An Overview of Web Development
(*for Designers*)

First of all, it is important to realize that there is a distinction between **front-end (AKA client-side)** web development, and **back-end (AKA server-side)** development. In this tutorial you will be introduced to front-end web developing. Front-end (client-side) web developing is much simpler than heavier server-side programing, and it is important that every designer today understands the basics of client-side code.

Although both the front and back ends are interrelated and important to the dynamic web we experience today, most interactive designers work together with developers on large projects that require server-side code. Client-side code is great for smaller websites, personal projects, mock-ups, data visualizations and more, so designers should be versed in writing for the client-side. However, when you get into designing a large site or application, say a social media site, or a website that has users with passwords or logins, well, that requires more complex code that needs to be interpreted by a server, and to execute these types of projects beyond a prototype it is best to work together with a developer or spend intensive time learning some server-side languages.

What can be Built on the Front-End?

- Basic Websites & Web Applications
- Basic Mobile Websites & Mobile Web Applications
  (Non-native apps & sites that run optimized for mobile but in a phone web browser)

What can be Built on the Back-End?

- Large Data-Driven Websites and Applications
- Websites that incorporate a **CMS (Content Management System)** such as Wordpress or Drupal; if writing a website for a client/customer, a CMS allows your client/customer to change & update certain elements such as text or images without having to write any code.
- Native Mobile Applications (apps that run native on iPhone, Android, etc)
- Computer software and operating systems

Learning the basics on the front-end side will introduce you to basic concepts on how computer languages work. This will make it much easier for you to talk and collaborate with developers; they will also respect you and trust you more if you understand what they do. The front-end talks to the back-end and even if you never get into writing back-end code, you will learn how the two sides relate and talk to one another.
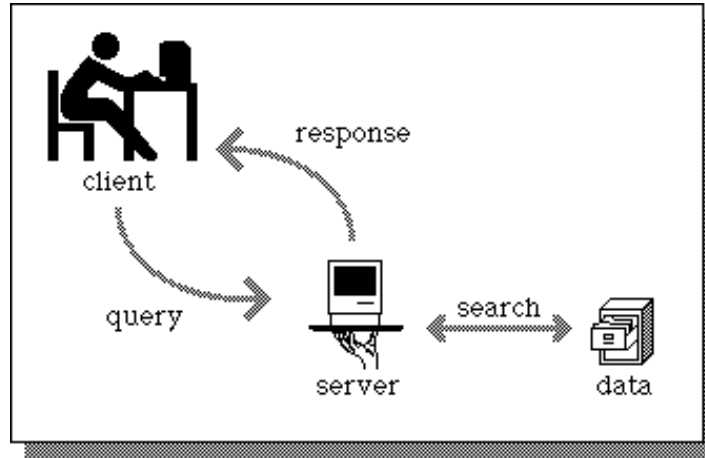
What does this mean: "Front-End is Client-Side and Back-End is Server-Side?"

A '**Client**' is a local computer such as your personal laptop or desktop.

A '**Server**' is usually a large computer with a lot of storage and memory that 'serves' a number of client computers.
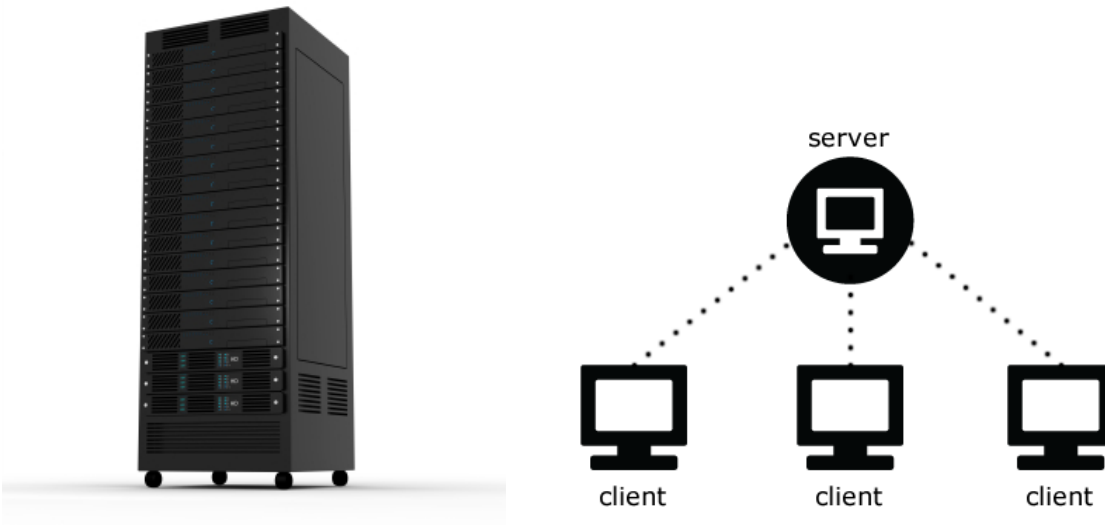
Small servers at businesses and organizations are typically large humming computers kept in a locked closet or room; they run the 'intranet' or internal internet only an organization has access to, and may have drives and storage space that all employees can access and write to. A main function of most company servers is to run the company's database, storing tables of information on customers and accounting information.

Servers are also used to '**host**' websites, or deliver ('serve') files, such as html files, images, and more, to the appropriate place on the internet. Most small to mid-sized websites today buy 'hosting' on a remote server when they purchase a domain name (url for a website); **godaddy.com**, **fatcow.com**, **wordpress**, and **github** are all examples of sites where you can buy space on a remote server in order to host your webpages and files. Github even offers free hosting if you keep your source code openly available (something you can't do if you're creating a professional website for a corporate client).

## Ex: buying space on a GoDaddy host server

Some large companies do in fact have the actual physical computers responsible for hosting their sites; Google for example does not buy space on a remote server to host their massive content but rather owns the hardware to host their own services and deliver them on the web. Of course, they also host content for many users; files you store on your Google Drive for example are being hosted by Google servers.

Below: Some of Google's servers…
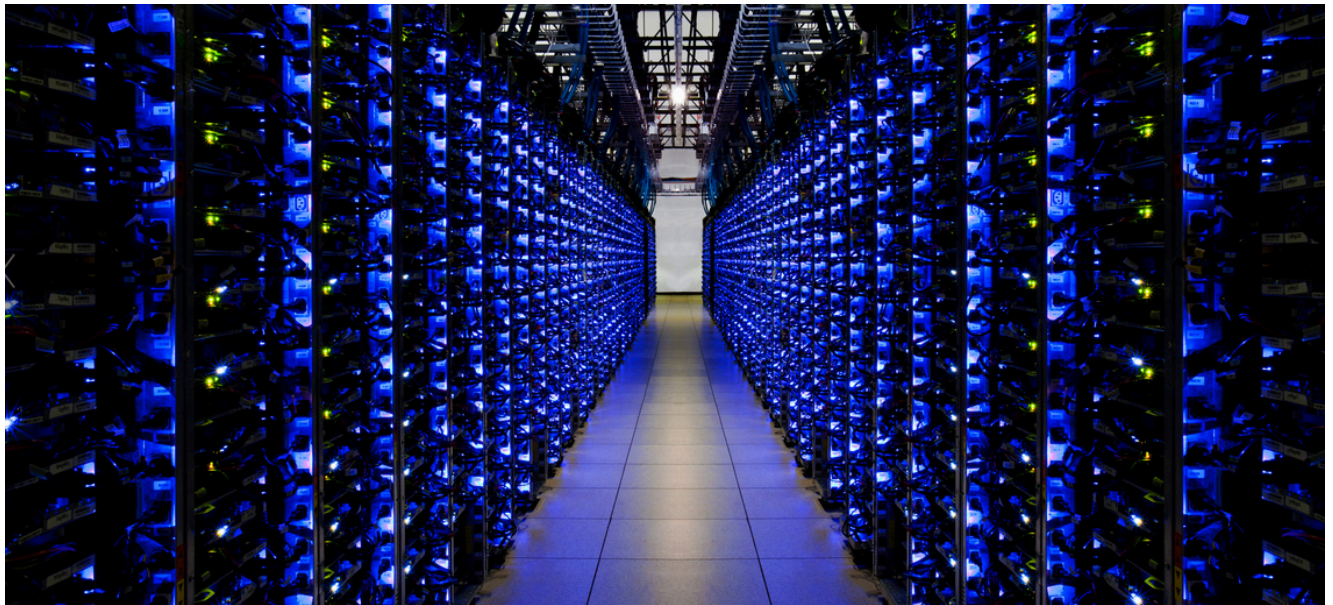
Take a better look at google's colorful high-tech data centers here:
https://www.google.com/about/datacenters/gallery/#/all

Getting back to what this means in terms of code…

The World Wide Web (WWW or W3C) is actually a massive system of interlinked **hypertext documents** accessed via the internet; hence **HTML (Hypertext Markup Language)** which is the language and structure that is required for **web browsers** such as Chrome , Safari, and Firefox to be able to read web pages.

Client-side means that the *action* takes place on the user's (the client's) computer.

Server-side means that the *action* takes place on a web server.

Basically a client (such as your laptop) talks to a web browser (such as Chrome, Safari, Firefox), and when client-side code such as **HTML**, **CSS**, and **Javascript** runs **locally** (on your client computer), any scripts/programming run **after** the HTML is loaded into the browser.

Server-side languages on the other hand need to be 'served' up by a server before the browser can read them. The scripts/programming for server-side files run **before** the HTML gets loaded into a page.

So if you are writing PHP or some server-side language locally (on your client computer) and you do not have a server connected to preview it in a browser, the browser will not be able to read it.

Sites like Wordpress and Facebook run on **PHP**, which is one common server-side language that is used to create large, dynamic websites. Many people who enjoy HTML do actually get into PHP and write their own PHP files, often times its because they have written a website for a client or friend, and that person wants to be able to update the content without writing any code. That requires a **CMS (Content Management System)** such as Wordpress or Drupal, both of which run on PHP.

If PHP is of interest, it requires a few more steps to set up your personal computer to preview your local PHP files in a browser, but basically all you have to do is set your personal computer up as a server. As long as you are not running a massive site on your local computer this is fairly easy to do and will not slow down a computer with a decent amount of RAM. If you plan on building many client websites however, you should buy space on a remote test server rather than use your personal laptop to preview & test a bunch of client websites.

Server-side coding is beyond the scope of this tutorial but it is important to know what server-side programmers to do and which languages do what on the server-side.
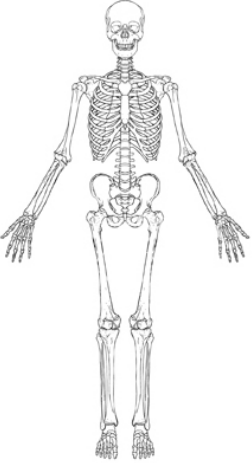
Quick Glance: Major Client-Side & Server-Side Code/Languages

Client-Side	Server-Side

• HTML
• CSS
• **JavaScript**
• **ActionScript**
  (used by Flash; now dying)
	• **PHP**
	• **SQL** and **MySQL**
	• **Ruby**
	• **Python**
	• **C and "C-like" Languages**
	  (C++, C#, Objective C)
	• **Java**
	• **Swift**
	• **ASP.NET**
	• **R**
	• **JavaScript (server-side JavaScript)**
	  (ex: Node.js)

Note: In actuality, HTML and CSS—while they involve writing code—are not actual scripts or programming languages like all of the others listed. Rather HTML provides an organized outline and hierarchy of the web page content and **CSS (Cascading Style Sheets)** provides styling and layout. HTML and CSS essentially provide the basic design hierarchy and styling of webpage content and are therefore extremely useful for designers to learn.

The next pages explain some of these languages in more detail. It is important that designers at least understand the roles played on the client-side; however, it is also good to know the basics of what some server-side languages do and which server-side developers are needed for interactive projects.

Details: Client-Side Code

HTML	CSS	JavaScript
The structure	The Skin	Make it dance

HTML (Hypertext Markup Language)

Hypertext Markup Language, or HTML, is a markup language that uses tags to describe the content of a webpage. A web browser reads HTML files and composes them into visible pages. You can think of HTML as an outline that you use to organize the content of the page into a hierarchy of importance.


```
<! DOCTYPE html>
<html>

<head>
<title>Appears in Browser tab only</title>
<link Your.css />
<link  Any other plugins /link>
</head>

<body>

    <h1> Heading Level 1- Most Important</h1>
        <h2> Heading Level 2  </h2>
            <p> paragraph 1</p>
            <p> paragraph 2</p>
            <p> paragraph 3</p>

        <h2> Heading Level 2  </h2>
            <img/>   an image
            <p> paragraph 4</p>
            <img/>   an image
            <p> paragraph 5</p>

        <h2> Heading Level 2  </h2>
            <ul>    An Unordered (bulleted) list
                <li> list item 1 </li>
                <li> list item 2 </li>
                <li> list item 3 </li>
                <li> list item 4 </li>
                <li> list item 5 </li>
            </ul>

</body>

<html>
```
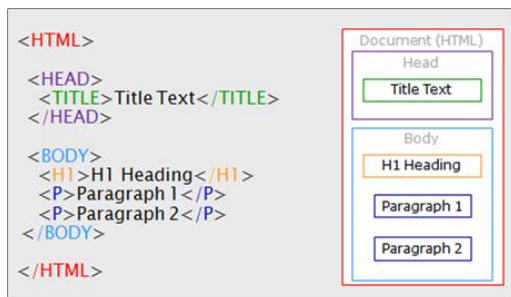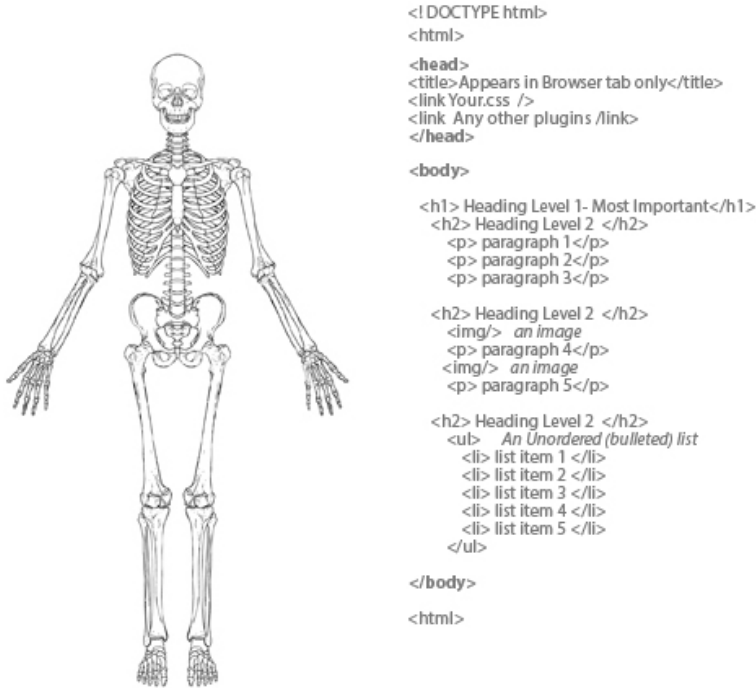

XHTML and **HTML 4** are the immediate predecessors of **HTML 5**, which is now supported by all browsers. HTML 5 has added support for better page organization and rich media. Tags such as <nav> <video> and <article> are examples of HTML 5 tags.

CSS (Cascading Style Sheets)

CSS stands for Cascading Style Sheets. CSS styles define how HTML elements get displayed in a page. They determine not only the design, colors, and typography, but also the way elements are laid out on the page.

There are 3 ways to insert CSS into an HTML document:

1. **Inline Styling**
   Adding styling to individual HTML elements. Anyone old enough to have edited the styling of their MySpace profile… well what you were actually doing was writing inline CSS.

   Ex: <div class="ui-bar ui-bar-e" **style="margin: 5px; font-size: 0.8em;">**

2. **Internal Style Sheet**
   Inserting all CSS for an HTML page into the head (<head></head>) of the HTML document using <style></style> tags.

   Ex:
   <head>
   <style>
   h1 {color: #FF0000;}
   h2 {font-size: 18pt;}
   p {margin-left: 25px;}
   </style>
   </head>

3. **External Style Sheet**
   Creating a separate CSS file, and connect an HTML page or pages to those styles by <link> tags added to the head (<head></head>) of the HTML document(s).

   Ex:
   <head>
   <title>About</title>
      <link rel="stylesheet" href="christie.css"/>
   </head>

CSS 3 is the most current version of CSS; the majority of its properties are supported by major browsers.

JavaScript

Unlike HTML and CSS, JavaScript is object-oriented code that involves actual programming. JavaScript is what adds most interactivity to webpages (though some interactive elements can be added using CSS). Things like slideshows, animations, and transition effects on a page are typically implemented through JavaScript.

Like CSS, JavaScript can be plugged into an html document in several places, namely in the head of the document, at the end of the document, or as an external document; some of this is dependent on the order the execution of the code needs to occur.

Ex: Internal JavaScript coded into the head of an HTML document.

```
<head>
<script>
function testFunction() {
    document.getElementById("christie").innerHTML = "Hello World!";
}
</script>
</head>

<body>
    <p>Click "Test", and our test function will be called.</p>
    <p>The function will display the message 'Hello World!'</p>
    <button onclick="testFunction()">Test</button>
    <p id="christie"></p>
</body>
```

Ex: External JavaScript linked into the head of an HTML document.

```
<head>
  <script type="text/javascript" src="scripts/jquery-ui.min.js"></script>
</head>
```


jQuery

jQuery is not another language, but rather a JavaScript library that makes it easier to write in a lot of the functionalities and interactive elements that would typically require long-hand JavaScript. jQuery essentially makes possible a kind of slang or short-hand JavaScript; to use jQuery it has to be linked into an HTML document head.

Details: Server-Side Code

While server-side code may not be useful for most designers to know how to write, it is useful to understand which languages do what and which types of developers are needed for which projects.

PHP

PHP (Hypertext Preprocessor) is a popular server-side scripting languages that is typically used make dynamic websites or web applications. PHP makes coding large, complex websites easier by shortening code and making it easy to create templates and other functionalities. Many CMS (Content Management Systems) such as WordPress and Drupal run on PHP for this reason. PHP can be easily embedded into HTML and when combined with CSS, Javascript, and MySql, a powerful level of advanced web development can be achieved.

SQL and MySQL

SQL stands for **Structured Query Language** and is designed for handling data in relational databases. Variations of SQL are used to maintain databases in commercial products such as Microsoft SQL Server, which is a commercial database management system written for Microsoft. However, most developers today use **MySQL** which is a free alternative to other SQL-based database systems. Because MySQL is free, cross-platform, and happily combines with PHP, it is the most popular database system today.

The combination of PHP and MySql is what allows many large social networking sites and applications such as Facebook to create user profiles and fetch content relating to data coming from specific users.

JavaScript (Server-Side)

Server-side JavaScript refers to JavaScript that, unlike regular JavaScript, runs server-side and is therefore not downloaded to the browser. Server-side JavaScript can be written 'asynchronously' which for our purposes, simply means that it can help things run much faster. **Node.js** is one popular server-side JavaScript Library often used in application development. Node.js is useful for real-time applications because its asynchronous nature allows for fast display of information without the need for refreshing.

Ruby & Ruby on Rails

Ruby is a dynamic, general-purpose, object-oriented programming language. Ruby syntax is similar to Python and Java and is highly customizable; it has become increasingly popular with the birth of the **Ruby on Rails**, a free, open-source framework that runs on Ruby and makes writing web applications much faster and easier. Like PHP, Rails is typically deployed with a database server such as **MySQL**.

C and "C-like Languages such as C++ C# and Objective C

C is a general purpose programming language and one of the most widely used languages of all time. C has been around since the 70s and many languages have been influenced by

the concepts and syntax of C including PHP, Java, Python, JavaScript, C++, Objective-C, and C#. C is still used today, typically for developing compilers and operating systems.

C++ was originally known as "C with Classes" and emerged in the early 80s as a language that evolved, branching off of C. C++ is still popular today and is useful in developing operating systems, desktop applications, software, and video games.

C# is also based off of C, and is a general-purpose, object-oriented language developed by Microsoft and is used to develop software and applications in Microsoft's .NET framework.

Ojective-C is another popular derivative of C and until 2014 has been promoted as the main programming language used by **Apple** for their OS X and iOS operating systems. The growth in Apple products and application development has made Objective-C a very popular language. At Apple's 2014 Developer's Conference it was announced that **Swift**, which is compatible with existing Objective-C programs, would be replacing Objective-C as Apple's language of choice.

Swift

Swift is a multi-paradigm, compiled programming language developed by Apple to replace Objective-C as the language used in developing for iOS and OS X applications. Swift has been considered a re-imagining of Objective-C using more modern concepts and syntax. It is 'multi-paradigm' which means it supports a variety of paradigms or programming styles—AKA makes it a more flexible language that can adapt to different projects. It can be seen as a move by Apple to make a language more competitive with Java, which is used as the core language for Google's Android mobile platform.

Java

Java is not related to JavaScript and is a separate language entirely. Java was released in the 90s and was influenced greatly by C and C++ and shares similar syntax. While Java isn't currently open source it is free and easily downloadable from Sun Microsystems. Java is object oriented and can run on many different platforms and operating systems. Because it is free, flexible, and allows for more simplified syntax than C++, Java is a highly popular developing language with over 9 million developers worldwide. Its popularity has increased with the launch of Google's **Android** mobile operating system, which runs on Java and runs Java applications.

ASP.NET

Developed by Microsoft and now open-source, ASP.NET is a framework for building dynamic webpages with HTML, CSS, and JavaScript. Knowledge of basic C# is needed to build applications with the ASP.NET framework. ASP.NET can be compared to PHP in that it is used to build powerful and scalable websites, particularly ones that speak to other Microsoft software and applications. For this reason it is more popular with businesses and corporations that use Microsoft systems or databases such as Microsoft SQL Server or Microsoft Dynamics.