Git & GitHub Workshop

Part 1

Hello!

I am Joshua Obogbaimhe

I'm the president of the ACM Computer Technology club and use Git and GitHub for almost all my projects. They are very powerful tools and are worth learning about to structure your development process.

Overview

- ➔ Brief history of Git & GitHub
- ➔ How the two are hosted
- ➔ Why use them?
- ➔ Setting up your GitHub account
- ➔ Creating a project
- ➔ Managing the project with Git

A Brief History

Git

- ➔ Founded in 2005
- ➔ Created by Linus Torvalds for the Linux Kernel
- ➔ Version control system
- ➔ Supports non-linear workflows
- ➔ Open source

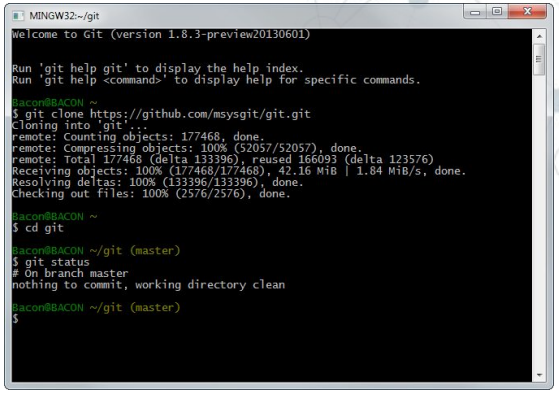
GitHub

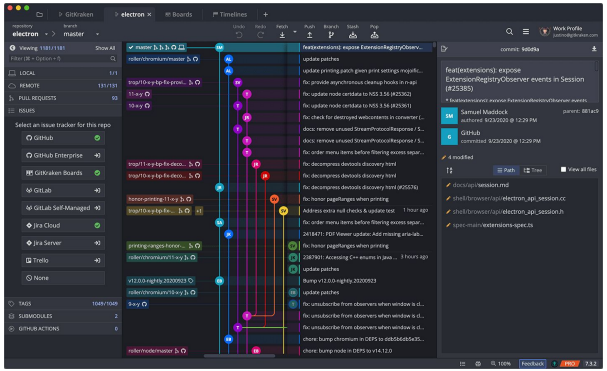
- ➔ Founded in 2008
- ➔ Website that hosts Git repositories
- ➔ Cloud-based
- ➔ Free to use (premium accounts)
- ➔ Social network of developers
- ➔ Issue tracking and collaboration

Hosting

Git

- Git repositories are hosted locally
- ‘.git’ folder contains all information about a project
- Repositories can be managed using a CLI or GUI
- Git Bash is a popular command line tool
- GitKraken and SourceTree are popular GUIs

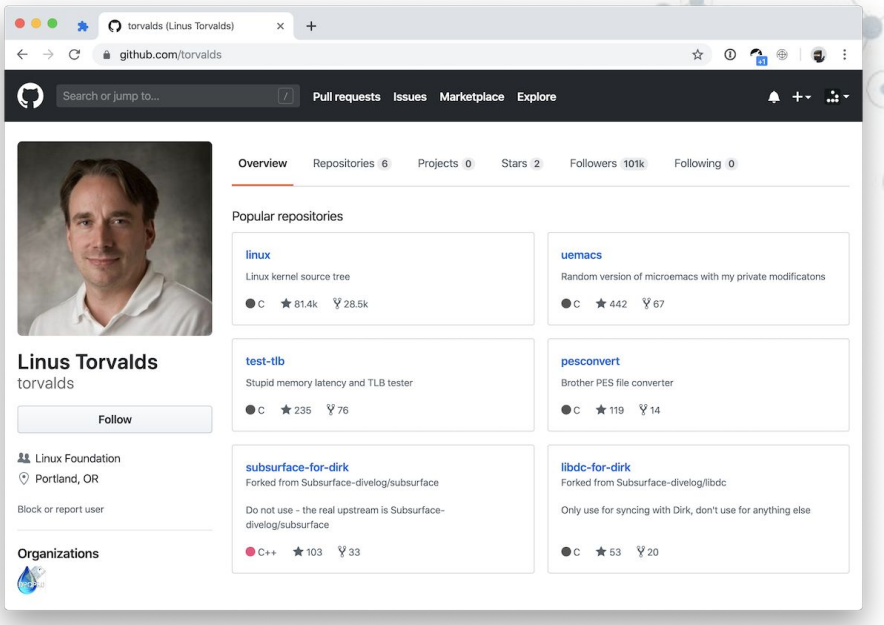




Hosting (cont'd.)

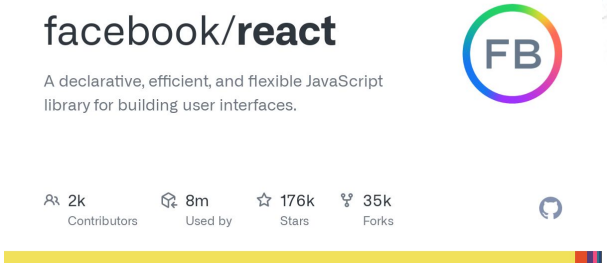
GitHub

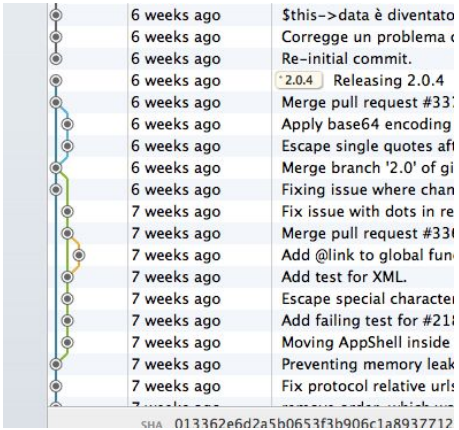
- GitHub repositories are hosted remotely
- Similar to other cloud providers
- Free to create an account
- Work can be viewed by others
- Contains features that allow for collaboration

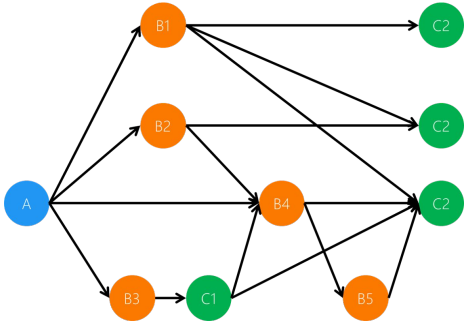


Why use Git and GitHub?

- ➔ Allows you to keep an ordered list of all the changes to a project
- ➔ Roll back changes if needed
- ➔ View popular open source projects and even contribute to them!
- ➔ Work in teams and collaborate using simple to complex workflows
- ➔ Great for recruiters to see your body of work
- ➔ Hosting capabilities



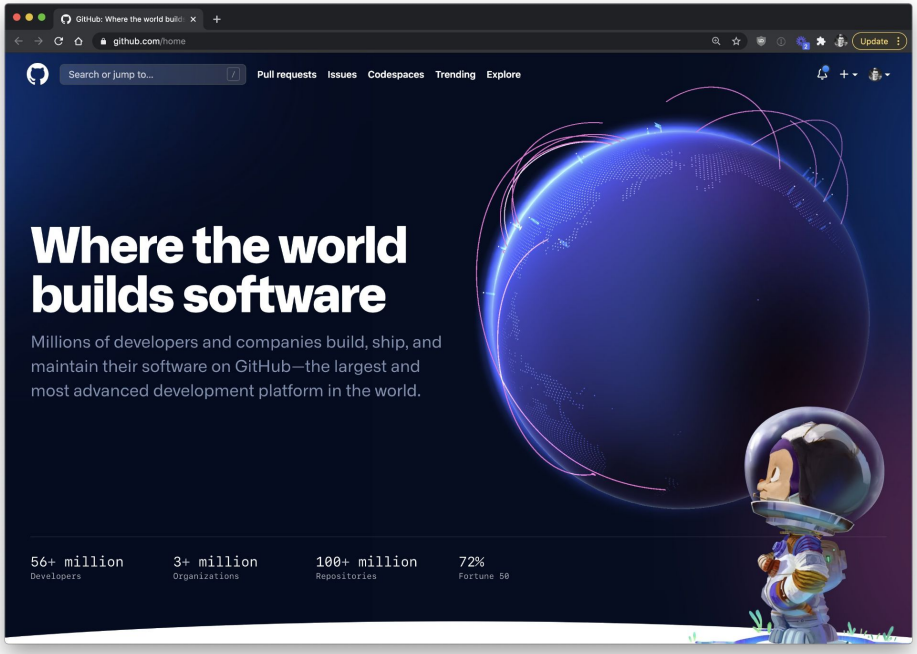




Off to GitHub!

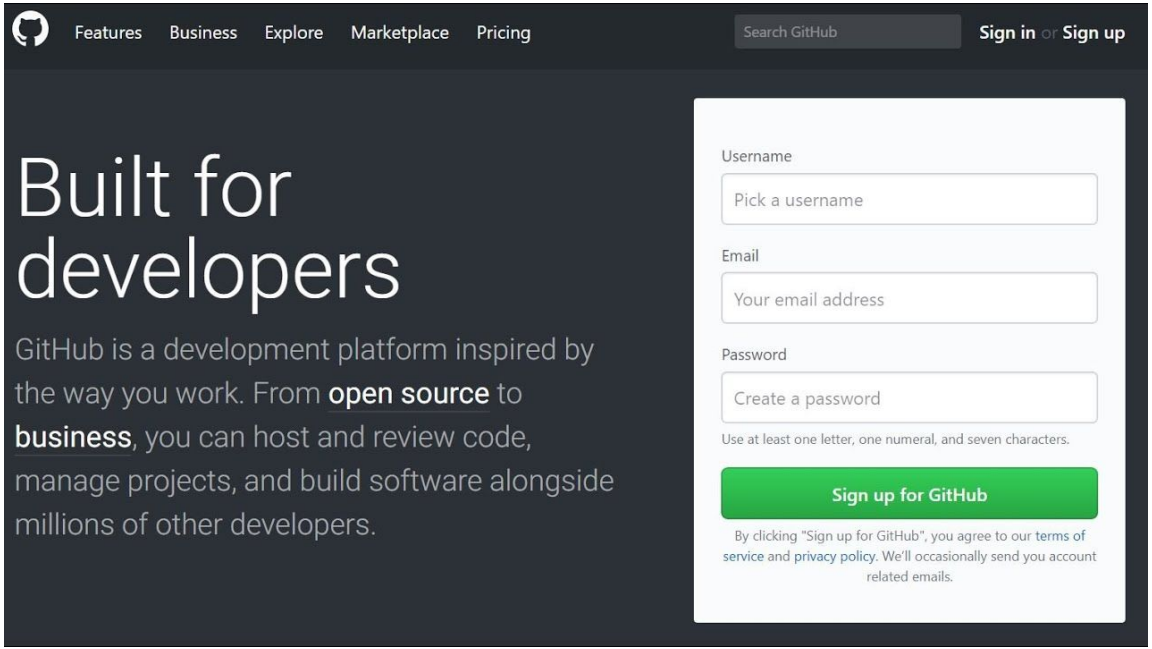
Let’s create a GitHub account and get started

Visiting the website

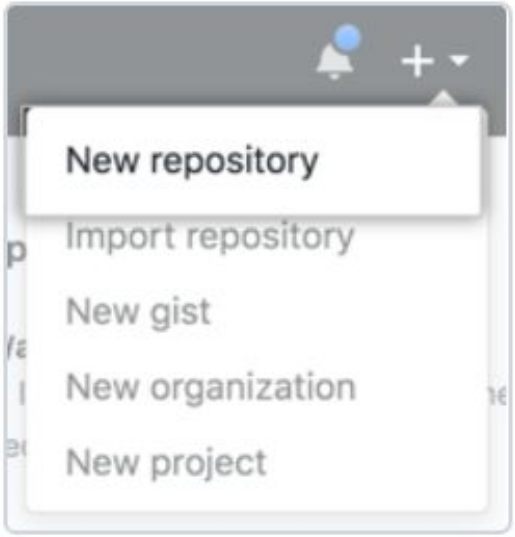


https://github.com/

Creating an account



Creating a repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * **Repository name** *

sammy / my-new-project

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-fiesta**?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

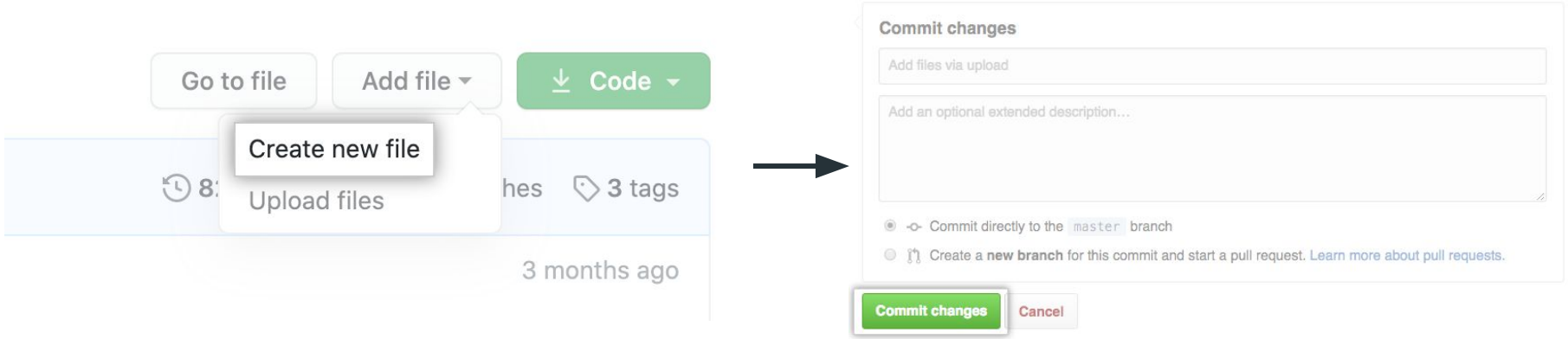
Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

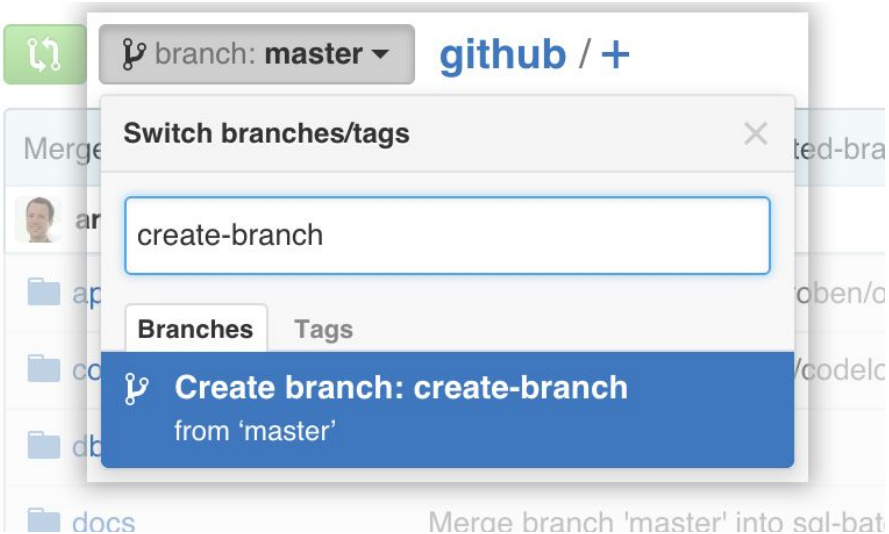
Choose a license
A license tells others what they can and can't do with your code. Learn more.

Create repository

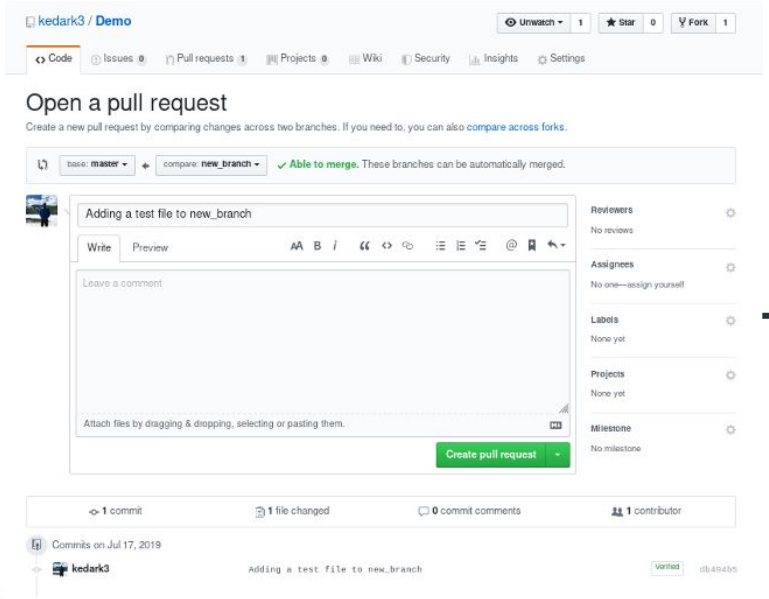
Creating a file

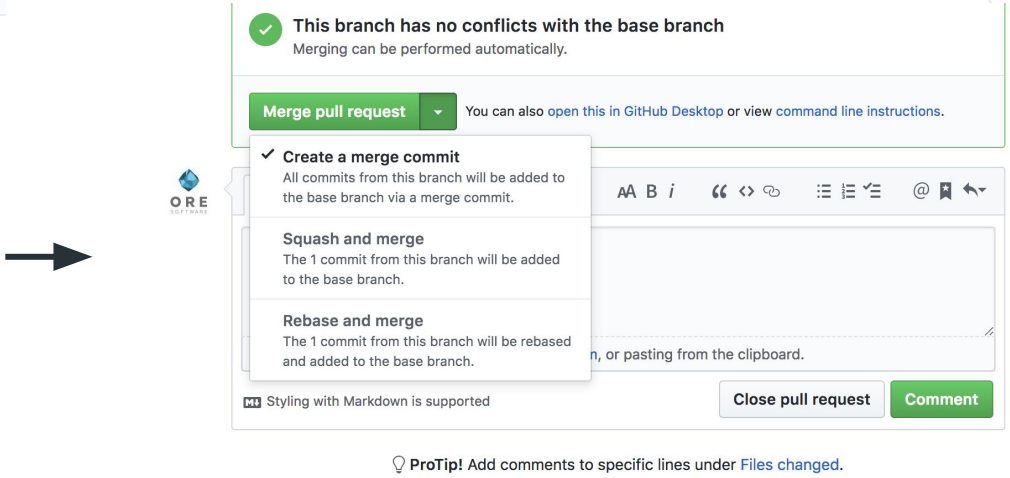


Creating a branch



Merging branches

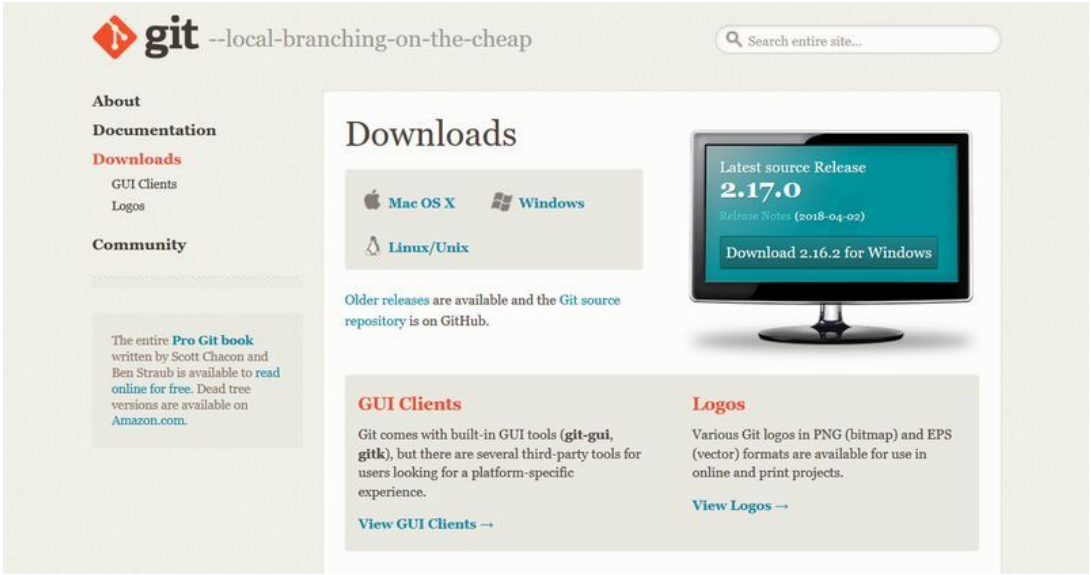




Off to Git!

Let’s install Git and get started

Installing Git



https://git-scm.com/downloads

git config

git-config - Get and set repository or global options

```
1. git config --global user.name "John Doe"
2. git config --global user.email "johndoe1984@gmail.com"
```

git init

git-init - Create an empty Git repository or reinitialize an existing one

```
1. git init
```

git remote

git-remote - Manage set of tracked repositories

```
1. git remote add my_project http://github.com/my-account/my-repository.git
```

git pull

git-pull - Fetch from and integrate with another repository or a local branch

```
1. git pull my_project master
```

git add

git-add - Add file contents to the index

```
1. git add cool_file.txt
```

or

```
1. git add .
```

git rm

git-rm - Remove files from the working tree and from the index

```
1. git rm cool_file.txt
```

git status

git-status - Show the working tree status

```
1. git status
```

git commit

git-commit - Record changes to the repository

```
1. git commit -m "Intial commit"
```

git push

git-push - Update remote refs along with associated objects

```
1. git push
```

git log

git-log - Show commit logs

```
1. git log
```

git checkout

git-checkout - Switch branches or restore working tree files

```
1. git checkout branch_c
```

git diff

git-diff - Show changes between commits, commit and working tree, etc

```
1. git diff branch_a branch_b
```

Questions?