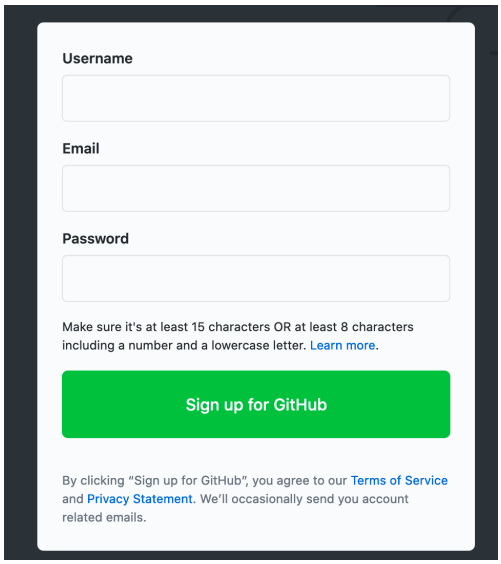
F28WP Web Programming
Lab Sheet 1
Installation and Configuration

Lab activities to help you prepare the development environment and configure your GitHub for the coursework.

1.1 GitHub

Git is an open source version control system. GitHub is an online system based on Git that provides many additional functionalities, especially for managing projects.
Connect to https://docs.github.com/en/github/getting-started-with-github.
This web site provides documentation on GitHub. Make sure you understand the main concepts (repository, organization, master, branch, issue…) and operations (fork, clone, commit, merge).

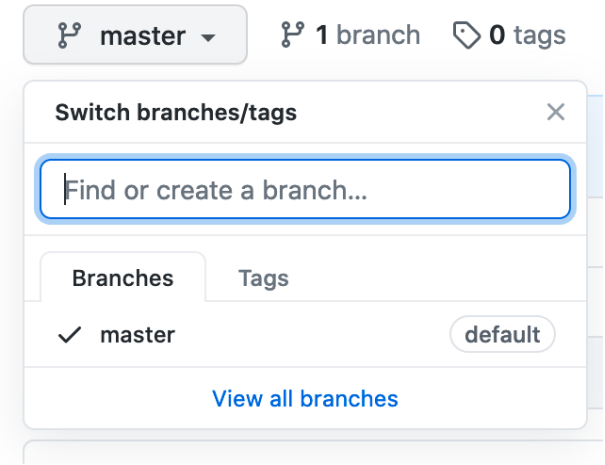
Sign up to GitHub by creating an account using your Heriot Watt University email address.



Create a public repository named F28WP-lab1. Edit the readme file to explain the content of your repository.

Create a new file in your repository (file1.txt). Note that you are asked to commit the file to the master branch. You need to provide a commit message and a description of the change you just made to the repository.

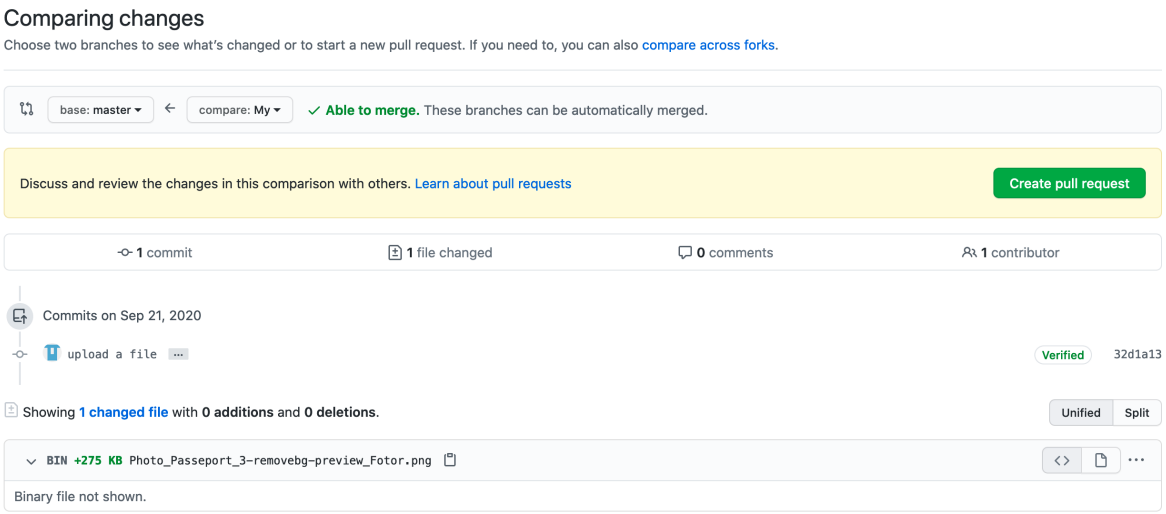
Create a branch named "myFirstBranch". A branch is a copy of your repository. It is used to make changes, try things… Click on master and type in the name



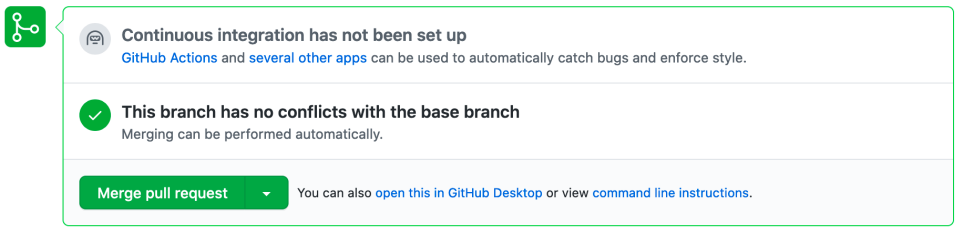
Upload any files from your desktop to the new branch of repository using drag and drop. The files are now in the branch and not in the master (which is the main copy of your repository).

Now, you are going to merge your branch to the master. That means that any change you made to the branch will be applied to the master. Click on “Pull requests” and and then “New pull request”.

Compare myFirstBranch to master and view the changes and create pull request.



Add a message and click “create pull request”. On the next page, click on “Merge pull request” and confirm.



Click on “code” and select master to check the changes.

You can list or create “Issues” where you would like to discuss matters with your group members if any. This might consist in pointing to bugs, or asking for change…

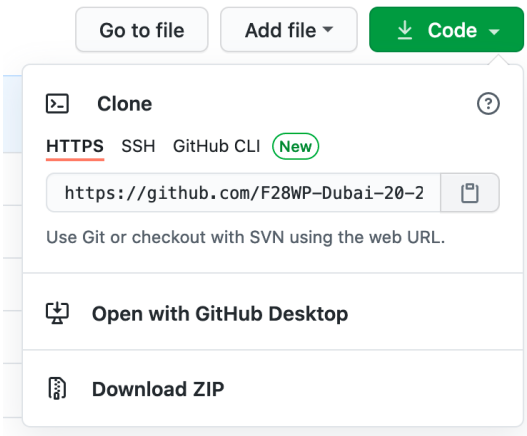
1.2 GitHub Desktop

The standard way of working with GitHub consists in committing changes to a repository from your desktop or laptop computer.

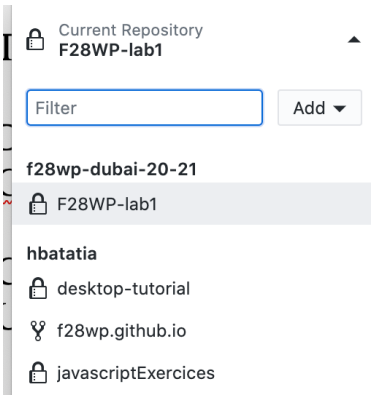
The easier way to do this is by using GitHub desktop which also provides Git commands.

Download and install GitHub desktop from https://desktop.github.com/ or by searching Github desktop on google.

Clone your repository to your local computer using GitHub desktop. First, get the repository URL by clicking on “code”, the drop down “code” menu, and copy the URL.



Paste the URL and click add.



You can also simply click “Open with GitHub Desktop” from GitHub.com. You have to select a directory (usually called working directory) on your local machine to store your cloned repository. Visit you file explorer to check that all files have been downloaded from GitHub to the working directory.

1.3 Editing code

To work with your html, css, javascript, PhP code (or any other language), you need an Integrated Development Environment (IDE). There are many of such environment online and on desktop.

Try using JSFiddle.net, jsbin.com, and codepen.io. The three offer three panes (or areas) to edit your HTML, CSS, and JavaScript code. You also can view results in a dedicated area and visualize a console space (where messages can be logged).

Type and run simple javascript code like

```
console.log (‘’ Hello World “);
```

or

```
window.alert(“Hello World”);
```

or

```
const person = prompt("Please enter your name", "your name");
if (person == null || person == "") {
  txt = "User cancelled the prompt.";
} else {
  txt = "Hello " + person + "! How are you today?";
}
window.alert(txt);
```

or

```
var myFunkyAlert = "The funky alert";
(function(fun) {
  alert(fun);
})(myFunkyAlert);
```

For the lab tutorials and exercises, you are advised to sign up to JSFiddle.

Explore public fiddles by visiting the public dashboard http://jsfiddle.net/user/forkids/fiddles.

1.4 Visual studio code

For the coursework, you can use Visual Studio Code that comes with extensions for many languages and frameworks.

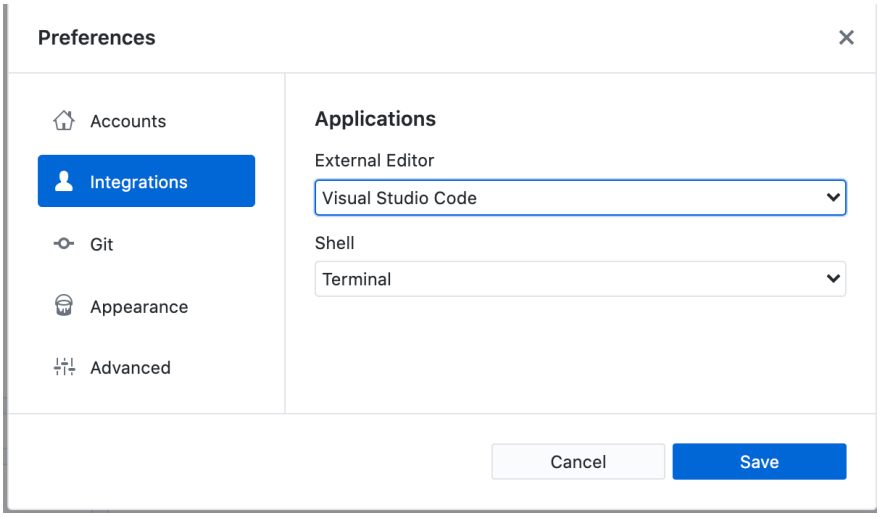
Download and install Visual Studio code from https://code.visualstudio.com/download. Explorer and install the following extensions for working with javascript :

- GitHub
- JavaScript (ES6 Code Snippets)
- ESLint
- Debugger for **Chrome**

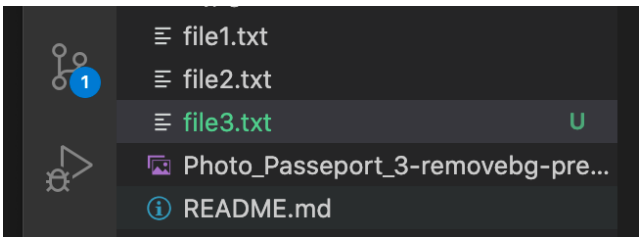
- Quokka
- Prettier
- Import **Cost**
- Path **Intellisense**
- View Node Package

You will install many others as you need them.

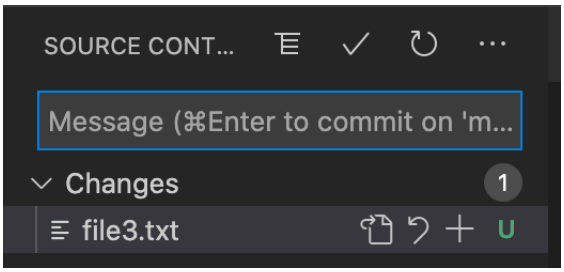
On GitHub Desktop, set Visual studio code as your default editor



While having your current repository set to F2WP-lab1, click “open in visual studio code”. You can now see your files in the IDE. Create a new text file (file3.txt), type and save some text. Note the change to the environment showing that one file has changed.

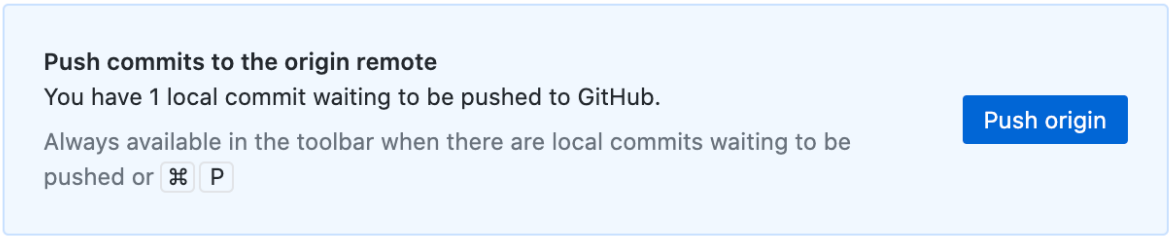


Click on the source control icon ().



Stage the changes (+), write a commit message and click the commit icon (√). Using the file explorer, check that your file is withing the working directory.

Switch back to GitHub desktop, you will see a message indicating a change in your working directory



You can push the changes to origin (that means to the online GitHub directory). Switch back to GitHub.com and check that your file3.txt has been uploaded with the commit message. Note that you have cloned the master branch. This is not a good practice. For groupwork, you should clone your branch and commit any changes to that branch before requesting to merge.

Under Visual Studio code, add a file called app.js

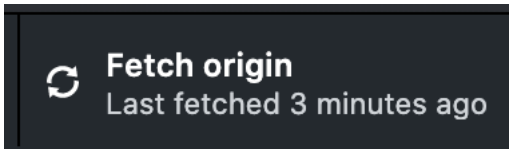
```
const num = 100;
let square = (function(a) {
    return a * a;
})(num);
console.log(`Square(${num})=` + square);
```

Run the code and inspect result on the console. Commit changes and push origin.

In GitHub.com, edit app.js (online) and add a comment line

```
//this is an anonymous function called immediatly when declared
```

Save and return to GitHub Desktop. Click on “Fetch origin” to download the new version of app.js



This is the method you are going to use to “download” what the other group members working on the same project have changed to the repository.

1.5 Coursework

Start forming a group with your colleagues (discuss with other students, ask if they need a group member, set up chat, post messages on the discussion board under vision).
When the group is formed and named, **one** of the group members creates an organization, with the name F28WP-Dubai-<**group**> , where <group> is your group name. She /he invites the other members of the group by adding their names to the organization as owners. Members receive an email to confirm joining the organization, they click on the link to confirm. One group member creates a repository for the coursework under the organization

(the repository must be public). Group members can create branches (click the drop-down menu beside *master* and type a name for your branch).

Members **clone** their branches to their *local git repository* (on their machine).

Members **commit** the changes (new files, changes...) to their local repository. They **push** regularly to the GitHub repository.

When merging a branch to the master becomes necessary, member create a **pull request**. They can then **merge request**.