Build your brand

Start Adding Code to GitHub

Visit our website

Introduction

WELCOME TO THE FIRST BUILD YOUR BRAND TASK!

This is the first in a series of “Build your Brand” tasks that will be continued in level 3 of this Bootcamp. These tasks focus on helping you to showcase your newly acquired development skills to peers, potential clients and employers. In this task, you will push some of the code that you have written to GitHub. Your GitHub repository is a place where you can share some code that you have written that demonstrate your skills. This will become an important component of your developer portfolio.

BUILDING YOUR PROFESSIONAL BRAND

Personal branding is a way of reinforcing who you are and what you stand for in your career and life

-@JosephPLiu ***http://www.forbes.com/sites/josephliu/2018/04/30/personal-brand-work/***

Professional branding is about the reputation that you build as a professional. It is how others (including your peers, colleagues, clients and potential employers) perceive you. Your professional brand lets others know what your special skills are, what services you provide and what values you esteem.

Whether you like it or not, you will be branded by what you do or don’t do. For example, a potential employer may take a dim few of a person in the software development field who doesn’t make use of online services like GitHub and LinkedIn, which are commonly used in this industry. Otherwise, someone may question a software developer’s passion for their field if they aren’t using tools like GitHub. As **this article** puts it, “In today's digital world, building your personal brand isn't a luxury, it's a necessity. And if you don't craft your personal brand by design, it'll be constructed by default.” This is probably truer in the software development industry than in any other field.

Many tools help build a professional brand online. To showcase your skills as a developer, few are more important than Github.

WHAT IS GITHUB?

As you know, Git is the foundation of many services that work on version control. The most popular and widely used of them all is GitHub. GitHub is an online Git repository hosting service. GitHub offers all of the functionality of Git and a lot more. While Git is a command-line tool, GitHub provides a Web-based graphical interface. It provides access control and many features that assist with collaboration, such as wikis and basic task management tools for all projects.

GitHub is not just a project-hosting service, it is also a large social networking site for developers and programmers. Each user on GitHub has a profile, showing their past work and contributions that they have made to other projects. GitHub allows users to follow each other, subscribe to updates for projects, like them by giving them a star rating, etc.

Each project hosted on GitHub will have its own repository. Anyone can sign up for an account on GitHub and create their own repositories. They can then invite other GitHub users to collaborate on their project. You can even host websites for free directly from your repository!

GITHUB AND YOUR DEVELOPER PORTFOLIO

As repeatedly stated, a **developer portfolio** (a collection of online programs that you have developed) allows you to demonstrate your skills rather than just telling people about them.

GitHub provides one of the most industry-recognised ways of sharing your code with others, including peers, prospective employers or clients. A well organised and documented GitHub repository can serve as a core component of a developer portfolio.

Even before seeing your work, prospective employers may also be impressed with the fact that you have experience in working with Git and Github.

README.MD FILES

When you add your code to GitHub, you can and should create README files. A README file is usually the first file that anyone interested in your code will look at. This file should describe your code. It should tell the reader what the project does,

why the project is useful, who maintains and contributes to the project and how a user can get your code to work.

A README file is essential for all software projects. However, for you as a data scientist, README files are especially important. Data scientists focus on writing code that analyses data to solve a problem, answer questions or find trends. However, for an interested party (like a prospective employer or client), it may not be easy to see what the purpose or value of your code is when they first see it. README files address this. It should let the reader know what the code does and why it is important. Therefore, learning to write clear, easy-to-read and detailed README files is an essential skill.

According to **this GitHub guide**, README files should contain the following:

- The project name.
- A clear, short, and to the point description of your project. Describe the importance of your project, and what it does.
- A table of Contents to allow other people to quickly navigate especially long or detailed READMEs.
- An installation section which tells other users how to install your project locally.
- A usage section that instructs others on how to use your project after they've installed it. Include screenshots of your project in action.
- A section for credits which highlights and links to the authors of your project if the project has been created by more than one person.

README files have a .md extension. "md" stands for Markdown. Markdown is a syntax that lets you style text. If you write text in a program like MS Word, you usually use the toolbar to select appropriate options to style your text (e.g. make certain text bold, underlined or formatted in another way). When creating Markdown files, you style your text using keywords and characters instead. For example, if you want to italicize text, you would surround the text with asterisks. For example, `In this paragraph *this text would be in italics*.`

Below is a summary of Markdown syntax taken from this **GitHub Guide**:

Headers	# This is the biggest heading you get. It is usually used for the title of a doc. ## This is a slightly smaller heading. ###### This is the smallest heading that you get.
Emphasis	*This text will be italic* _This will also be italic_ **This text will be bold**

	__This will also be bold__ _You **can** combine them_

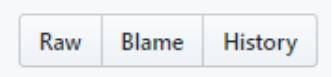
Unordered Lists	* Item 1
	* Item 2
	  * Item 2a
	  * Item 2b

Ordered Lists	1. Item 1
	1. Item 2
	1. Item 3
	  1. Item 3a
	  1. Item 3b

Images	![GitHub Logo](/images/logo.png) Format: ![Alt Text](url)
Links	http://github.com - automatic! [GitHub](http://github.com)
Blockquotes	As Kanye West said: > We're living the future so > the present is our past.

Extra resource

*To see an example of a README file, go **here**. Notice how the README file is rendered in the browser. Now click on "Raw" to see the Markdown for this file.*

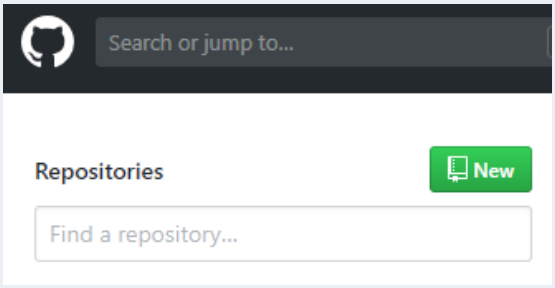


*For more information about Markdown, see the Markdown cheatsheet (additional reading) provided by GitHub **here**.*

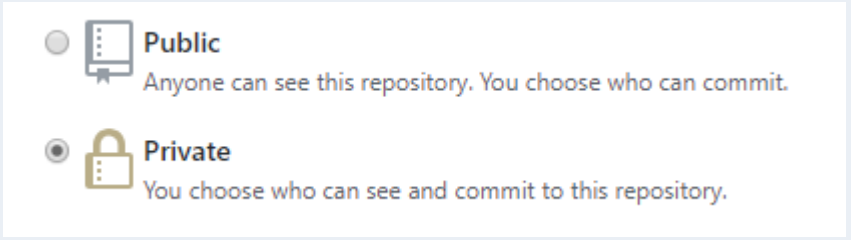
Compulsory Task 1

Follow these steps:

- Login to GitHub using the account you created in the previous task.
- Create a new repository by selecting the 'New' button as shown in the image below.



- Name the repository 'gitTask' and make sure that it is private.



- You now have two git repositories. The one you have been using in the previous compulsory tasks on your local machine and this remote repository that you have just made on GitHub. Push the repository on your local machine to the remote repository on GitHub by following these steps:
    - Open your terminal or command prompt and change directory (`cd`) to the folder task1_project created above.
    - **Add your remote repository** using the following command:
      `git remote add [shortname] [url]`
      E.g. `git remote add task4` **`https://github.com/HyperionDev/gitTask.git`**.

      Now you can use the short name (e.g. task4) on the command line in lieu of the whole URL. The URL will be indicated under the heading shown below once you have created your repository on GitHub.

      ...or push an existing repository from the command line

    - Push your local repository to your remote repository using the following command:
      `git push [remote-name] [branch-name]`
      E.g. `git push -u task4 master`

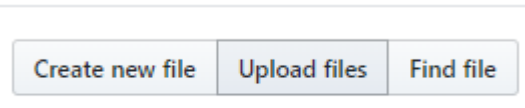
- Invite your mentor to be a collaborator to the repository you have created. Help **here**. Take a screenshot to show you have done this and paste it in this task's folder.

- Once your mentor has marked this task as complete (and not before!) you can delete the repository that you have created here since it doesn't store any meaningful application code. Help **here**.

Compulsory Task 2

Follow these steps:

- Push the code, data source and report for the last two Capstone Projects in this level to GitHub. If you don't like working with git, you could also just select the "Upload files" in your repository in GitHub button and drag and drop the files to this repository.



- Add detailed README files for each project that you have pushed to GitHub. In each README file, explain:
    - what the purpose of the project is,
    - what data is being analysed (add links to the datasets if appropriate),
    - how the data is being analysed (add links to .ipynb files if appropriate) and
    - what the main findings are (add links to report documents where appropriate).
- Add **students@hyperiondev.com** as a collaborator to all the repositories that you have created. Take a screenshot to show you have done this and paste it in this task's folder.

Optional Task

Follow these steps:

- You can use **Pandoc** to convert .docx documents to .md files. To do this:
    - Install Pandoc (Instructions **here**).
    - Copy and paste the .docx document that you would like to convert to a .md file into the folder where Pandoc is installed. E.g. copy eda.docx to this folder.
    - Open the command line interface and change directory (cd) to the folder where Pandoc is installed.

      E.g. `cd C:\Users\HyperionDev\AppData\Local\Pandoc`
    - Run the following command (type it into the command line interface) to convert the .docx file into a .md file: `pandoc -o name_of_output_file.md --extract-media=./ name_of_file_you_want_to_convert.docx`

      (E.g. `pandoc -o output.md --extract-media=./ eda.docx`)
    - Once this command has been executed, you should be able to see the .md file in the folder where pandoc is installed.
    - Push the .md file to GitHub.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.