

documentation/1-Introduction.md



1 - Introduction

This is an introduction to the UBC Red Cross website. If you want to get right to making changes and maintaining the website, skip this doc. Otherwise, I'll go over the basic tools that the website is built on, what they are, and how they're used.

Note that I'm not too experienced in web design or the technical aspects of a website either. These docs will cover what I know. If you have any questions, feel free to shoot me (Duncan) an email at duncan@wapta.ca, even if I'm no longer part of Red Cross. I'll be happy to help to see that the website gets the maintenance it needs!

Finally, I'm aware that this is going to be a significant amount of work to set up and maintain. If you decide that you need to drop this, I'll provide instructions for taking down the website.

Table of Contents

1. Overall Structure
2. Code
3. Git
4. GitHub
5. Porkbun
6. Getting Started

Overall Structure

This website is built on code, a mix of markdown, HTML, and more. It's a modified version of the Lab Website Template. This code is controlled using Git, a version control system. It is stored in GitHub, an online site, which also builds the website so that it can be displayed online. Finally, to give us a custom URL, we've bought a domain name from Porkbun which points to the Github website build.

If you understood none of that, no worries! I'll get into each part in the section below.

Code

The code includes everything visible on the website, including descriptions, text, and pictures. It also determines how the website is structured and formatted, and controls website things like how pages link to each other. To make changes to any of the above, you will need to edit it, usually by adding or deleting pictures to folders, or editing text files.

The code includes many files and folders in various programming languages and formats. It's a modified version of the Lab Website Template, a template for make websites that's available [for free online](#). It is very complicated, but luckily for you, making the changes you need to make only requires editing a few files, and does not require any programming knowledge.

Git

Git is a version control system, which means software that lets you go back to earlier “saves” of your work (among other things). Most code is “tracked” using Git, meaning when you make changes, you organize them using Git software so that if you screw up, you can go back to an earlier version.

Learn more about Git on [their website](#) if you want. You don't need to understand too much about it though, I'll show you how to use it in the following docs.

GitHub

GitHub is pretty much just Google Docs for code. It lets you store your code online, on the GitHub website so that others can see it and work on it. It has some fancy features too, like being able to build our website so it's visible online.

You may notice that in the *Code* section above, I linked you to the Lab Website Template, which is on GitHub. Our website code can also be found on [GitHub](#).

When our code is updated, GitHub runs a series of commands that turn that code into a website. You can visit this website at this [URL](#).

Porkbun

Porkbun is a website hosting service. We use it because having “github” in the URL from the section above looks messy, and won’t display on a Google search. We pay 12\$/year to reserve our website’s [current URL](#). Porkbun links this URL to the website on GitHub.

Getting Started

In the following docs, you can learn how to manage the website.

- 2 - Web Presence:
Manage how the website appears on the internet.
- 3 - Getting Software:
Set up the software tools you need to make changes to the website.
- 4 - Using Software:
Learn how to use the software tools to make changes and preview the website.
- 5 - Making Changes:
Learn where go to change different parts of the website.

The template upon which this website is based also comes with some very useful documentation. Learn more about the website [here](#).

Good luck, you’ve got this!