git

Goal: Familiarize yourself with the basic git commands (add, commit, pull, push). Learn to write commit messages and inspect the commit history.

Creating a repo and uploading code

Create a Github account if you don't have one. Also, create a github pages repository. This is a normal github repository, but whatever html files you have around are served by github. For example if my username is **anyoneai** and the repository name is **anyoneai.github.io**, then my repository url is **http://github.com/anyoneai/anyoneai.github.io**, and **anyone on the internet** can now access **anyoneai.github.io**.

Clone the repository, add the files of personal webpage you developed earlier, and commit and push those changes to the github remote repository. Use a commit comment describing your initial work. Wait a bit (the server takes ~10 min to update) and you should be able to access your website at <username>.github.io/index.html or simply <username>.github.io.

Note that your website may not load the CSS correctly in the server; if that happens, check that the link to the css file is relative to the website's root.

Working on a partner's repository

Afterwards, you must choose a partner from your current team and give her/him access to your repository as a collaborator. Now, each of you will work on the other's site, adding a new feature.

Clone your partner's repository. Check that the webpage displays correctly locally.

New contact form

The new feature consists of a contact form to message your partner for jobs. The body must also include, at the bottom, a contact form with the fields **email** and **message**, and a submit button. Note that the form will not really send emails because it won't be connected to a server; you'll just be coding the interface. The style of the form must match that of the rest of the website.

Commit and push the changes, including a relevant commit message, to your partner's repo, and browse the website to check that they display correctly in the server as well.

Improve design

Think of a way to improve the design of your partner's website and implement it. Choose something simple, such as adding a padding to some elements o a bottom border to divide sections.

Commit and push the changes, including a relevant commit message, to your partner's repo, and browse the website to check that they display correctly in the server as well.

Reviewing your new website

After your partner has finished with **your** website, pull **his** changes (now in the remote github repo) back to your computer, review the added code and compare it to the one you developed for your partner. Use the command **git log** to see when your partner commits.