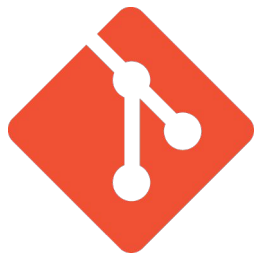


Git & GitHub Workshop

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



```
MINGW32~/git
Welcome to Git (version 1.8.3-preview20130601)

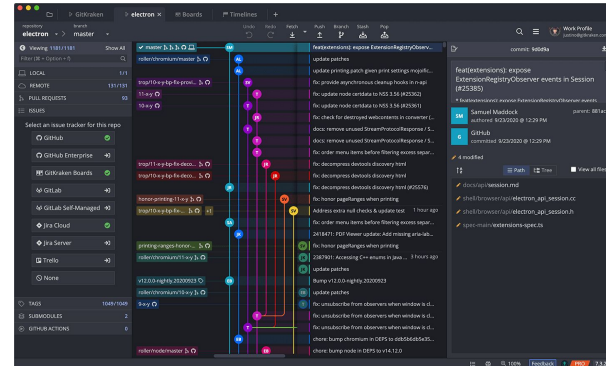
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

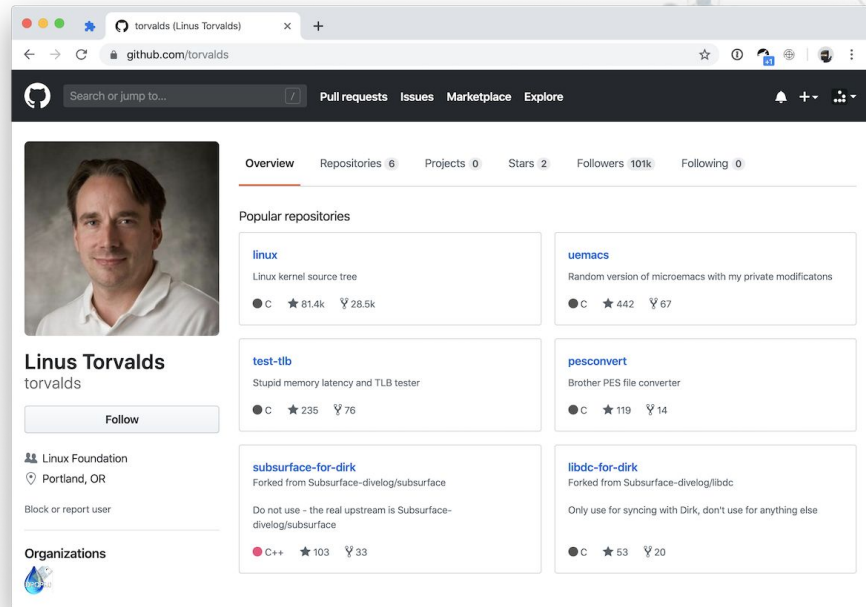
Bacon@BACON ~/git (master)
$
```



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?

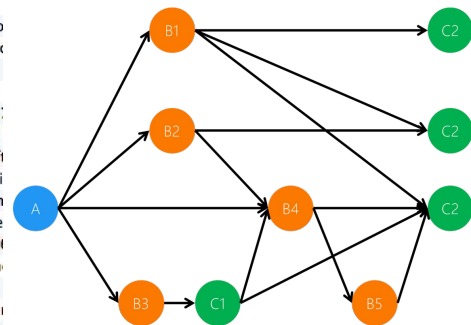
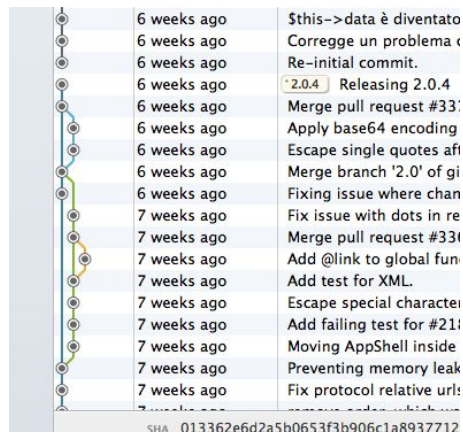
- Allows you to keep an ordered list of all the changes to a project
- Roll back changes if needed
- View popular open source projects and even contribute to them!
- Work in teams and collaborate using simple to complex workflows
- Great for recruiters to see your body of work
- Hosting capabilities

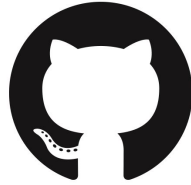
facebook/react

A declarative, efficient, and flexible JavaScript library for building user interfaces.



2k Contributors 8m Used by 176k Stars 35k Forks

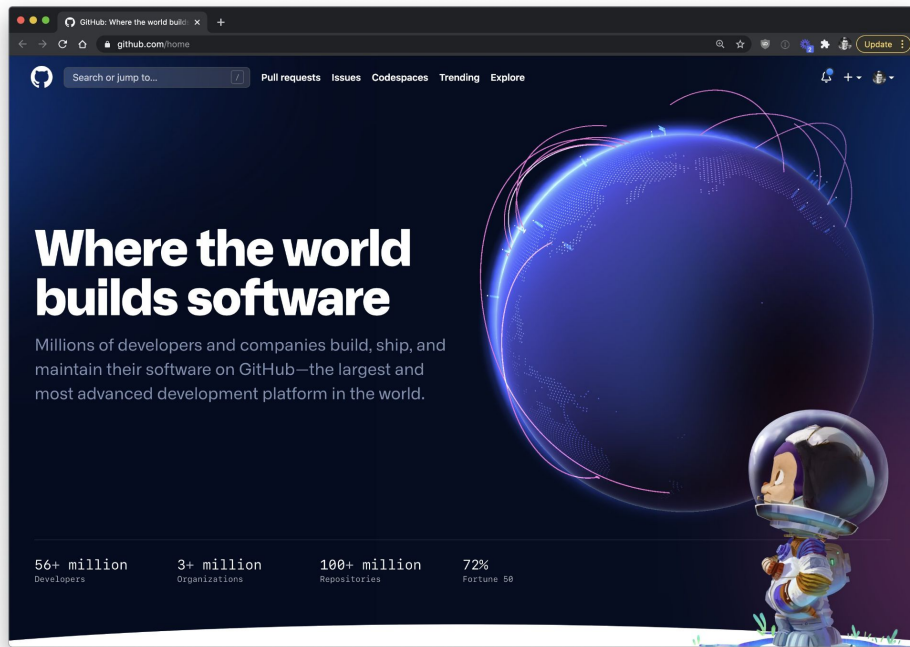




Off to GitHub!

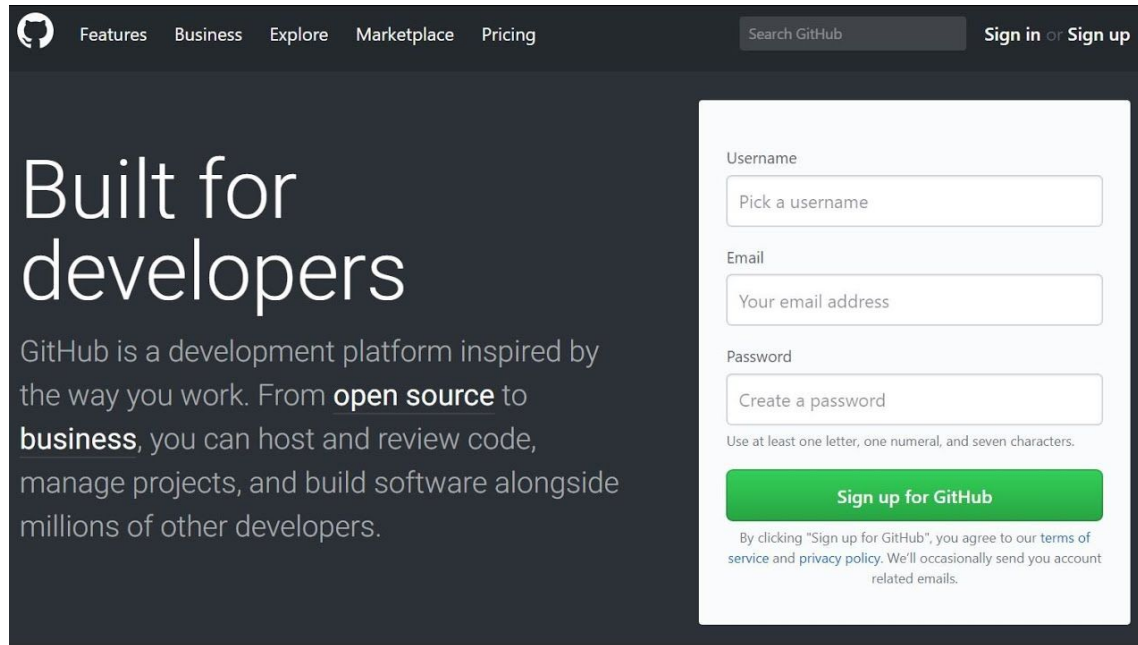
Let's create a GitHub account and get started

Visiting the website



<https://github.com/>

Creating an account

A screenshot of the GitHub website's sign-up page. The page has a dark blue header with the GitHub logo, navigation links (Features, Business, Explore, Marketplace, Pricing), a search bar, and 'Sign in or Sign up' links. The main content area is dark blue with the text 'Built for developers' and a paragraph about GitHub. On the right, there is a white sign-up form with fields for Username, Email, and Password, a green 'Sign up for GitHub' button, and a disclaimer about terms of service and privacy policy.

GitHub

Features Business Explore Marketplace Pricing

Search GitHub

Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

Your email address

Password

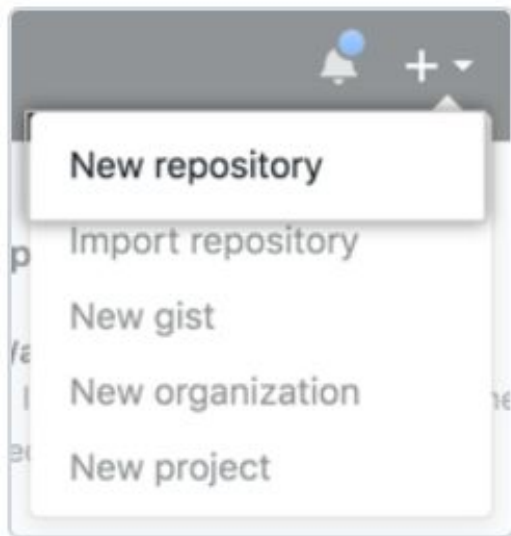
Create a password

Use at least one 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](#). We'll occasionally send you account related emails.

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 *

 sammy

Repository name *

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**

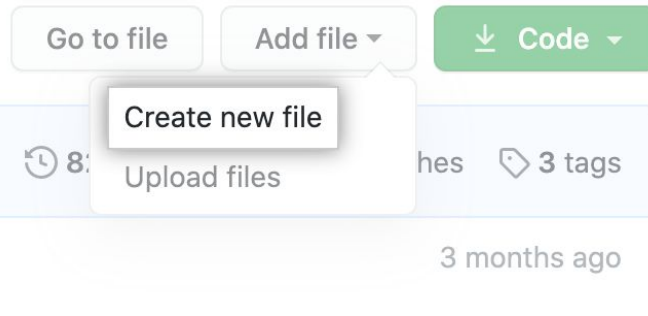
Choose which files not to track from a list of templates. [Learn more](#).

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more](#).

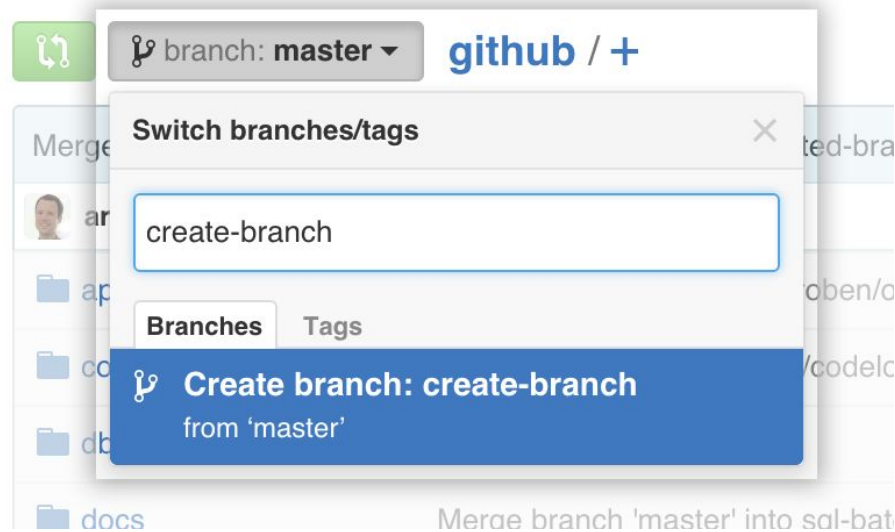
Create repository

Creating a file

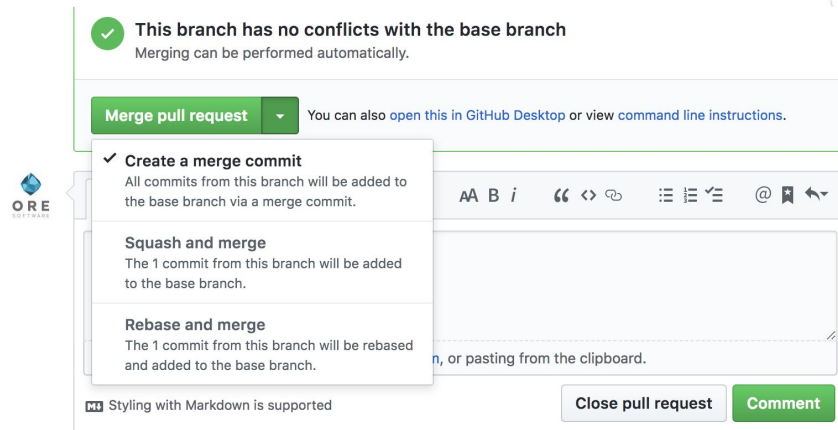
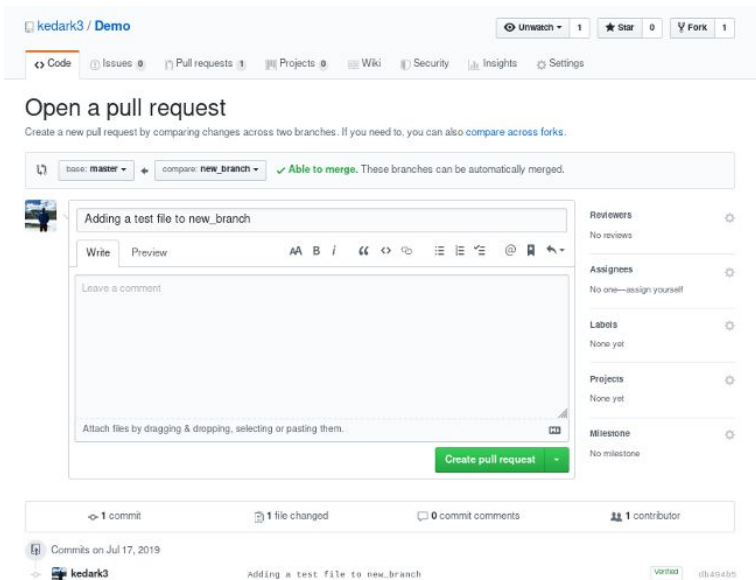


A screenshot of the 'Commit changes' dialog box in GitHub. The dialog has a title 'Commit changes'. Below the title, there is a text input field labeled 'Add files via upload'. Below that is a larger text area labeled 'Add an optional extended description...'. At the bottom, there are two radio button options: 'Commit directly to the master branch' (selected) and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the very bottom, there are two buttons: 'Commit changes' (green) and 'Cancel' (gray).

Creating a branch



Merging branches



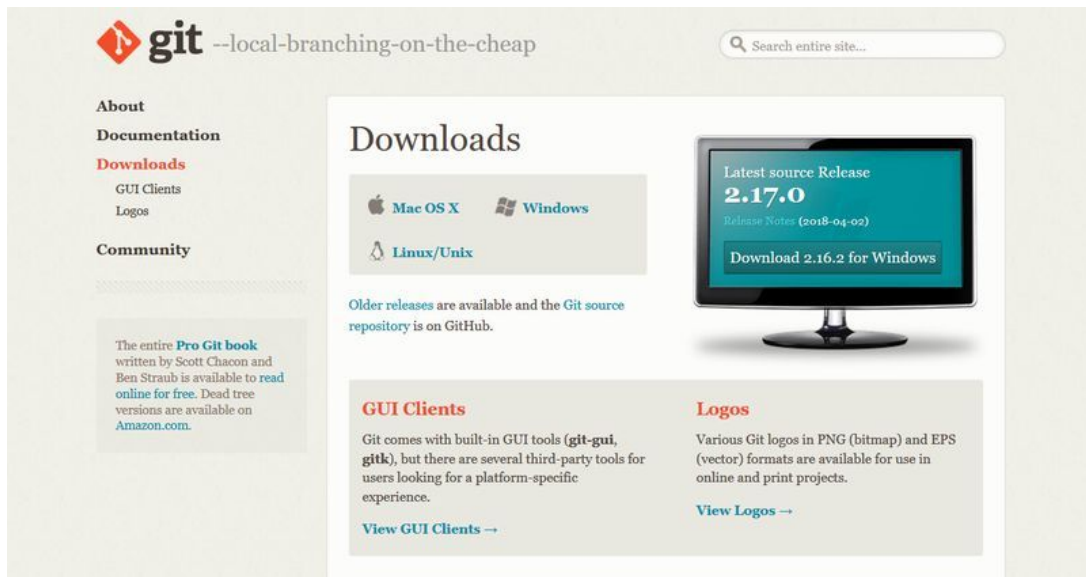
💡 ProTip! Add comments to specific lines under [Files changed](#).



Off to Git!

Let's install Git and get started

Installing Git



The screenshot shows the Git website's Downloads page. At the top left is the Git logo and tagline "--local-branching-on-the-cheap". A search bar is at the top right. The left sidebar contains links for "About", "Documentation", "Downloads" (highlighted), "GUI Clients", "Logos", and "Community". Below these is a note about the "Pro Git book". The main content area is titled "Downloads" and features a box with links for "Mac OS X", "Windows", and "Linux/Unix". To the right is an image of a monitor displaying the "Latest source Release 2.17.0" and a button to "Download 2.16.2 for Windows". Below the download links, text states "Older releases are available and the Git source repository is on GitHub." At the bottom, there are sections for "GUI Clients" (with a link to "View GUI Clients") and "Logos" (with a link to "View Logos").

git --local-branching-on-the-cheap

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows Linux/Unix

Latest source Release
2.17.0
Release Notes (2018-04-02)
Download 2.16.2 for Windows

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.
[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.
[View Logos →](#)

<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
```

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric rings, suggesting a hierarchical or multi-layered structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

Questions?