# Practical: HTML and CSS

## Aims

* Familiarisation with basic web functionality: HTML and CSS
* Publishing web sites with gitpages

## Tasks

### Learn about HTML

If you are not already familiar with HTML, read up about it on MDN. We are not going to be looking in detail at HTML5 features such as HTML5 forms, canvas, multimedia etc but feel free to have a look.

### Learn about CSS

Similarly you need familiarity with CSS on MDN including selectors and combinators.

## Investigate, Make a question

Now that you have some familiarity with HTML and CSS, have a look at the file [styling-lists.html](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Styling_lists) which comes from the MDN tutorial.

First, fork this repository https://github.com/stevenaeola/proglabs\_js on github and clone your forked version to your local machine. Set the upstream remote to be my original repository

Within your local copy make some changes to the styling-lists.html file, and/or add one or more new files within the html\_css directory. Perhaps you can investigate until you find a change that has a useful, interesting, beautiful or surprising result. You can see the result of your change by opening the local file with a web browser

Commit and push your change to your forked repo

Make a pull request (PR) from your update fork to the main branch of my repo

## Write a document

Write a one-page HTML document about a topic that interests you, this could be a hobby or a general topic. Use this as a way to tell me more about yourself. Over the course of the following practicals, we will build upon this website to explore client-side functionalities.

You should include

A heading which includes text on a coloured background

A list (play around with the attributes of lists, make them look different,

Some hyperlinks

An image

A table

A consistent, but non-default font

All style information in a separate CSS file You can view, test and develop your page by accessing the page from within your file browser: it will launch a web browser and access via the file: scheme rather than the http: scheme that you use for web pages hosted on the internet (see the list of URI schemes if you're interested to find out more).

## Publish your document

Publish your web page by using gitpages. To do this you will need to create a repository with a specific name, add your files to it and then push them to github. For full instructions see how to work with GitHub pages, in particular creating a GitHub pages site - but call your base file index.html rather than index.md.

The name of the repository you choose is important: it should be <user>.github.io where <user> is your GitHub username. To get your files associated with this repository you can either:

git clone the repository into your local machine then move your files into the (empty) clone; or

git init within the directory holding your files, then set the GitHub repo as the remote and git pull.

In either case you will need to git commit your changes and git push them to the remote repository to get them to appear on the gitpages site.

## Use a framework

Make a copy of your HTML page and change it so that it uses either bootstrap or an alternative. If you've used one of them before then try another one. You should have to change very little of the HTML content, only the CSS classes referred to. Use templates that are provided in the documentation where relevant.

## Make it mobile-friendly

If you have time, make your page responsive for mobiles, by using

## A navbar (collapsible)

a grid layout for content Look in the documentation for examples of use.