GitHub Practise and Setting up of Gitpod

**Step 1| Sign up an account at Github**

Go to <http://www.github.com> and sign up for an account.

**Step 2| Install the Gitpod extension for your browser**

Google “gitpod extension <browser name>”, where *<browser name>* is the name of the browser which you are using. Then add the extension to your browser.

**Step 2B| Using Git on your computer**

1. Download Git from <https://git-scm.com/downloads>
2. Install Git, just stick to all the defaults

**Step 3A| Surf to the following URL:**

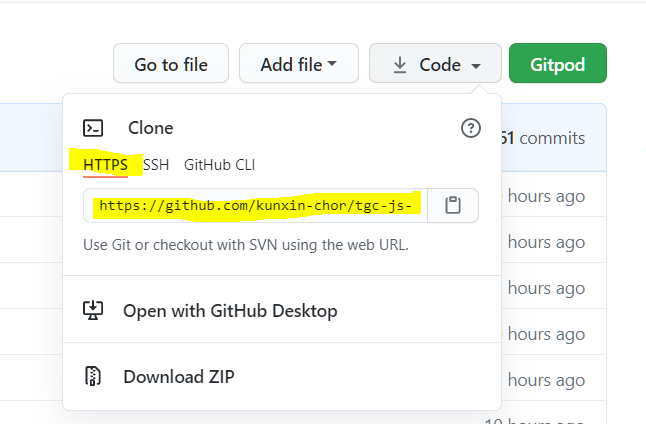
CI has provided us with a *github template* which we can create projects from. We will make a copy of it on our Github account.

First, point your browser to the following URL: <https://github.com/Code-Institute-Org/gitpod-full-template>

Second, click on the “Copy Template” green button near the upper right corner. You will then be asked for the name of your new Git repository.

Finally, after the resulting page has loaded, click on the green button that says “Gitpod”.

**Step 3B| Clone your Git inside your local host machine**

1. Open the terminal in a folder where you want to have your git repository cloned to. The git repository will be placed inside a folder inside the current folder.
2. Click on the [Code] button
3. Make sure *HTTPS* has the red underline
4. Copy the URL highlighted below:  
     
   
5. Type in git clone <copied url> in the terminal
6. Open the newly created folder in your IDE.

**Step 4| Create a simple website**

Create a file name “index.html”, and create a header, and three paragraphs of random text (you can use lorem ispum if you wish).

**Step 5| Perform your first commit**

At the terminal, enter the following commands:

|  |
| --- |
| git add . git commit -m "Initial commit" git push |

**Explanation:**

* *git add .*  : **stage** all modified file
* *git commit -m "commit message"*  : Commit add **staged** files with the given commit message
* *git push :* Push all commits to the online repo hosted at Github.

## COMMITTING AND PUSHING NEW CHANGES

What you have done so far is your initial commit. Now we explore what to do when you make changes to your project.

**Step 1| Upload an image to your GitPod Project**

Drag the image (or use the file upload feature) into GitPod.

**Step 2| Display the image using HTML and the <img> tag**

**Step 3| In the terminal, view files that have been changed**

Type in the following command and you should see files that are new and that have been modified.

|  |
| --- |
| git status |

**Step 4| Stage the new files**

In the terminal, type:

|  |
| --- |
| git add . |

**Step 5| Commit the files**

Use:

|  |
| --- |
| git commit -m "Added hero image in index.html" |

**Step 6| Create a new HTML file named page2.html**

Add in some perfunctory tags and content into *page2.html*

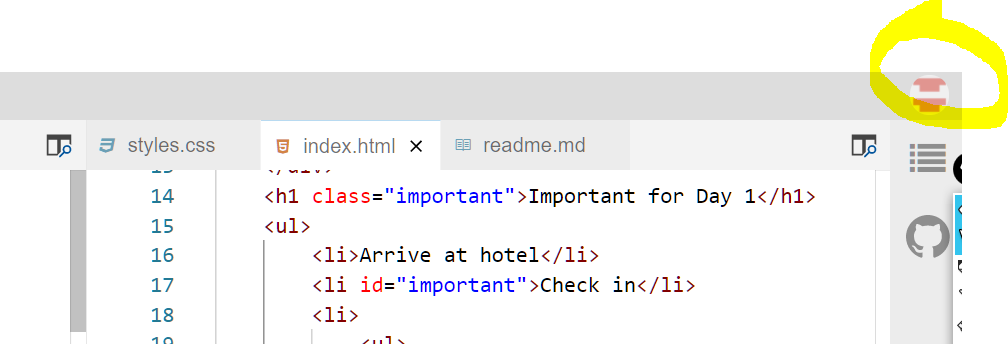
**Step 7| Repeat step 3 and step 4 again**

You can commit multiple times without pushing to your remote Github.

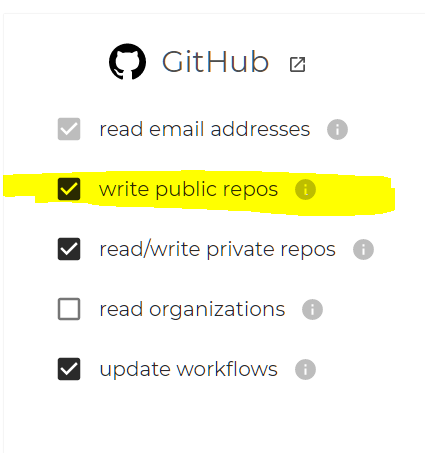
*For what it's worth, you can stage multiple times without committing*

**Step 8A| [GITPOD USERS] Grant Gitpod the rights to push**

1. Click on your avatar icon in the upper right corner



1. Select *Open* *Access Control*
2. Make sure *Write Public Repo* is checked (the rest aren't necessary).



**Step 8| Make a push**

Type the following in the terminal:

|  |
| --- |
| git push |

## BRANCHING

Your repo can consist of many branches; each branch is a "copy" of your project. You start with the *master* branch by default.

We branch whenever we want to work on a new feature of our project. It creates a virtual copy of the current branch that we are on.

**Step 1| Create a new branch named *with-nav-bar***

|  |
| --- |
| git checkout -b with-nav-bar |

**Explanation:**

* *checkout:* Switch to a branch
* *-b* ***branch name***: create a branch with the name ***branch name*** before switching

**Step 2| Check your current branch**

|  |
| --- |
| git branch |

**Step 3| Create a horizontal nav bar**

Use a mixture of *<ul>* and *<li>*, and Flexbox to quickly create a nav bar.

**Step 4| Stage and commit**

With the following commands in the terminal:

|  |
| --- |
| git add . git commit -m "Added navbar basics" |

**Step 5| Push to remote Github**

|  |
| --- |
| git push -u origin with-nav-bar |

The **-u** means set the *upstream* of the current branch to the remote Github.

From now on, you just need to do a git push

## 

## MERGING BRANCHES

**Step 1| Go back to the master branch**

We can switch branches with the *checkout* command:

|  |
| --- |
| git checkout master |

**Step 2| Perform the merge**

The syntax for merging is *git merge <from-where>*.

Replace *<from-where>* with the name of the branch which you want to merge from.

So in our case, we will do:

|  |
| --- |
| git merge with-nav-bar |