How to share code with...

Git / GitHub

Objectives

- Introduction to git
- Introduction to GitHub
- Make your own local repository and push it to GitHub
- Share the repo with the professor and TAs
- Use feature branches and pull requests to push modifications

Git vs GitHub

- Git
  - Version control system
  - Local
  - For managing your source code history
- GitHub
  - Hosting service
  - Cloud-based
  - For managing Git repositories.

GLOSSARY

- repository: A repository, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history.
- commit: a Git object, a snapshot of your entire repository compressed into a SHA
- branch: a lightweight movable pointer to a commit
- clone: a local version of a repository, including all commits and branches
- remote: a common repository on GitHub that all team members use to exchange their changes
- fork: a copy of a repository on GitHub owned by a different user
- pull request: a place to compare and discuss the differences introduced on a branch with reviews, comments, integrated tests, and more
- HEAD: representing your current working directory, the HEAD pointer can be moved to different branches, tags, or commits when using git checkout

Basic commands

git init: initializes a brand-new Git repository and begins tracking an existing directory. It adds a hidden subfolder within the existing directory that houses the internal data structure required for version control.

git clone: creates a local copy of a project that already exists remotely. The clone includes all the project's files, history, and branches.

git add: stages a change. Git tracks changes to a developer's codebase, but it's necessary to stage and take a snapshot of the changes to include them in the project's history. This command performs staging, the first part of that two-step process. Any changes that are staged will become a part of the next snapshot and a part of the project's history. Staging and committing separately gives developers complete control over the history of their project without changing how they code and work.

git commit: saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. Anything that's been staged with git add will become a part of the snapshot with git commit.

More commands

git status: shows the status of changes as untracked, modified, or staged.

git branch: shows the branches being worked on locally.

git merge: merges lines of development together. This command is typically used to combine changes made on two distinct branches. For example, a developer would merge when they want to combine changes from a feature branch into the main branch for deployment.

git pull: updates the local line of development with updates from its remote counterpart. Developers use this command if a teammate has made commits to a branch on a remote, and they would like to reflect those changes in their local environment.

git push: updates the remote repository with any commits made locally to a branch.

Intro to Git

- Git Handbook
  - https://guides.github.com/introduction/git-handbook/
- Git CheatSheet
  - https://github.github.com/training-kit/downloads/github-git-cheat-sheet/
- Git Branching
  - https://learngitbranching.js.org/
- Tutoriel GitHub
  - https://lab.github.com/githubtraining/introduction-to-github
- GitHub Backpack
  - https://education.github.com/pack

README.MD AND MARKDOWN

- README is the first file one should read when starting a new project. It's a set of useful information about a project and a kind of manual. It is the first file Github or any Git hosting site will show when someone opens your repository
  - https://blog.bitsrc.io/how-to-write-beautiful-and-meaningful-readme-md-for-your-next-project-897045e3f991
- And the .md extension comes from markdown which is a markup language for text formatting. Just like HTML it is a markup language to make our documents presentable.
  - https://www.markdownguide.org/cheat-sheet/

Your own repo

- Make sure git is installed locally (should also be available on your lab account)
- Create a new repository called
  - **seg3103_playground**
- Create and commit a README.md
  - https://gist.github.com/jxson/1784669

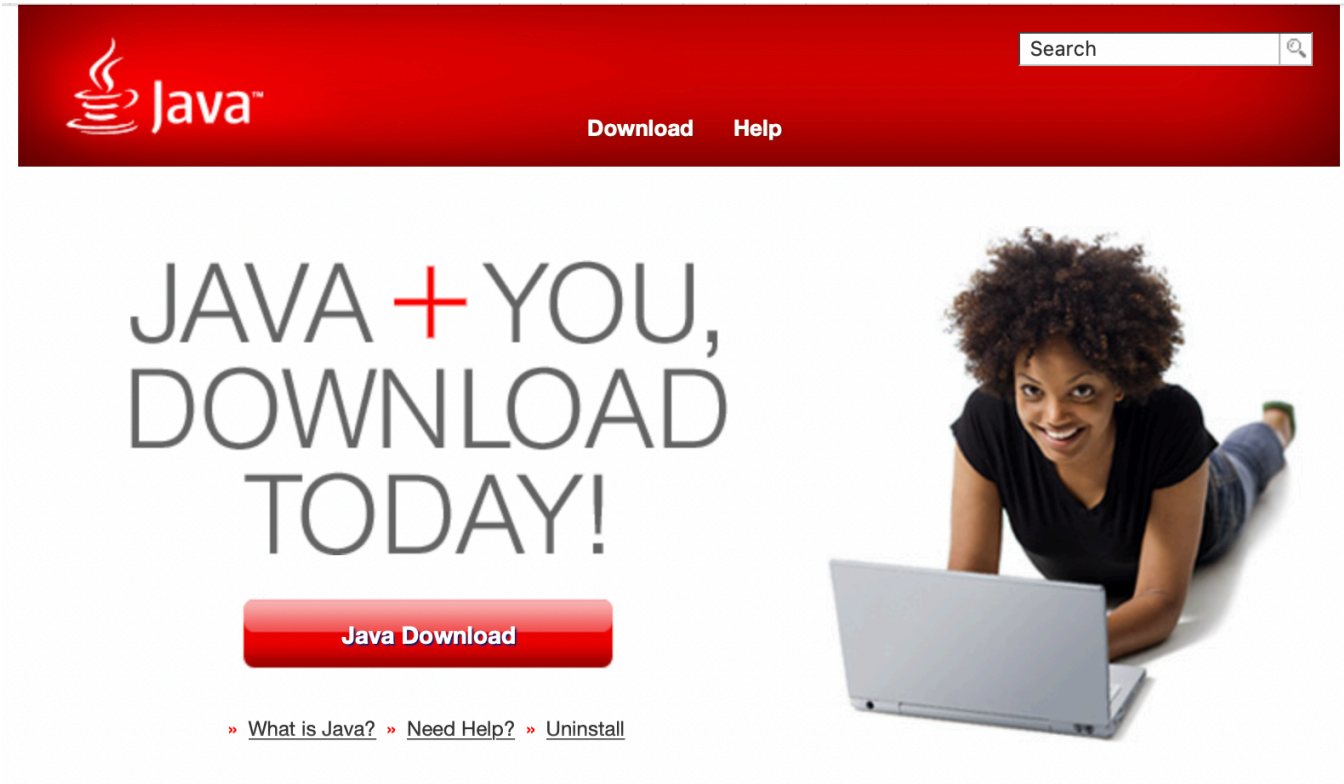
GITHUB

- Create an account with GitHub
  - https://guides.github.com/activities/hello-world/
- Push your repository to GitHub
  - **seg3103_playground**
- Update your README.md file
- Commit and push to GitHub

Other resources / articles

- **My Git Workflow** (Article)
  - https://blog.osteele.com/2008/05/my-git-workflow/
- **git rebase -i HEAD~25** (Video)
  - https://www.youtube.com/watch?v=V53cpDt2dr0
- **GitHub Actions** (Article)
  - https://www.bytesized.xyz/github-actions-tutorial
- **How to undo (almost) anything with Git** (Article)
  - https://github.blog/2015-06-08-how-to-undo-almost-anything-with-git/

ENV. JAVA



https://java.com/

```
08:14 /tmp/newmath_java $ ./bin/run
Newmath (type 'exit' to exit program)
Numerator: 10
Demoninator: 5
10 / 5 = 2
Numerator: 20
Demoninator: 3
20 / 3 = 6
Numerator: exit
```

Env JUnit

https://junit.org/junit5/

```
08:14 /tmp/newmath_java $ ./bin/test

Thanks for using JUnit! Support its development at https://junit.org/sponsoring


├─ JUnit Jupiter ✔
│  └─ NewmathTest ✔
│     ├─ div_ok() ✔
│     └─ div_by_zero() ✔
└─ JUnit Vintage ✔

Test run finished after 31 ms
[         3 containers found      ]
[         0 containers skipped    ]
[         3 containers started    ]
[         0 containers aborted    ]
[         3 containers successful ]
[         0 containers failed     ]
[         2 tests found           ]
[         0 tests skipped         ]
[         2 tests started         ]
[         0 tests aborted         ]
[         2 tests successful      ]
[         0 tests failed          ]
```

Env Elixir (+ ExUnit)

elixir

HOME INSTALL LEARNING CASES DEVELOPMENT GUIDES DOCS BLOG

Elixir is a dynamic, functional language for building scalable and maintainable applications.

Elixir leverages the Erlang VM, known for running low-latency, distributed, and fault-tolerant systems. Elixir is successfully used in web development, embedded software, data ingestion, and multimedia processing, across a wide range of industries. Here is a peek:

```
iex> "Elixir" |> String.graphemes() |> Enum.frequencies()
%{"E" => 1, "i" => 2, "l" => 1, "r" => 1, "x" => 1}
```

Check our getting started guide and our learning page to begin your journey with Elixir. Or keep scrolling for an overview of the platform, language, and tools.

News: Elixir v1.11 released

Search...

OFFICIAL CHANNELS

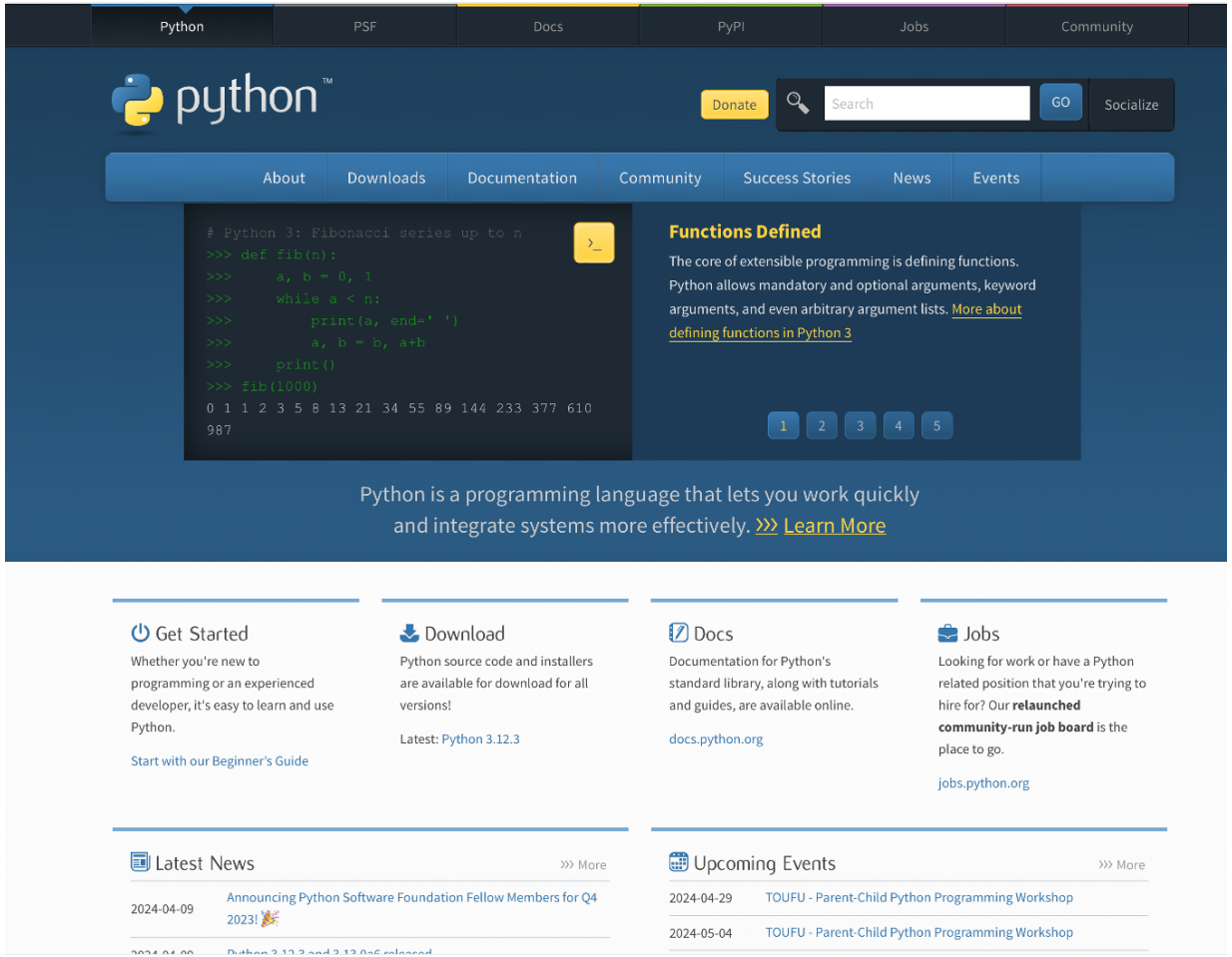
- Source code & Issues tracker
- #elixir-lang on freenode IRC
- @elixirlang on Twitter

Watch the Elixir mini-documentary!

```
08:36 /tmp/newmath_ex $ ./bin/run
Erlang/OTP 23 [erts-11.1.7] [source] [64-bit] [smp:8

Interactive Elixir (1.11.4) - press Ctrl+C to exit (
iex(1)> NewmathEx.div(5,2)
{:ok, 2.5}
iex(2)> NewmathEx.div(5,0)
{:error, "Cannot divide by zero"}
iex(3)>
```

```
08:37 /tmp/newmath_ex $ ./bin/test
...

Finished in 0.03 seconds
1 doctest, 2 tests, 0 failures

Randomized with seed 617712
```

ENV PYTHON (+ PYUNIT)



```
12:31 /tmp/newmath_py $ ./bin/run
>>> newmath.div(6,3)
2.0
>>> newmath.div(6,0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/private/tmp/newmath_py/src/newmath.py", line 3, in div
    raise ZeroDivisionError("Division by zero is not allowed.")
ZeroDivisionError: Division by zero is not allowed.
>>>
```

```
12:31 /tmp/newmath_py $ ./bin/test
..
----------------------------------------------------------------------
Ran 2 tests in 0.000s

OK
```

SUBMISSION

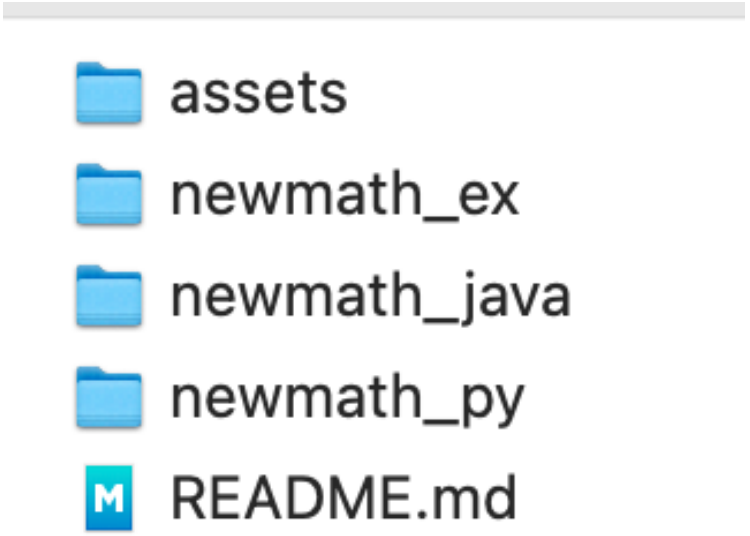
- Git + GitHub repo
- Share your repository with the teacher and TA (s)
- Java / Junit code
- Elixir / ExUnit code
- Python / PyUnit code
- README.md
  - Instructions for running your code
  - Screenshots to prove you can run the code

Both steps are required:

1. Share your repository with the teacher and your appropriate TA:
   - aforward@hey.com
   - gvira@uottawa.ca
   - mdemc071@uottawa.ca
   - aghad051@uottawa.ca
2. Upload these in Brightspace as text:
   - Your name, student number, and lab title
   - Link to your GitHub repo

REPOSITORY STRUCTURE

- assets: screenshots
- newmath_ex: Elixir code
- newmath_java: Java code
- newmath_py: Python code
- README.md: Instructions for running your code



Markging Criteria

- 30% Repo structure
- 40% README.md
- 30% Java, Elixir, Python env.