Intro to Web Design

Toronto Hacker Club | Mozilla Inc.
Minh Ha | Alexander Kallaway | Justin Richardsson

Who are we?

Alexander and **Minh** are self-taught software developers, and they work at Vretta Inc.
Justin is a visual and concept designer, fascinated by Information Architecture and UI/UX research.

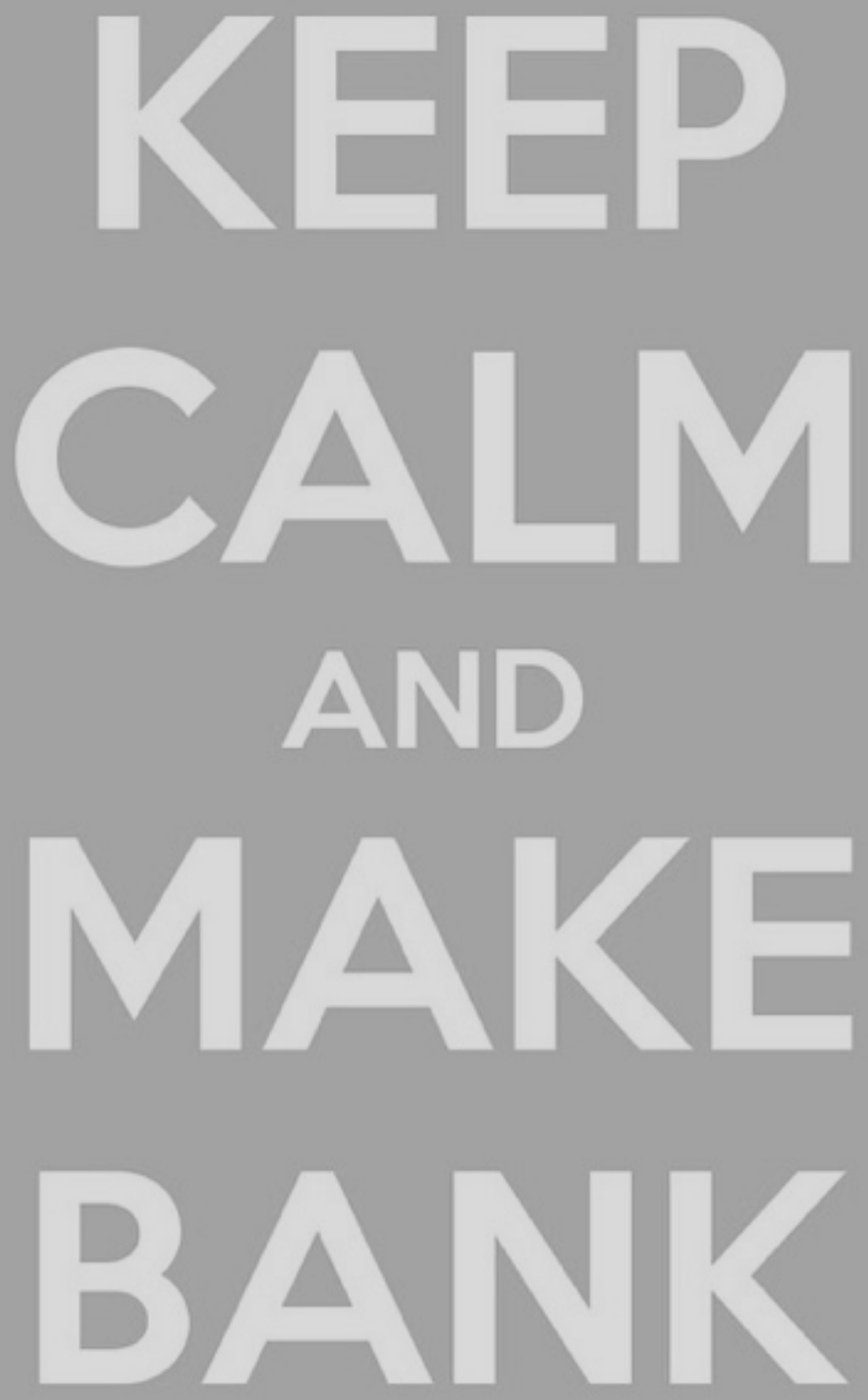
Alex is interested in psychology and habit formation, as well as in the optimization of the learning process. Minh's passion for technology led the former suit-clad finance professional to become a software developer.

Alex enjoys playing the violin, learning languages and about different cultures. He loves his wife Anna and their dog Oreo. Minh loves playing guitar like a rockstar, learning everything code and following cool news in tech. Most of Justin's free time is spent reading, cooking, out sailing, hiking or getting in trouble on his bicycle.

We love debugging the world by teaching it how to code with Free Code Camp;

Yes, programmers and designers can also have a cool and fun life.

and devs get paid very well ;)



Let’s talk about the Web
…Internet, the web, servers and all that jazz.

Internet is a defined as a global system of interconnected computer networks, and each of these networks share or exchange information through something called ***Protocols***.

Each of these ***protocols***, in short, establish certain ***standards*** how these computer networks send and receive ***information*** and how they handle ***errors***.

One of these protocols is the one known as ***HTTP*** (Hyper-Text Transfer Protocol), which is the one in charge of most of what we call ***the*** *(World Wide)* ***Web***.

The Web is only a part of Internet. But Internet is in fact much, much more than just the Web.

HTTP communicates two machines: the ***Client*** (your computer, cellphone, tablet, or any other receiving end you can think of with the assistance of the ***Web Browser***) and the ***Server*** (a more powerful computer somewhere else in the world, “serving” the information to your device and many other clients at the same time).

The most common way those Servers transfer information to the Client machines, is done through HTML, CSS and JavaScript.

HTML (HyperText Markup Language) is the main language through which information is displayed in the web. The structure and content of the page. You’re going to learn a whole lot of its basics today.

CSS (Cascading Style Sheets) Is a styling language, designed to help you customise the way the page is displayed, literally. In other words, it’s web’s make-up. We’re also going to taste quite a bit of it here.

JavaScript is an interpreted computer programming language, originally made as part of the browser so that the clients could interact with the page being displayed. We won’t see much of it today as this topic is a bit more advanced. The good news is you don’t really need it for this site we’re building.

There’s much more information, but enough with these boring lectures and let’s get to what we came here for!

Designing a website
…And just so you won’t make the usual mistake

*First and before anything else, this **visual design primer***

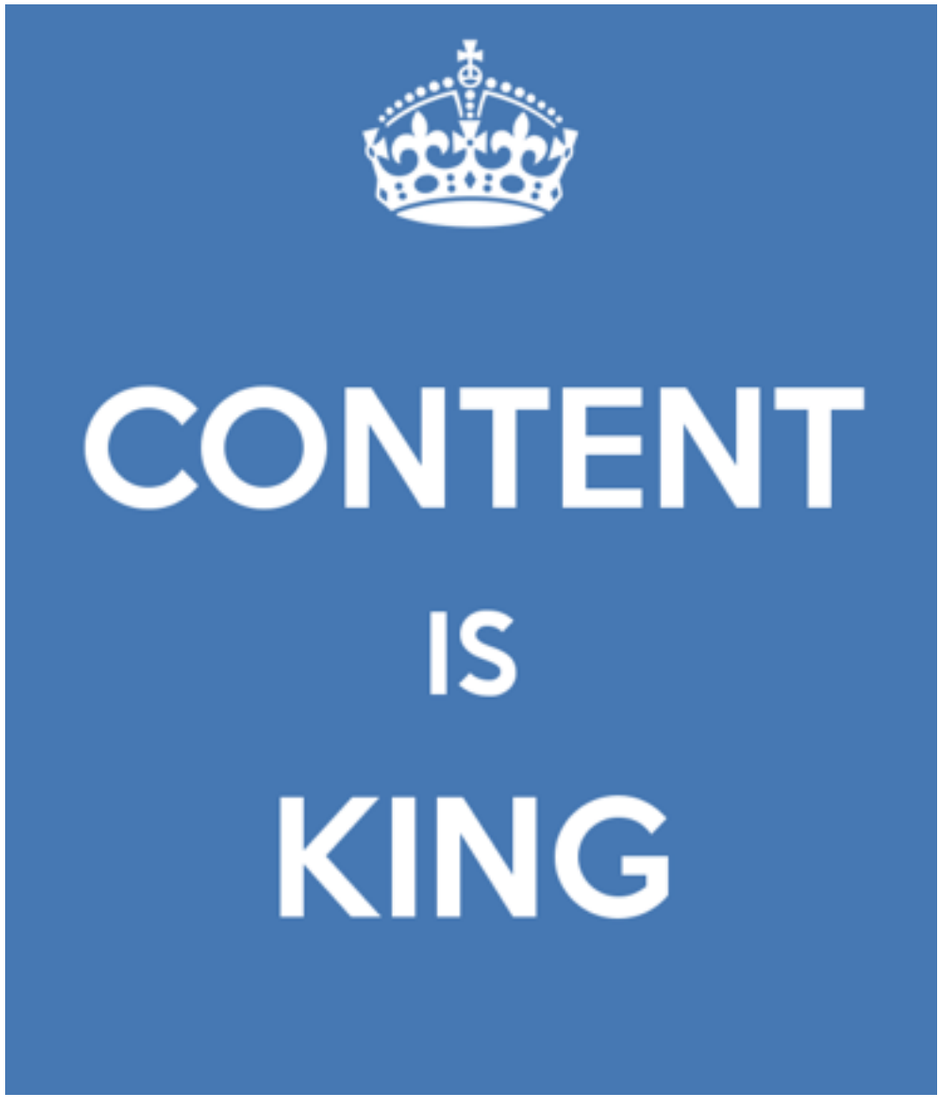
Don’t become another despicable artist-wannabe, focusing only on aesthetics and forgetting most of the things made by smart people have a well thought and defined function (even if the function is to move and inspire others). Be smart too! Work with a purpose.

Design is more a science than an art. Yes, it’s an art too, like Martial Arts, in that they require hours over hours of practice and persistence in order to acquire mastery. But there’s a method to this mayhem… and you’re designing a webpage for a reason.

Don’t waste your visitor’s time and attention.

Granted, your site should look nice, but it’s better to have an ugly yet functional website doing its job efficiently than a frivolously pretty page that doesn't work at all.

(If not, ask Microsoft and how they’ve made it,
until they’ve finally learned… sort of, almost)



Designing a website (continuation)
…Content is the single most important ingredient

Begin by defining what the ***website's purpose*** is, meaning what you intend to show and tell. Your visitors *must* be able to easily figure out ***what to expect*** from your site, as well as ***how to navigate*** through it.

And believe me, if they can't "get it" in the first few seconds, they'll drop your site like a hot potato and never turn back —Unless your site is one of those weird government pages, and your users are stuck with you.

Don't make your users feel stuck!

So, the idea is for you to have a ***well defined story*** to tell them. You basically will hold their hands and ***walk them through*** whatever process they're there to accomplish.

It is your mission to make sure they have a pleasant and stress-free experience, so they'll want to come back.

Keep in mind they may or not know what you're talking about, and it may well be their first time doing something like what your site is there to help them do. Guide them.

All this is called ***User Experience Design*** *(UX)*, and it's a highly sought-after, marketable skill. It is a very interesting field, and its results make all the difference in the world.

Furthermore, well planned content and structure will make the search engines (Google, Yahoo, Bing, etc.) like your site and make it show up closer to the beginning of the lists in any relevant search results. You want this, or else how are new visitors going to find out about your site?

This concept makes part of what we call ***S.E.O.*** *(Search Engine Optimisation)* and though many insist it is a myth, or even a fool's errand, well planned content is not only what gives a site sense and understandability; it also will make your job remarkably easier. Bad content will force you to remake the site many times over.

So, what content do we need?
…Titles, texts, images, and so on.

Go ahead and take a look at a couple of websites. Have you noticed how they consistently have more or less the same elements?

1. Name, logo, or some other sort of way to identify the site. Notice how it usually is at the top left part of every page. (***A brand***)

2. Some sort of menu bar with the main sections of the site, sometimes with a space to type and search for stuff. (***Navigation***)

3. A ***main space*** showing the bulk of the content, images, links to other sites or parts of the same.

4. A closing at the bottom with some general information about the site, it's owner and some secondary functions. (***Footer***)

And these are just some of the elements found in most sites. They're of course not mandatory at all, but you'll be hard-pressed to find a website without these.
Go on and try. I can wait.

See? also, did you notice how these elements are pretty much in the same size and place throughout the pages of a same site? This coherence is key. It's what's going to help your users from having to learn how to use your site every time they go deeper.

There are other elements, like sidebars, rotating image galleries and so on… Identify them.

Let’s define our content.
…And let’s do it right from the start, shall we?

It is safe to say if you’ve read this far in this document, you’re actually interested.
With some time and dedication you’ll quickly become phenomenal at this. I promise.

Thus, I dub thee “Design Superstar”. Let’s get to work already:

1. ***Our site’s brand:*** ☝️ ☝️ ☝️ ☝️ ☝️ (good guess!)

2. ***Navigation:*** We should tell the world how amazing you are, add some photos of nice stuff, and then add some way for our visitors to leave a message. Alas, you won’t be able to make this work just quite yet, but it’ll be ready and waiting until whenever you learn how to do that too.

3. ***Main space:*** What we mentioned above, but why don’t we also add an intro before all that.

4. ***Footer:*** Some additional contact info like “where to find us” (both in real life and online) and then some of the usual legal stuff to make sure they know we mean business.

As we are manually coding a basic website from scratch and without assistance of those “shortcuts” we’ve mentioned before, we’re going to skip a whole lot of additional details, but you'll be able to go deeper into the specifics of those on your own since you’ll already have the right vocabulary to begin with.

On a side note, those shortcuts, are called Libraries, Frameworks and APIs; and they’re all sorts of fun to learn and use. But don’t skip the basics or you’ll regret it for sure.

HTML!

The actual content and its structure.

Hands in that cookie jar!
…Finally, I know.

Let's open your favourite ***text editor***. Notepad or TextEdit? Righteous! You hardly need anything else anyway. Really!

Unless you're interested in having the editor autocomplete words, make suggestions and highlight the code to make it easier to read. Sounds good, eh? *Get a proper* ***Code Editor***.

Wanna do it like the pros? Why don't you give Sublime text or Atom a try? They're free, really good, sufficient, obscenely popular —and suspiciously similar. They both get it done.

Personally, I absolutely love Coda, but it's only available on Apple devices —yes, I can code on my iPhone and iPad—, and it can be pricey if you're not sure you like it.
There are many other different options, but I believe you're probably going to like one of these three.

What I'm trying to say here is that no matter how you pronounce it, all good tomato makes good sauce in the right hands. Choose your flavour, learn to use it and carry on.

Type our contents in a ***new document*** the same way you normally would an essay for school, save it as "***index.html***" and open it in your browser. Looks a bit messy, but it is indeed a page opened in a browser showing the contents.

Basic HTML+CSS code alone cannot break your computer.

On to making this code-y.

```
Design Superstar

About me
Gallery
Contact

I am the one and only Design Superstar
Web Developer - Graphic Artist - User Experience
Designer

About
This is Design Superstar's website, and it was
created so he could display a bunch of wolf and food
pictures, because wolfs are pretty awesome and who
doesn't like food. Right?

Gallery

Contact Me
Name
Email Address
Phone Number
Message

Location
301 Front St. W  Toronto, ON • M5V 2T6  Canada

Around the Web
About Design Superstar
Made with heaps of ♥ by John and Jane Doe.
Copyright © Design Superstar 2015
```

HTML structure - Document foundation

Most HTML documents share a basic structure:

- Document type.
- Start of document.
- Head, and stuff in it (settings, if you will).
- Body, and the actual content in it.
- End of document.

We'll go through them, and analyse in more detail as needed. Remember there's also plenty of research material online if you want to look deeper into anything.

Also *very important*, HTML like most languages has its own syntax and grammar, sort of… and it doesn't appreciate when you don't write it properly. And neither will the users.

Most HTML ***Elements***, are written like so:

Start tag *End Tag*
<tagname attribute="value">…content…</tagname>

And I say most because some elements are ***self-closing***, like the ***images*** (<img>), the ***line breaks*** (
), ***resource links*** (<link>, not to be confused with the links you click on), etc.

By the way, you can also write optional ***comments*** to help you document your code, and they won't be processed by the browser. In HTML it's done **<!-- like this -->**.

Let's add some structure to the document. Save and open it.

```
<!DOCTYPE html> <!--Learn this by heart, document type -->
<html> <!--start of the document-->
<body> <!--body's start-->
Design Superstar

About me
Gallery
Contact

I am the one and only Design Superstar
Web Developer - Graphic Artist - User Experience
Designer

About
This is Design Superstar's website, and it was created
so he could display a bunch of wolf and food pictures,
because wolfs are pretty awesome and who doesn't like
food. Right?

Gallery

Contact Me
Name
Email Address
Phone Number
Message

Location
301 Front St. W  Toronto, ON • M5V 2T6  Canada

Around the Web
About Design Superstar
Made with heaps of ♥ by John and Jane Doe.
Copyright © Design Superstar 2015
</body> <!--body's end-->
</html> <!--end of document-->
```

HTML structure - Document's Head

Before we get into laying our content, we should set the document up a little.

You should tell the browser we're going to be working on a page with contents in English, and what our page's title is going be.

The ***lang attribute*** we added to our HTML ***tag*** —like all attributes in HTML— is written with the attribute ***label***, followed by an ***equal sign*** and then the attribute's ***value*** in between quote signs (either single or double, both need the same at each attribute).

Then, let's set the page's title to something of your liking.

I'll set this one as "Design Superstar", for the sake of coherence. It's got a ring to it, methinks.

This title is the one that's shown on the browser window/tab.

Also, this is the title that search engines are going to be showing for your site.

There are other head tags, like <meta> (which is a self-closing tag) to set things like the site's description, its keywords, whether it should be read from right to left,etc.

Also, here is where you'll load external files for styles — Which we'll do in a while too.

You can also load external script files, libraries, and many other types of crazy things.

```
<!DOCTYPE html> <!--Learn this by heart, document type -->
<html lang="en"> <!--start of the document-->
<head>
   <title>Design Superstar</title>

</head>
<body> <!--body's start-->
Design Superstar

About me
...
```

HTML structure - Navigation

From now on there come many new elements.
Remember to research online for more info on them.

See what we did here? we added ***<nav>*** (a navigation block), ***<a>*** (an actual, clickable link), ***<h1>*** (heading, level 1, the main title) ***<ul>*** (an unordered list), and a few ***<li>*** (list items). As you can see they are ***nested inside each other***, forming a ***hierarchy***. One happy family.

This ***hierarchy*** is visible here because I indented the code on purpose, but indenting here doesn't change the result in the browser and it's only so that it's easier to read as you work —or if anyone else needs to work on your site.

Thus, *<nav>* is a ***parent*** to all the other elements and a ***child*** to *<body>*. The *<li>*s are siblings. You get the deal.

Also, some of these elements got attributes.

Think of ***id***s as sort of a first name. It should be unique in your whole code and it helps the browser pinpoint a specific element amongst the rest. There are those who question its use but that's a debate for other time.

The ***href*** is a hyperlink reference, and it tells the *<a>* where to take you when you click it. If you had written another site's address in there, it'd have taken us to that site. Nifty, isn't it?

It is very very ***important to close the elements*** that require so, or you'll end up in ***nesting hell*** (that's what coders call the annoying situation, if you ever hear that expression).

```
…
   <title>Design Superstar</title>

</head>
<body>

   <nav> <!-- Navigation -->
      <a id="brand" href="#page-top">
         <h1>Design Superstar</h1>
      </a>
      <ul id="navbar">
         <li><a href="#about">About me</a></li>
         <li><a href="#gallery">Gallery</a></li>
         <li><a href="#contact">Contact</a></li>
      </ul>
   </nav><!— End of navigation -->

I am the one and only Design Superstar
Web Developer - Graphic Artist - User Experience Designer
…
```

HTML structure - Header

A few of these are in fact ***new*** since the coming of ***HTML5*** (the latest revision of HTML, which hasn't been officially released yet but pretty much everyone is using it already).

A key idea in HTML5 is ***Semantics***, and it's sad to say an alarming lot of professional developers and designers do not apply, understand or even know what it is for.

You'll be able to read the code more ***clearly*** in a ***better organised*** document —which in turn makes it more legible by the browser and other machines such as ***assistive devices***. It is not only a legal requirement for major websites and an additional trick to improve your S.E.O. rating, but also a nice gesture of ***support*** for those with ***disabilities***. Let's be excellent to each other. Mind ***accessibility***.

<*header*> defines the beginning of any given part of the document and here serves as the introduction to our site.

We added an **<*img*>**, with a source file named *success.png*, located inside an *img* ***folder*** parallel to the *index.html* we are working on. This ***path*** is called ***relative*** because it's written in reference to the current document.

Our friend ** ** is a very popular wildcard and it's basically a container. More on it later (as there's plenty to say about it). A **<*span*>** is just a chunk of text. Think of ***class*** attributes as last names. Many elements can be of one same class when they share things in common (sizes? colors?).

<*em*> is used for emphasis, semantically. Remember **<*br*>**?

```
...
        <li><a href="#contact">Contact</a></li>
      </ul>
   </nav>

   <header> <!-- Header -->
      <img src="img/success.png">
         <div class="intro-text">
            <span class="name">I am the one and only
               <br><em>Design Superstar</em></span>

            <span class="skills">Web Developer -
Graphic Artist - User Experience Designer</span>
         </div>
   </header> <!-- End of Header -->

About
This is Design Superstar's website, and it was created so he
could display a bunch of wolf and
...
```

HTML structure - ‘About’ and ‘Gallery’

Ok, now we have a ***<section>***. Semantically, a section is a thematic ***grouping of content***, typically with a identified by using a heading. Thus, we’ve added a second level heading, ***<h2>***, which works as a subtitle… All this makes more sense if you think of chapters in a book.

Our first section there also has the *id* ‘***about***’, not only to remind us what the *h2* is already telling us, but so we can use it for styling later on when we get into that deal.

Later on, we have a ***<p>*** (paragraph) element.
Pretty self-explanatory.

For the ***‘Gallery’*** section, we of course have the corresponding *heading*, and then another *unordered list*, which then we fill with the images.

These images have been customised in size and orientation, just to make them easier to fit. Normally, images require a whole lot of attention on their own, and that’s more on the realm of visual design than web, but you may at least want to learn how to crop and properly optimise images for web.

Also, unlike regular picture taking and sharing, there are many different types of images, each with their own sets of characteristics, capabilities and weaknesses.

There’s a right type of image for whatever you’ll use it for.

Allow me to repeat myself: Don’t forget to close your tags, nest and indent them. You’ll thank yourself later for it.

```
…
Graphic Artist - User Experience Designer</span>
        </div>
   </header>

   <section id="about"><!-- About Section -->
          <h2>About</h2>
          <p>This is Web Rockstar's website, and it
was created so he could display a bunch of wolf and
food pictures, because dogs are pretty awesome and
who doesn't like food. Right?</p>
   </section><!-- End of About -->
   <section id="gallery"><!-- Gallery Section -->
      <h2>Gallery</h2>
      <ul class="grid">
         <li><img src="">
            </li>
         <li><img src="img/gallery/wolf1.png">
            </li>
         <li><img src="img/gallery/cookies.png">
            </li>
         <li><img src="img/gallery/pasta.png">
            </li>
         <li><img src="img/gallery/wolf2.png">
            </li>
         <li><img src="img/gallery/fruits.png">
            </li>
      </ul>
   </section><!-- End of Gallery -->

Contact Me
Name
Email Address
…
```

HTML structure - ‘Contact’ section

The plot thickens. We’re going to be using a ***<form>*** for this *contact section*, just so we can allow our users to ***input*** different bits of information in order for them to leave us a message. Each one of those ***<input>*** elements has its own corresponding ***<label>*** which in turn has a ***for*** *attribute*, equal to the *id* of the *input* it is related to.

Before we go any further, I don’t think you’ve noticed the way we’ve been writing the *ids* and *classes*, which is on purpose, in order to make it legible and to keep it clean.

Of course we’re being semantical enough to know what the element is for though we sure could do better, and then it is written in a specific way called a ***notation***.

The notation we’re using is ***camelCase*** (you can see it in the form’s *id*, ***contactForm***; because it just makes sense to me.

The way you use this *notation* is by capitalising the initial every word except the first one, no spaces in between and avoiding connecting words (for, by, from, etc.)

You will —once again— find your own flavour in time.

We added some attributes to these tags, mostly operational stuff to make it work when you start working under the hood.

```
…            </li>
        </ul>
    </section><!-- End of Gallery -->

    <section id="contact"><!-- Contact Section -->
        <h2>Contact Me</h2>
        <form id="contactForm" novalidate>
            <div class="form-item">
                <label for=“name”>Name</label>
                <input type="text" placeholder="Name"
id="name" required>
            </div>
            <div class="form-item">
                <label for=“email”>Email Address</label>
                <input type="email" placeholder="Email
Address" id="email" required>
            </div>
            <div class="form-item">
                <label for=“phone”>Phone Number</label>
                <input type="tel" placeholder="Phone
Number" id="phone" required>
            </div>
            <div class="form-item">
                <label for=“message”>Message</label>
                <textarea rows="5" placeholder="Message"
id="message" required></textarea>
            </div>
            <button type="submit">Send</button>
        </form>
    </section>

Location
301 Front St. W  Toronto, ON • M5V 2T6  Canada

Around the Web
…
```

HTML structure - Footer

Since we've been so good at putting this thing together in a smart and functional way, let's just get it over with so we can move on to the visual aspects of it.

We've added some *third level headings*, ***<h3>***, just to give some hierarchy to the contents in this *footer*. We also grouped the bits of info into *divs* to make everything be more organized and understandable.

There's an ***<i>*** element, which originally was meant to semantically give a different 'voice' or tone to the content, but in this case we're going to be using to show the ***social media icons*** with the help of ***Font Awesome***'s library — therefor the classes… there's documentation online for those and many hundreds of other icons and symbols.

```
…       </form>
    </section>

    <footer> <!-- Footer -->
        <div id="footer-above">
            <div><h3>Location</h3>
                <p>301 Front St. W<br>Toronto, ON • M5V
2T6<br>Canada</p></div>
            <div><h3>Around the Web</h3>
                <ul>
                    <li><a href="#" class="button
social"><i class="fa fa-fw fa-facebook”></i></a>
                    </li>
                    <li><a href="#" class="button
social"><i class="fa fa-fw fa-twitter”></i></a>
                    </li>
                    <li><a href="#" class="button
social"><i class="fa fa-fw fa-linkedin”></i></a>
                    </li>
                    <li><a href="#" class="button
social"><i class="fa fa-fw fa-dribbble”></i></a>
                    </li>
                </ul>
            </div>
            <div><h3>About Web Rockstar</h3>
                <p>Made with heaps of &hearts;<br>by <a
href="http://qlip.in">John and Jane Doe</a>.</p>
            </div>
        </div>
        <div id="footer-below">
          Copyright &copy; Design Superstar 2015</div>
    </footer> <!-- End of Footer -->

    </body> <!--body's end-->
</html> <!--end of document-->
```

And now, let’s make it look nice…

Let’s CSS this out!

Preamble

Let's first create an additional file, which here I've named ***style.css***, placed it into another folder (it's general good practice to organise external files in folders according to their type or function in the page).

Also, as you can tell, I've linked the minified css file for *Font Awesome* in order to make the social icons work.

Here's a link to their cheat sheet with all the hexadecimal codes for their symbols:

https://fortawesome.github.io/Font-Awesome/cheatsheet/
(tip: you'll use the last 4 characters instead of the ones already there)

———————————

Furthermore, you can type CSS styling in the HTML file (usually done in the document's head), or even inline as an attribute for the elements. This last option is common in email marketing, since mail most applications are not capable of rendering CSS the same way a browser does.

You may, however, probably not want to use it on a day to day basis, for it'll make your code difficult to maintain.

By the way, that's how you comment in CSS
/* Comment blah blah comment. */

```
<!DOCTYPE html> <!--Learn this by heart, document type -->
<html lang="en"> <!--start of the document -->
<head>
   <title>Design Superstar</title>
   <link href="css/style.css" rel="stylesheet">
   <!-- Custom Font to get the social media icons-->
   <link rel="stylesheet"
      href="font-awesome/css/font-awesome.min.css">
</head>
<body> <!--body's start—>
   <nav> <!-- Navigation —>
…
```

```
<!———————————————————————————>
<!-- This is just an example.-->

<head>
   <title>Design Superstar</title>
   <style>
      <!-- This is just an example.
           CSS code can go in here too,
           but will probably make your file really
           long and annoying -->
   </style>
</head>
<body style="/*More CSS*/">
```

CSS styling

With CSS you can modify pretty much any aspect of most HTML elements, but some elements will let you do it one way while others will not. It's your job as a designer to know what works on what elements and which browsers.

These days most browsers are more or less on the same page in terms of standards, but there's still work to be done in that sense.

CSS relies on inheriting the attributes down the nesting chain into child elements, unless defined manually or overruled by the element's specification.

Elements are addressed *by their name*, ***ids*** are addressed *with a # sign* before the id name, and ***classes*** are dealt with by *placing a .* beforehand.

Also, you can apply styling to ***multiple elements*** at the same time by *separating them with commas*.

Some elements have ***states***, like when you ***hover*** over it with the mouse or when you ***click***/tap it. Those states are done *using :state* right after the element name

Some attributes can be worked with in a shorthand manner, like the ones that have top, bottom and sides (those are handled in a clockwise sweep). Others have a defined order for the different possible values, like the border:

border: 1px solid white;

```
body {
   margin:0;
   padding:50px 0 0;
   color:gray;
   font-family:"Helvetica Neue",Helvetica,Arial,sans-serif;
}
p {
   font-size: 20px;
   margin:0 auto;
}
a,
a:hover {
   color:aquamarine;
}

h2,
h3 {
   color:#18bc9c;
   font-weight: bold;
   text-align: center;
   text-transform: uppercase;
}
```

Why not get hands on?

Thanks, and good luck!

I’m looking forward to learning from you.