

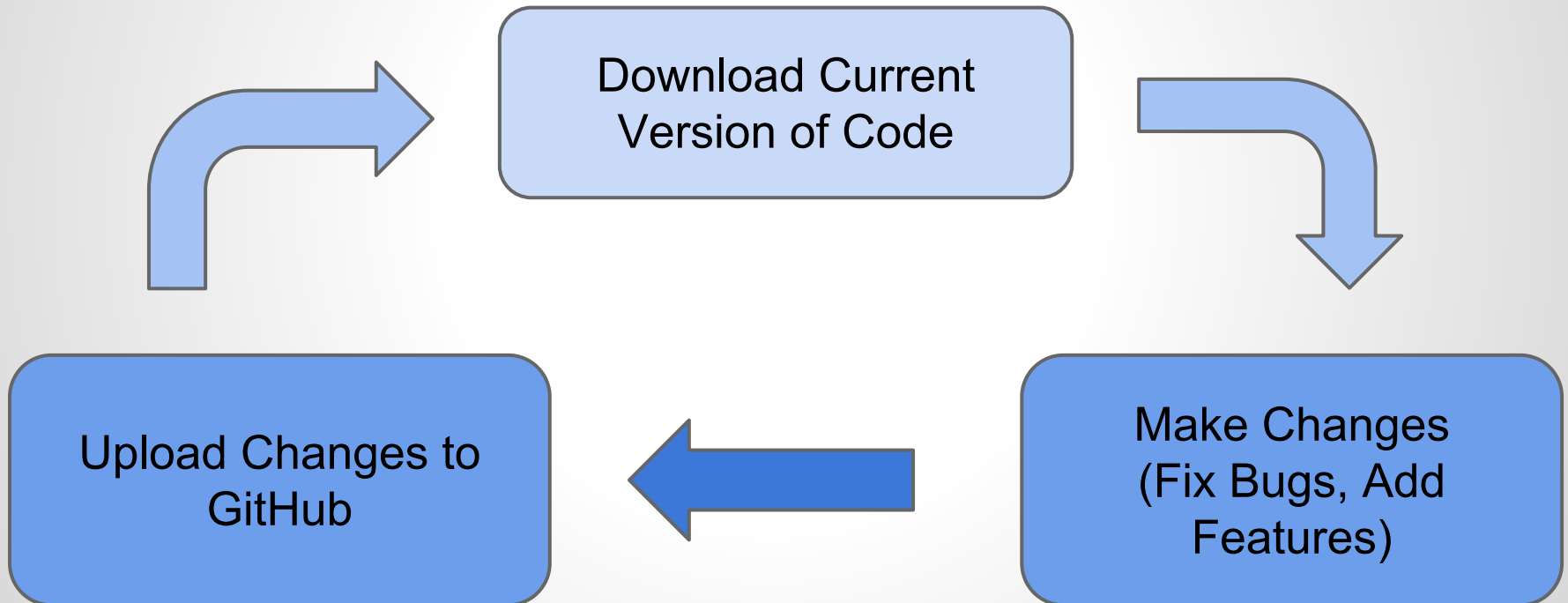
# **CODE4SAC**

GitHub Workshop

# What is GitHub?

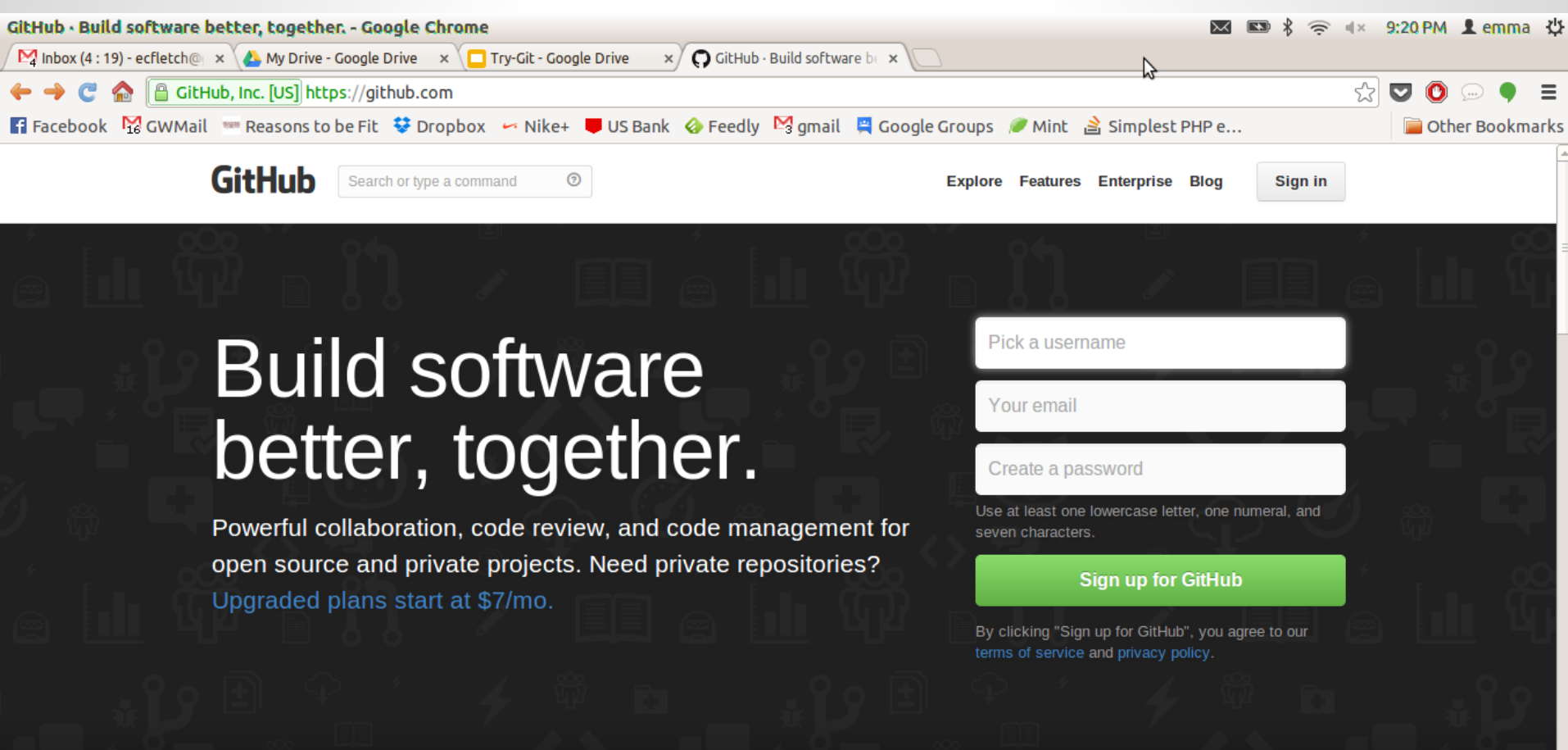
GitHub is a web-based version control system that allows coders to work collaboratively on projects.

# How It Works



# Create a User Profile

Go to [github.com](https://github.com)



GitHub · Build software better, together. - Google Chrome

4 Inbox (4 : 19) - ecfletch@ x My Drive - Google Drive x Try-Git - Google Drive x GitHub · Build software better, together. x

GitHub, Inc. [US] <https://github.com>

Facebook GWMail Reasons to be Fit Dropbox Nike+ US Bank Feedly gmail Google Groups Mint Simplest PHP e... Other Bookmarks

GitHub Search or type a command

Explore Features Enterprise Blog Sign in

# Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#).

# Install Git

Follow the instructions for your operating system

<http://git-scm.com/downloads>

**Windows Users:** For this workshop we will be using the Git Bash command line interface, you may download both the GUI and bash versions of Git but we will not be covering the GUI

# Set-Up Git

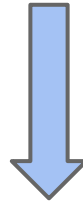
Open the terminal window and type the follow commands

```
$ git config --global user.name "Your User Name Here"
```

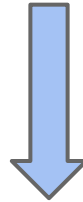
```
$ git config --global user.email "your_email@example.com"
```

# Choose Your Repository

In search box type 'code4sac'



Click on the 'git-lessons' repository



Here you can browse the code and see what other developers have been up to, take a look around

# git clone

**clone:** copies down an existing repository from GitHub

```
$ git clone git://github.com/code4sac/git-lessons.git
```



# `git pull`

**pull:** gives you the latest version of the code, it pulls down any changes that have been made since you last synched with the GitHub repository

```
$ git pull
```

# Lets Make Some Changes

1. Navigate to git-lessons/about
2. Create a directory titled <Your Name>/
3. Create a file called bio.txt

For example my directory structure now looks like

`git-lessons/about/emma/`

and has a file inside called bio.txt

`git-lessons/about/emma/bio.txt`

# Tell Me About Yourself

Write something about yourself and save the file

```
Hi, my name is Emma.
```

```
I like coding things that matter.
```

Add a picture or any other file that you feel helps  
describe you

# git status

**status:** shows you any differences between your local version of the code and the current GitHub version of the code

```
$ git status
```

# git add

**add:** adds a file to the staging area. The staging area is where changes we've made get ready to be uploaded to GitHub (but they haven't been uploaded yet)

```
$ git add bio.txt
```

# git commit

**commit:** saves all the changes in the staging area to the local repository

```
$ git commit -m 'adds Emmas bio'
```

The -m is the message flag, the string afterwards is what other developers will see associated with your changes

# git push

**push:** updates your local repository commits to the GitHub repository. Always pull before you push! Your push may be rejected if changes do not match up properly with the GitHub repository

```
$ git pull
```

```
$ git push
```

# Reference

```
$ git clone <repository>
```

```
$ git pull
```

```
$ git status
```

```
$ git add <filename>
```

```
$ git rm <filename>
```

```
$ git commit -m 'message about commit'
```

```
$ git push
```