

MM1: Web Design Fundamentals

Practical 1.10: GitHub Pages

John Jennings

Academic Year 2016-2017

Contents

| | |
|--|----------|
| Introduction | 1 |
| Tasks | 1 |
| Task 1: Create a GitHub Account | 1 |
| Task 2: Create Your GitHub Repository | 1 |
| Task 3: Creating an Orphan Branch | 2 |
| Task 4: Adding Your Website Files | 2 |
| Task 5: Pushing Your Changes to GitHub | 2 |
| Task 6: Making changes and updateing your site | 2 |
| Task 7: Using a Custom Domain | 3 |

Introduction

So you want to launch a simple website but don't want to have to go through the dull process of setting up yet another hosting package. There is an easier solution. If you just want to launch a simple static website you can use GitHub Pages to host your site for free.

Tasks

Task 1: Create a GitHub Account

- Goto <http://www.github.com> and follow the process of creating a new account
- Use your Knumber as your username
- Have a look at the following optional resources to find more about Git:
 - <https://backlogtool.com/git-guide/en/>
 - <http://rogerdudler.github.io/git-guide/>
 - <https://www.lynda.com/Git-tutorials/Git-Essential-Training/100222-2.html>

Task 2: Create Your GitHub Repository

The files that make up your website will need to be stored within a GitHub repository.

- Ensure you have logged into Github.com and then goto <https://github.com/new>
- Create a repository called **WebDevelopment**
- Now open up terminal in Mac or command prompt on Windows and type the followign commands to firstly change directory to the **Desktop** and then copy or clone this new Github repository (empty at the moment) to your computer. Once you got your local copy, move into the project folder using the cd command.

```
cd
cd Desktop
git clone https://github.com/K00000000/WebDevelopment.git
cd WebDevelopment
```

Note: Make sure that you change the clone URL to the URL of your GitHub repo. This can be found on the main project page in Github.

Task 3: Creating an Orphan Branch

Now you need to create a new orphan branch within your repo that will hold all of your website files.

This new branch should be called gh-pages.

```
git checkout --orphan gh-pages
```

Task 4: Adding Your Website Files

Now that your repo has been properly setup it's time to add all of the HTML, CSS and JavaScript files that make up your website.

Using **Finder** or **Windows Explorer** copy the contents of your local root folder into the **WebDevelopment** repository. This Repository is now your local root folder and should contain the css, img, js folders and all your html files.

Once you have added the local root folder contents to your repo you need to commit the changes. To do this you can use the following commands.

```
git add --all
git commit -m "Adding pages"
```

Task 5: Pushing Your Changes to GitHub

Okay so you've got all your files where they need to be. The only thing left to do now is to push the new gh-pages branch up to GitHub. You do this using the git push command.

```
git push origin gh-pages
```

That's it! Your website should now be available at <http://k0000000.github.io/WebDevelopment/>.

Task 6: Making changes and updateing your site

From now on you can make changes to your site as required by editing / updating the files in your new Local Root Folder (e.g. **WebDevelopment**) and then executing the following commands in the terminal from the **WebDevelopment** folder :

```
git add --all
git commit -m "Add a messge here that describes what you added or changed"
git push origin gh-pages
```

Task 7: Using a Custom Domain

The last thing I want to cover in this practical is how you can use your own domain name with your new GitHub-hosted website.

First you will need to create a new file in your GitHub repo called CNAME that contains the domain name (or subdomain) that you wish to use. This file should be placed in the gh-pages branch if you are using project-pages (as we have been in this post). If you are using user-pages the file should be placed in the master branch.

Your CNAME file might look like the following:

```
jennin.gs
```

Next you will need to update the DNS records for your domain name. This is usually done through a control panel provided by your domain registrar.

If you want to use a root domain (such as jennin.gs) for your website you will need to setup a new A record that points to the IP address:

192.30.252.153 or 192.30.252.154

If you are using a subdomain (such as blog.jennin.gs) it's best to create a new CNAME record that points to your GitHub user subdomain (**username**.github.io). This is so that the DNS will be automatically adjusted if the servers IP address changes on GitHub.

Note: For information about the difference between CNAME and A records check out this video.

It may take a little while for your DNS changes to take effect. This is usually no more than a few hours. Once the changes have gone through, you should be able to access your new website from your custom domain name.