

COMP2068 – JavaScript Frameworks

1 Getting Started with GitHub

1. Intro to GitHub and Version Control
2. Installing the GitHub Client
3. Get a GitHub Account
4. Create a Repository
5. Your First Commit

Intro to GitHub and Version Control

GitHub for Version Control

- In this course we will be using GitHub for Version Control
- This will count towards your final mark in each assignment.



What is Version Control?

- Revision control, also known as **version control** and source control (and an aspect of **software configuration management**), is the management of changes to documents, computer programs, large web sites, and other collections of information.
- Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1".
- When the first change is made, the resulting set is "revision 2", and so on.
- Each revision is associated with a **timestamp** and the person making the change.
- Revisions can be **compared**, **restored**, and with some types of files, **merged**.

What is Git?



- **Git** is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.
- **Git** was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.
- As with most other distributed revision control systems every **Git** working directory is a full-fledged **repository** with complete history and full version-tracking capabilities, independent of network access or a central server.

What is GitHub?

GitHub

- **GitHub** is a **Git** repository **web-based hosting service** which offers all of the distributed revision control and source code management (SCM) functionality of **Git** as well as adding many of its own features.
- Unlike **Git**, which is strictly a **command-line tool**, GitHub provides a web-based graphical interface and desktop as well as mobile integration.

3 Good Reasons to use GitHub

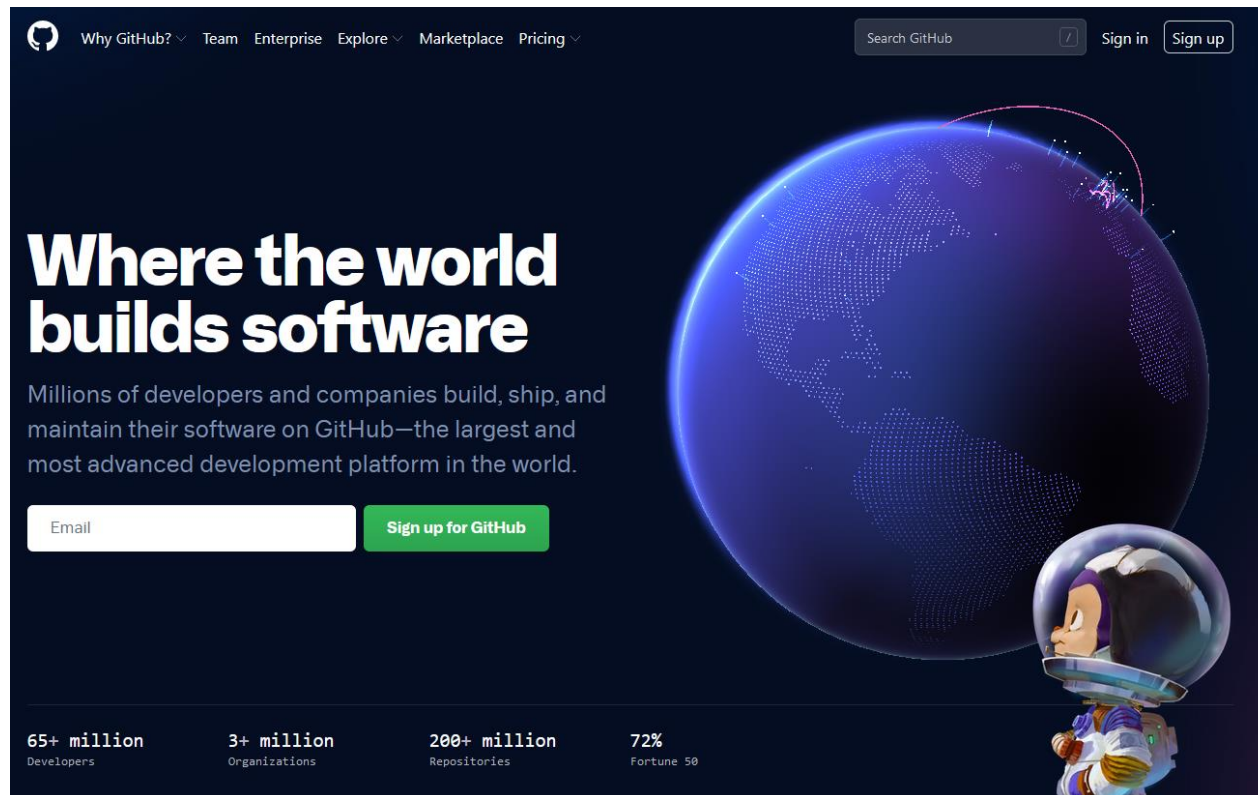
Here are some great reasons to push projects to GitHub:

1. **Version Control** — Everything on GitHub is stored in the **Git** version control system. Version control allows you to experiment and make mistakes in code without messing up your final product.
2. **Keep your code in one place** — Whether you work on multiple computers or just want to get some old projects off your computer, GitHub is the perfect place to store your projects online.
3. **Collaboration** — Once your code is on GitHub, you can invite others to work on your code with you. Send them a link to help you debug a problem.

Get a GitHub Account

Create a GitHub Account

- Go to <http://www.github.com>
- Create an account



Free Plan (after being acquired by Microsoft)

- Free Plan is default for new users with unlimited private repositories
- Click Sign Up and fill in your information

Join GitHub

Create your account

Username *

 ✓

Email address *

 ✓

Password *

 ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.
[Learn more.](#)

Free

The basics for individuals and organizations

- ① Unlimited public/private repositories
- ① 2,000 automation minutes/month
Free for public repositories
- ① 500MB of Packages storage
Free for public repositories
- ① Community support

Welcome to GitHub!



Eduardo Jaime
eduardojaime

Software Engineer with more than 9
years of experience focused on
Microsoft Technologies

[Edit profile](#)

30 followers · 1 following · 10

4Q [Software](#)
eduardojaime86@outlook.com
<https://eduardojaime.github.io/>

Achievements



[Overview](#) [Repositories](#) 27 [Projects](#) [Packages](#)

Pinned

[Customize your pins](#)

[BankAccountsDB](#)

PLpgSQL

[DataStructuresPractice](#)

C#

[RockPaperScissorsConsoleApp](#)

This is a quick Rock-Paper-Scissors Console App demo

C#

[CsharpDataStructures](#)

C# Data Structures project using .NET Core

C#

[PythonPractice](#)

Answers to varied Python exercises

Python

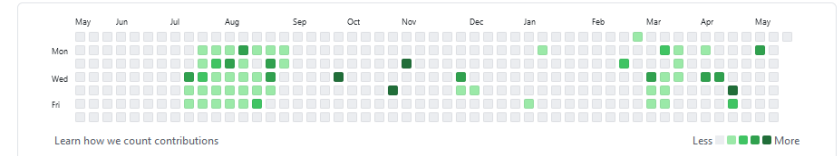
[ContactManager](#)

Demo ContactManager Winforms App to show connection to a database and simple insert

C#

101 contributions in the last year

[Contribution settings](#)



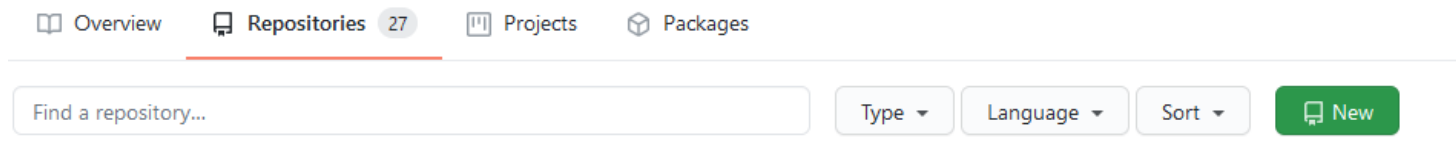
Create a Repository

Create a Repository

- A **repository** is the basic unit of GitHub, most commonly a single project.
- Repositories can contain **folders** and **files**, including images – anything your project needs.

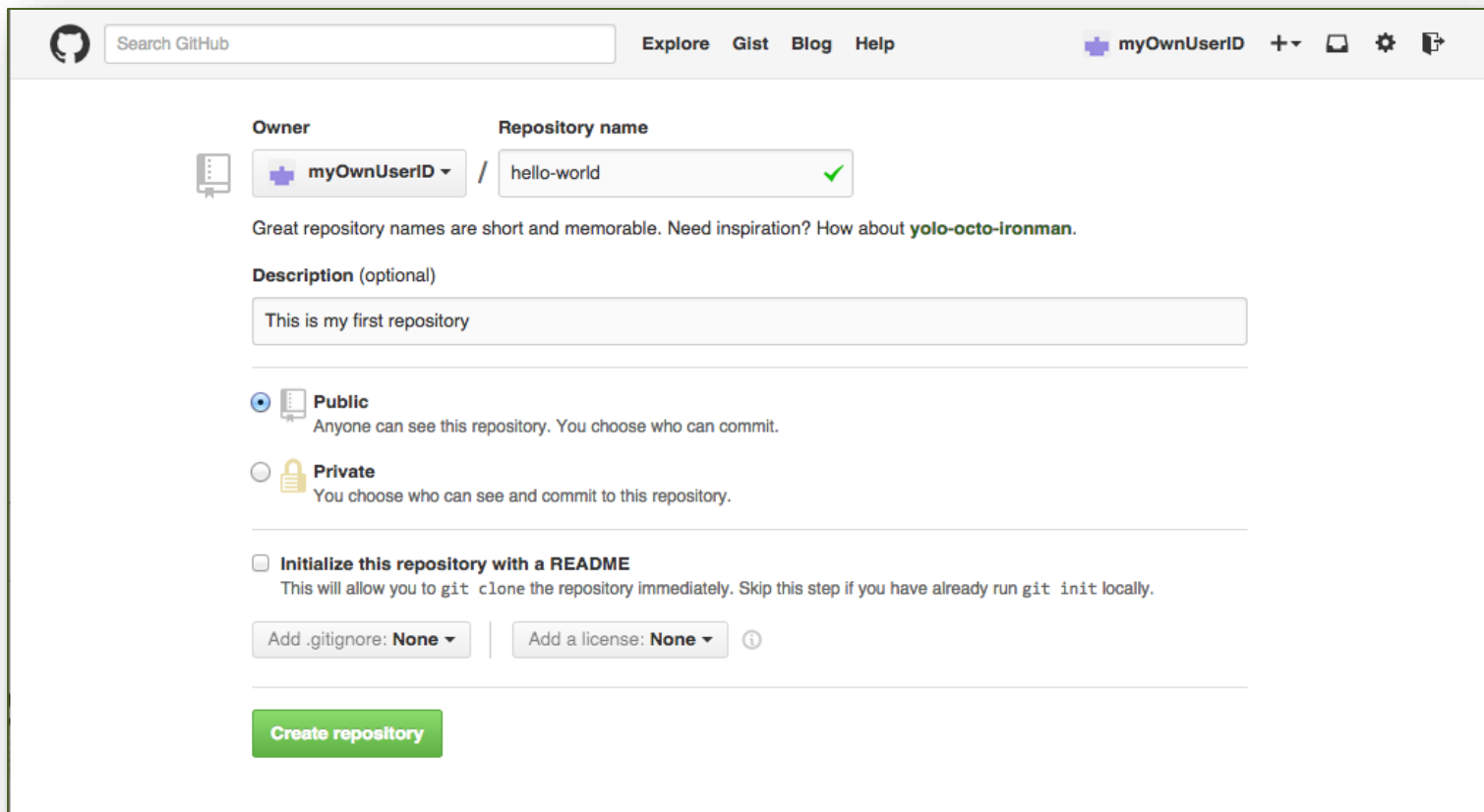
To create a new repository:

1. Navigate to Repositories and click New



Create a Repository (continued)

2. Name your repository (example: hello-world).
3. Write a short description.
4. Click **Create repository**. Boom, repository!



The screenshot shows the GitHub 'Create repository' page. At the top, there's a search bar and navigation links for 'Explore', 'Gist', 'Blog', and 'Help'. The user 'myOwnUserID' is logged in. The main form has two sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'myOwnUserID' and the 'Repository name' is 'hello-world', which is marked as valid with a green checkmark. Below this, a tip suggests repository names should be short and memorable, with an example 'yolo-octo-ironman'. The 'Description (optional)' field contains the text 'This is my first repository'. Under the 'Visibility' section, 'Public' is selected, indicating that anyone can see the repository. The 'Private' option is also available. There's a checkbox for 'Initialize this repository with a README', which is currently unchecked. At the bottom, there are dropdown menus for adding a '.gitignore' file (set to 'None') and a license (set to 'None'). A green 'Create repository' button is at the bottom of the form.

Search GitHub

Explore Gist Blog Help

myOwnUserID

Owner: myOwnUserID / Repository name: hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **yolo-octo-ironman**.

Description (optional): This is my first repository

☒ Public: Anyone can see this repository. You choose who can commit.

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

☐ Initialize this repository with a README: This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

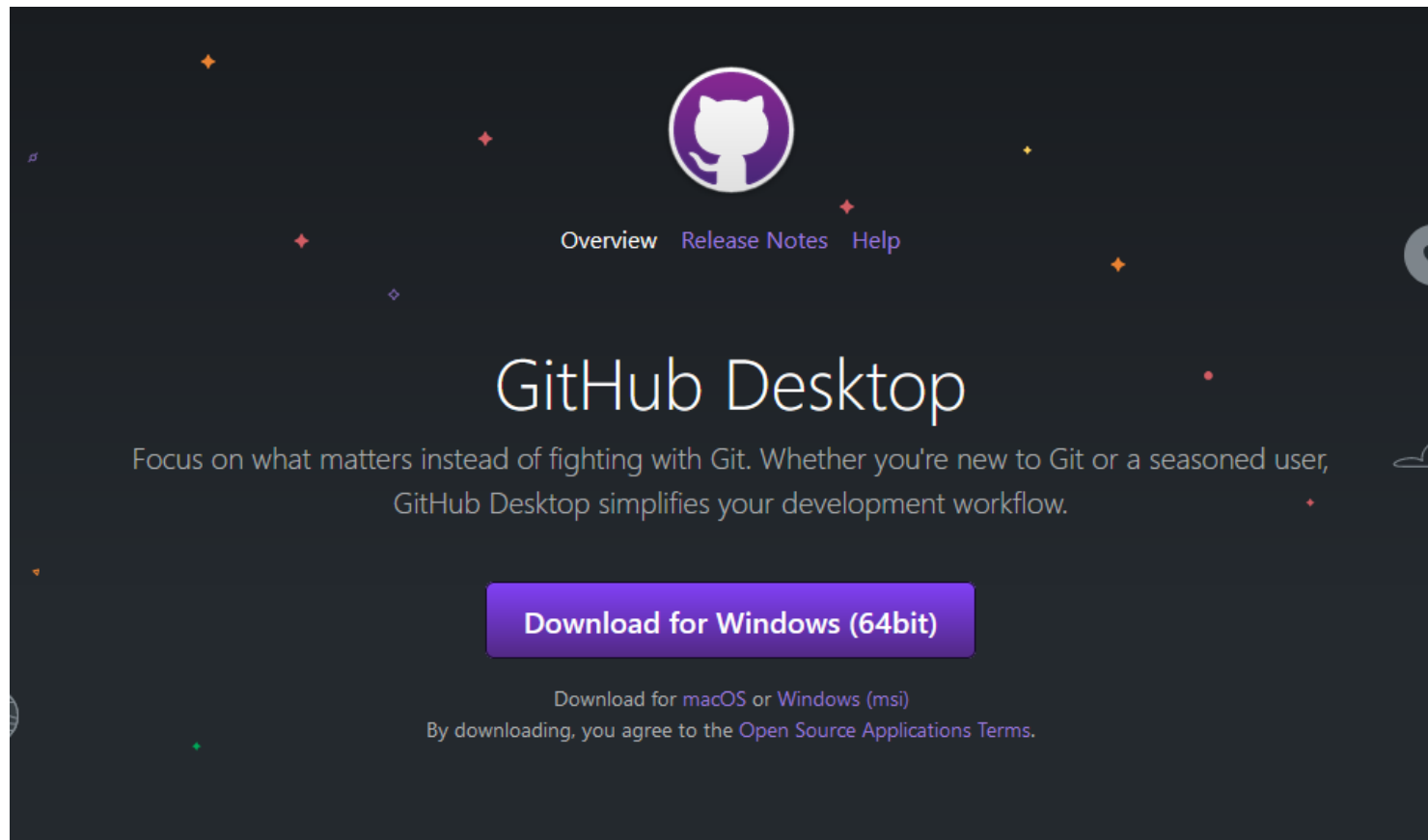
Add .gitignore: None | Add a license: None ⓘ

Create repository

Installing the GitHub Client (Desktop)

Download GitHub Desktop (Windows & Mac)

- Navigate to: <https://desktop.github.com/>
- Video Tutorial: Installing and using GitHub Desktop: <https://youtu.be/sObZ61W66GU>



Your First Commit

Make a Commit

- On **GitHub**, saved changes are called **commits**. A bunch of commits can read like the history of your project.
- Each commit has an associated **commit message**, which is a description explaining why a particular change was made.
- Thanks to these messages, you and others can read through commits and understand what you've done and why.