Web Fundamentals - Course Preparation 😊

Welcome to the Code First: Girls community

It's an exciting time to learn coding! Technology is now ubiquitous, and has become the most accessible toolbox for progress in our society. (And the more pragmatic amongst us might appreciate the resultant demand for technical talent.)

A lot of us have grown up seeing the effects of technological advancement, many of us only as consumers. If you're reading this, it's probably safe to assume that you understand why it's important to learn to code, and have some idea of the things you can do with it.

We're going to equip you with the **tools** to push you in the right direction in development, and we hope to stretch your imagination of the things you can create with code. It's time to take a peek under the hood, and understand the creation of the products and tools so prevalent in our lives today.

Coding is both a science and an art. A bit like cooking; a chef will follow a recipe, and each type of dish requires certain ingredients, but the end result is subjective and contains a little bit of your personality. Learning to cook might seem intimidating at first, but once you try a recipe to make your favourite food (or a part of it), it all becomes a little easier and much more fun.

There are probably a number of questions that might come to mind when you first start coding, we'll try to answer some of them right now; the necessary foundational knowledge you need to know.

1. What is coding? Is it different from programming? Is it hard? What do developers do? Are they nice?
2. What will we be learning?
3. What can I do with the skills I will learn in this course?
4. How should I learn coding? (This is synonymous with is there a "best" way to learn coding?)
5. Is there anything I can do to prepare for the course?

NOTE:

DEMOS will be in blue with a salmon background - Like this. TASKS will be in red with a pale-blue highlighted background - Like this.

Introduction to coding and the tech community

When we talk about "code", "source code", a collection of computer instructions written using some language (a programming language) that is **readable to humans**. When a bunch of code is put together in an *ordered* fashion for a *specific* purpose, we call it a program.

Programming is understood to be the **process** of making a concrete solution to an abstract problem for the computer to solve. For our purposes, we will only be at one programming language in this course - JavaScript. (One way of thinking of this - programming languages allow you to carry out mathematical equations in a program, and not all types of coding are meant for this purpose. *"Programming" is a subset of "coding".*)

Coding is, therefore, simply the art of speaking to computers. It's important to note that the **purpose of coding** is to **manipulate data**; which is the storage of information on your computer. If you look at coding this way, you'll realise that coding is really almost like learning a new language, except possibly easier ☺

Our predominant focus will be teaching you how to create your own landing page online. Therefore, we will be focusing on **markup** languages. More about this will be explained in Section 2.

The developer community is a vibrant, growing one. This article in Mashable probably explains it better:

From the Internet's earliest days, programmers would congregate in chat rooms and on forums to ask questions, swap code, and brag about their latest software masterpieces. The old stereotype of the anti-social, code-obsessed geek couldn't be further from the truth. … Thanks to the very developers who used to haunt those fleeting chat channels and techie forums, we now have the bright, bold, user-friendly colors of the social web, where the current generation of coding wizards can connect with seasoned veterans to brainstorm the future of the Internet.

The direct corollary of this is the creation of shared web communities, such as GitHub and StackOverflow, making it easier than ever to learn and contribute. 😄

2. What will we be learning?

Over the course of the next 6 weeks, we'll be building a foundation for essential Web Development concepts and skills, such as:

- How websites are made and how the internet basically works,
- How to create and publish your own webpage,
- How to code in groups and contribute to the developer community,
- Skills & tools for coding and carrying on after the course.

Our key focus will be on **Front-End Web Technologies**, mainly **HTML**, **CSS**, an important web framework, **Bootstrap**, and Javascript in the form of a framework, **jQuery**.

Front-End Web Development is about creating and styling the parts of a website, web service or application that users interact with (as opposed to the processes that happen behind the scenes; "back-end web development"). Skill in Front-End coding is easy to visualise and comes easily with practice. A good eye for design and an interest in the user journey helps too.

For this reason, Front-End technologies are also referred to as **client-facing**, or **user-facing**, technologies.

Another important distinction is between markup languages (HTML), stylesheets (CSS), and programming languages (JavaScript). We saw earlier that a programming language is one that is used to build solutions to problems, e.g. searching the web for particular information. What is **markup language**, and what is a **stylesheet**? Note: Every markup language consists of a set of **markup tags.**

TASK: Find out! (*hint 1*, *hint 2*, *hint 3*)

An Overview of Different Web Technologies

 **HTML** *HyperText Markup Language*	 **CSS** *Cascading StyleSheets*	 **JavaScript**
a markup language **Describes** the **structure** of web pages. HTML documents are described by **HTML tags**, and each tag **describes** different element.	a stylesheet language **Describes** the **presentation** of web pages, including colors, layout, and fonts. It enables **responsive design**; for one to adapt the presentation to different types of devices.	a programming language Allows user interactivity, and enables web pages to be dynamic. (Usually without needing to reload the page) *It is a type of programming language, scripting.
Makeup "equivalent": When you select individual products to form your makeup set/kit.	*Makeup "equivalent":* When you select and apply "themes" and colour schemes to your makeup.	*Makeup "equivalent":* When you apply special makeup, such as sparkly things, or different and beautiful

e.g. a powder brush and face powder is in completely different category than eyeliner	e.g. matching the colour of your blush with your lip makeup	eye makeup for a particular occasion e.g. Urban Decay's Naked palettes
Here is a basic HTML demo from W3Schools.	Here is a basic CSS demo from W3Schools.	Here is a basic JS demo from W3Schools.

How do they all come together?: Check out this basic example! (You're not expected to understand any of the syntax 😊)
Why are we using a make-up analogy? Far too often, we restrict ourselves to not thinking outside of the box in coding, and I wanted to show that you can indeed use a fun example, such as makeup, to think about Front-End Development.

3. What does this course lead to?

Many CodeFirst: Girls alumni have gone on to a range of careers in tech, some of them even founding their own companies.

Perhaps you will join the developer community as well, or perhaps this will be a bridge for you to develop more server-side skills and learn to write your own applications and services. Here's an example of some things you can create with the tools we will teach you in this course.

4. How should I learn coding?

There are many excellent resources on the web for learning a lot of the material we will be covering. We don't want to reinvent the wheel and will unashamedly point you towards better sources of information when they exist. Rather than teach you everything from scratch, we aim to guide and support your learning, focusing on core concepts and exercises to jump straight into coding.

In order to make this course as beneficial for you as possible, we will be running workshop-styled sessions, allowing you to **practice**, and **ask us as many questions** as much as possible. The forum feature on our **Facebook group** will also be really handy. Some course instructors might also create a **Slack** group.

As you grow in your development journey, you will find that Google and StackOverflow are your best friends, and GitHub is almost like an online home.

5. Is there anything I can do to prepare for the course?

One of the biggest challenges in a course like this is dealing with the different operating systems and hardware that you'll be working on.

That does mean that there's a bit of setting up that you will need to do to be ready for the first session. Please be patient with this - installing software is sometimes fiddly and not always predictable. Most of the work we get you to do won't be like this!

If you have any problems, please contact your instructor or the CF:G team!

To Do	What/Why?	Links	Done ?
1. Create a GitHub Account and get your GitHub Student Developer Pack **(N/A for Professionals)**	GitHub is, informally, a code sharing and publishing service, and a developer network. Formally, it is a web-based repository hosting service for a version control system (*that tracks file version changes)* called Git (*more later).* The Student Pack provides useful tools for free, for **current students only, not available for non-students (Professionals & General courses).**	GitHub, Student Developer Pack	
2. Install Google Chrome	Chrome is a free web browser provided by Google. It comes with a good set of developer tools that we will be using over the course.*	Google Chrome.	

3. Sublime Text** or Atom	A Text Editor is a Graphical user interface (GUI) that is built for writing and editing code that can be processed directly by the computer. We have chosen these as our recommended editors as they: (a) gives a good user experience on mac, linux and windows; (b) is easy to get started with; and (c) can be easily customised when you get more advanced. Sublime text has a slight advantage for front end development work due to the autocomplete function. However people occasionally have issues with Sublime, in which case Atom is comparable an alternative.	Sublime Text or Atom	
4. Install GitHub Desktop Client	A GUI for you to start using GitHub to collaborate on projects, without having to touch your command line.	GitHub Desktop Client	
5. PRE-CLASS PREP WORK***	In the first session we will be looking at HTML and the overall structure of the internet. We won't be spending much time on these, so it will help if you've at least seen them before. To help everyone has a basic amount of knowledge to start, please **complete the whole of Project 1**, *(this will probably take about 20-30 mins),* and watch a **3-min video on how the Internet works**	General Assembly Dash, Cisco Internet Video	

Additional Notes:

* Firefox also comes with an excellent set of developer tools (via its firebug extension). There doesn't seem to be much between Chrome and Firefox+firebug as far as the tools we'll be using go. The decision to recommend Chrome was fairly arbitrary, but means that everyone will be using the same software.

Sublime Text costs $70 but **you can use it for free as it has an indefinite evaluation period. The duration of the course seems like a pretty reasonable period of time to evaluate the software. If you go on to use it after that you should consider getting a license - it's a good product and the guys who make it have to eat!

*** **Optional Additional Prep Work**: (*For those of you who would really like to be prepared*) Sign up to Codecademy. On the Codecademy web track: (Only the lessons are free, so we'll stick to doing those 😆)

1. Do the lessons from "1 Introduction to HTML".
2. Do the lessons from "2 HTML Structure: Using Lists".
3. Do the lessons from "3 HTML Structure: Tables, Divs, and Spans".

Well done for getting to this point! You should now have all everything you need to start the course.

We look forward to meeting you in your first session!

□□□