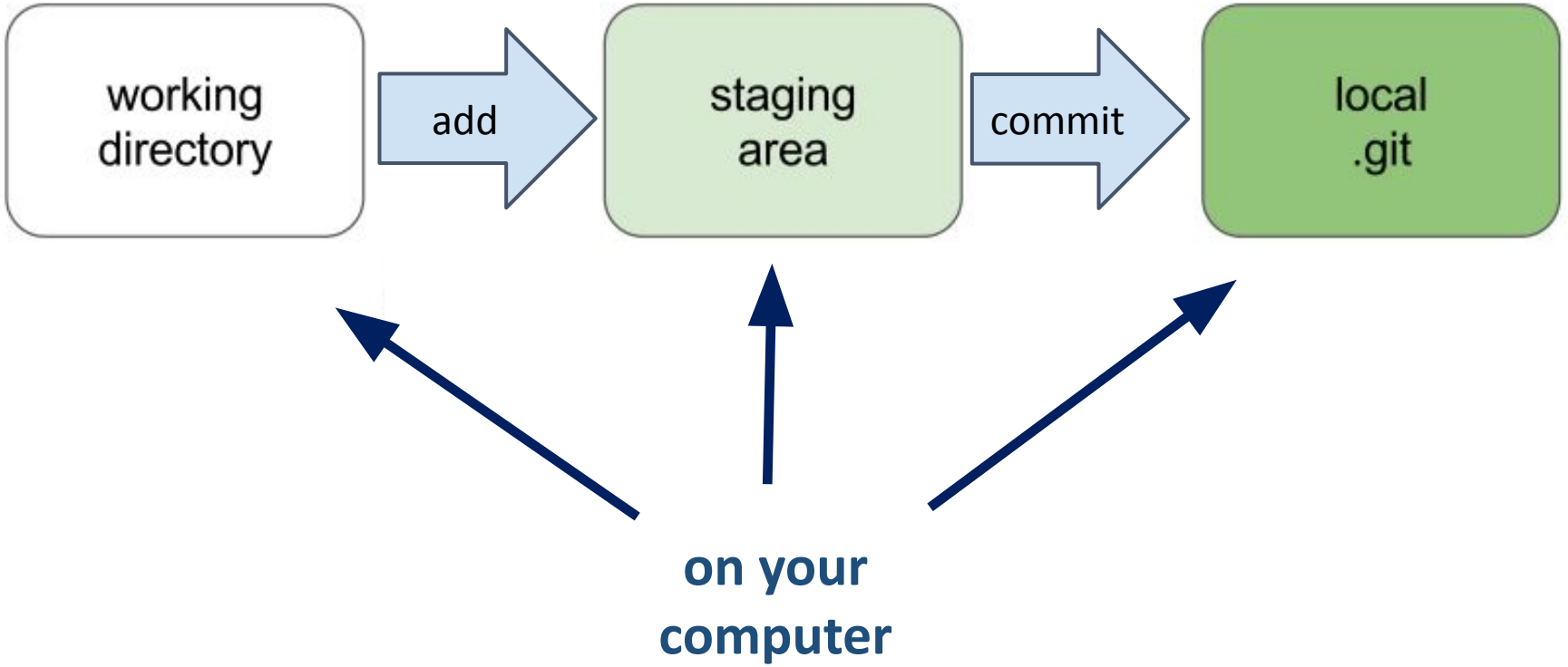
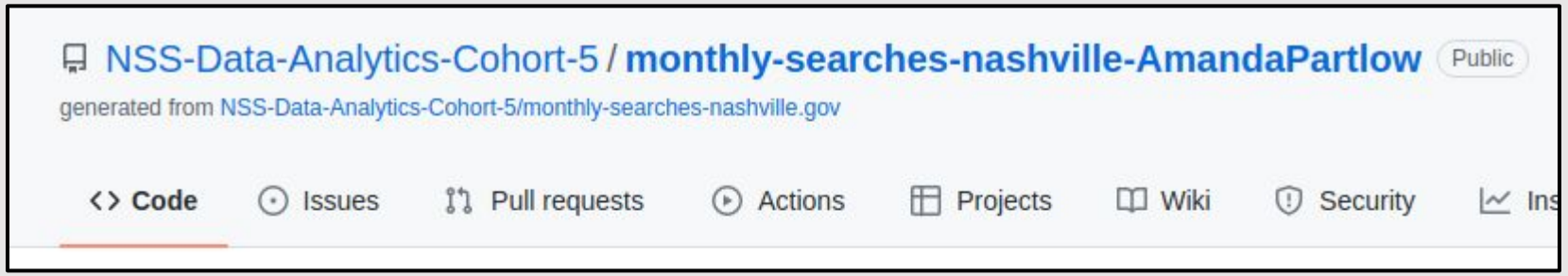



Image: https://edav.info/github.html

Repository on GitHub

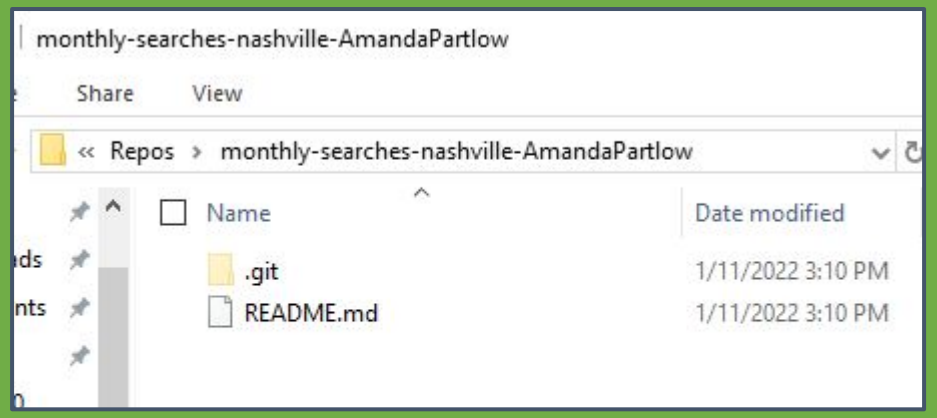



Internet

Clone

Push

Your Computer




Local Copy of Repository

# Create a PAT

Before pushing changes, we must create a personal access token on GitHub.

1. Open *settings* by clicking on your profile picture in the upper right corner.
2. In the menu on the left, choose *Developer Settings*, then choose *Personal access tokens*
3. Choose *Generate new token*, give it a name, and ensure that you have selected the **repo** scope
4. Take the resulting token and save it somewhere (or email to yourself).

# Recommended Workflow

1. Work on files locally. When done, run

   ```
   git add <files>
   ```

2. Commit changes:

   ```
   git commit -m '<message>'
   ```

3. Push to GitHub:

   ```
   git push origin main
   ```

   When prompted for a password, paste your PAT.

The first two steps have already been done.

# Recommended Workflow

1. Work on files locally. When done, run

```
git add <files>
```

2. Commit changes:

```
git commit -m '<message>'
```

3. Push to GitHub:

```
git push origin main
```

origin indicates that we are pushing to GitHub

When prompted for a password, paste your PAT.

# Recommended Workflow

1. Work on files locally. When done, run

```
git add <files>
```

2. Commit changes:

```
git commit -m '<message>'
```

3. Push to GitHub:

```
git push origin main
```

We'll learn about what main means later.

When prompted for a password, paste your PAT.

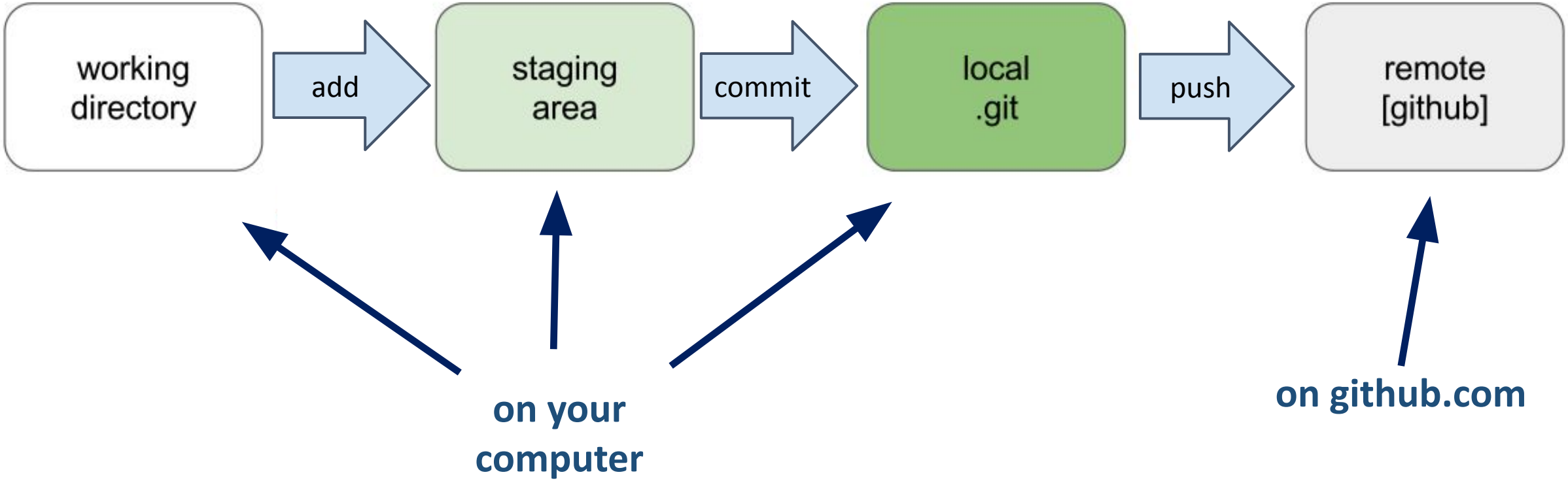



Image: https://edav.info/github.html