# Overview of HTML

Now that we've got your attention regarding HTML, let's learn more about it.  HyperText Markup Language (HTML) is the main markup language for all web pages. HTML elements are the basic building-blocks of the internet.

## What is HTML

HTML is the language of the Web. It stands for Hypertext Markup Language. Its purpose is to let us communicate with the browser the meaning of the content that we wish to place on a webpage. To achieve this, HTML defines a number of **tags**that we can wrap the contents with. Here is an example of a tag:

<h1>Hello World</h1>

HTML tags **describe the content they contain.**For this example, there is a h1 tag that stands for heading 1. There are far more we will learn about!

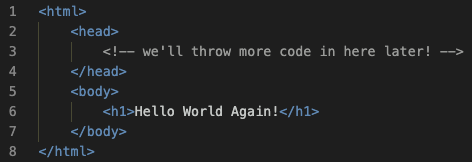
**Want to see what what it looks like behind the scenes?**

1. Open up your favorite website.
2. Right click in your browser and click **Inspect Element**. A new module should open at the bottom of your browser with various tabs. We're currently interested in the **Elements** tab. When you click on it, you should see the HTML

## Key Points:

**Opening and closing tags**

Since tags are supposed to wrap things, **most tags come in pairs**: one opening and one closing tag that denote the beginning and end of content. Tags without a leading forward slash are called opening tags while tags with leading forward slashes are called closing tags.



In the next example above, <h1> is an opening tag - it's equivalent to you telling the browser "Hey, I'm going to start putting in heading 1 content now". </h1> is a closing tag that means "OK, I'm done with the heading 1 content."

**Nesting**

Notice that between the opening and closing HTML tags on lines 1 and 8, we also have head and body tags. **Tags can encapsulate other tags**. This is called nesting. Nested items are **indented with a tab** to make the document easier to read.

The hierarchy that arises from **nesting** is called the **DOM** - Document Object Model.

**Commenting**

The format for commenting in an HTML file is as follows:

<p> Some paragraph content. </p>

<!-- This is a comment -->

# Parts of an App: Front-end v. Back-end v. Database

## Front-End

Front-end refers to the tech that allows information to be displayed to a user. This includes the design and images you see, the layout of the text and the webpages, and even some of the interactivity. This comes in HTML, CSS, and Javascript. For the Pre-Bootcamp, we'll give you a taste of HTML and JavaScript.

## Back-End

Back-end refers to the tech that moves and manages the data behind the scenes. There are many programming languages that can be used. They include: Python, Javascript, C#, Java, Ruby, and PHP.

## Database

As it might sound, the database is where all the data is stored. Wonder where your Netflix cue lives? It's in a database that Netflix owns and manages.

# The Request and Response Cycle

As many know now, the web is simply a network of computers that interact and communicate with each other. But how exactly do they communicate? What a great question! The most common and basic way computers communicate with each other is through the **request and response cycle**. There are generally two types of computers involved in this process: the **client** and the **server**.

## The Request

As you might have guessed, the request and response cycle begins with the client computer making a request by entering an address into a web browser. Once the request is on its way, the first stop it makes is at the **Domain Name Server (DNS)**.

#### The DNS

The actual address of a web page is an IP address which looks like a series of four period-separated numbers. Let's say I own a house on 123 Main Street and I chose to name my house Breezy Acres. The actual IP address of a web page is equivalent to my street address, and the name Breezy Acres is equivalent to the domain name you typed into the browser. So the **DNS** reads the URL and routes your request to the proper IP address.

## The Response

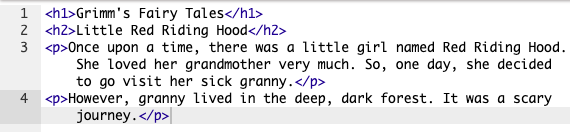
Once the **server** receives the client's request, given that the server approves the request, a **response** is prepared on the server-side. There are many different ways the server can prepare a response for the client (querying a database, handling logic, executing algorithms, etc) but we'll cover that in more detail once the bootcamp starts. An important thing to note here is that although the response can be prepared in many different ways, it will always be some combination of **HTML, CSS, and JavaScript**!

## 

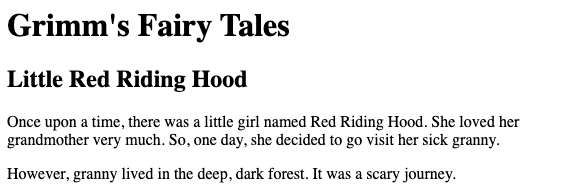
# Patterns: HTML

Programming is often like music: there's patterns and predictability if you know how to listen for it. Take a look at the two images below. One of them is the actual HTML code and the other is what a user would see based on that code. See if you can answer the questions below.

### HTML Code



### User Interface



**Head**

The head tag is one of the 3 core tags required in a HTML document. It contains all the metadata about your site and is not displayed on our page. Here's an example head section:

<head>

<meta charset="utf-8">

<title>My Awesome Webpage</title>

<meta name="description" content="The text here describes what the webpage

is about. It's what will show up in search results for search engines like

Google under the title of the webpage. It's important that this is relevant

to your page and well-written!" >

<link rel="stylesheet" href="my\_css\_file.css">

<script src="my\_javascript\_library.js"></script>

<script src="another\_javascript\_file.js"></script>

</head>

Let's go through this line by line again:

**<head>**

This is the **opening head tag**which indicates that we are about to begin talking about the properties of the document.  It's clever that the <head> tag holds onto the properties and brains of the page.

* **<meta charset="utf-8">**

Properly-encoded web pages declare the encoding to a browser through a meta tag in the header. **Without this tag,** a browser may not know to switch to the proper encoding and characters may be displayed as gibberish.

* **<title>My Awesome Webpage</title>**

This is the title of your web page, which means that when you open this page in the browser, the tab that opens it will read "My Awesome Webpage". This will be the name that it gets bookmarked with, as well as the name that is used when it is displayed as a search result in a search engine.

* **<meta name="description" content="description content">**

The description meta tag is used by search engines when displaying results.

* **<link rel="stylesheet" href="my\_css\_file.css">**

This line links a **stylesheet** to our page, which will determine how our HTML elements are **visually displayed** on the page. We will learn more about what goes into my\_css\_file.css in the CSS section of Web Fundamentals!

* **<script src="my\_javascript\_library.js"></script>**

This line links a **JavaScript** or **jQuery** file to our document. JavaScript makes our pages **interactive**. We will learn more about these files in the jQuery section.

**NOTE: You can link to as many stylesheets or JavaScript files as you want within the head tags.**

**</head>**

This is the **closing head tag**. It indicates that we are done talking about the properties of our page and we can move on to the body!

**In case you're curious: more on meta tags with an example...**

Like we mentioned, meta tags make your web page more meaningful for search engines like Google.

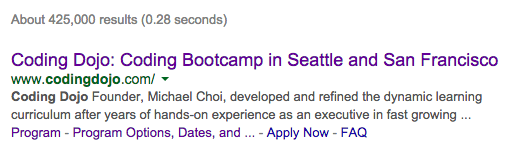
The *content*attribute of the *description* meta tagdescribes the basic purpose of your web page (a summary of what the web page contains). For each web page, you should place a concise and relevant summary in this section.

For example, this description:

<title>Coding Dojo: Coding Bootcamp in Seattle and San Francisco</title>

<meta name="description" content="Coding Dojo Founder, Michael Choi, developed and refined the dynamic learning curriculum after years of hands-on experience as an executive in fast growing...">

This is what shows up in Google's search engine results page:



The **"Coding Dojo: Coding Bootcamp in Seattle and San Francisco"**comes from the <title> tag.

# Tags

It's time to learn about the most common tags that you'll be using inside the <body> tag. We're going to get into the common elements that you might want to place on a webpage, which are:

* Headings and paragraphs of text
* Images
* Links
* Lists
* Tables
* Forms

In this section, we will cover everything but forms.

## Headings

**A heading is a section title**, which means that often (but not always) each section of your page will have a heading. There are 6 levels of headings that you can use, named <h1> through <h6>, each indicating the importance of its section.

Let's look at our current page for examples. At the top of this page, you see Optional: Tags in large and bold letters. This is an **<h1>**tag, written as...

<h1>Common Body Tags</h1>

...and it's **the main heading** for the whole page.

## Paragraphs

Any chunk of text is a paragraph and, therefore, needs to be encapsulated in paragraph **<p>**tags.

Here's an example of how to use <p> tags:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse cursus

velit lectus, odales lorem id orci blandit, ac tincidunt lorem po

rta. Sed euismod a arcu sed mollis.</p>

<p>Maecenas imperdiet risus at nisl aliquet, eu ullamcorper enim imperdiet. Sed

id metus consectetur, sollicitudin eros at, dapibus ipsum. Morbi cursus nibh

sit amet port, pretium sagittis mi. Fusce rhoncus i

mperdiet eros, ac porta ligula ullamcorper in. Suspendisse nulla urna, facil

isis non nunc ut, faucibus condimentum leo.</p>

## Images

There are two ways that we use images on a web page: as **page elements** (such as album art in Spotify, or the photos in your Facebook feed), or as **background images** (this is covered in the CSS section).

Images have two required attributes: **src** and **alt**.  The **src** attribute stands for **source**. This is the link to where the image is residing. The **alt** attribute stands for **alternate**, which is a few words of text to describe the image, in case it fails to load. This is also used by screen readers, for the vision-impaired.

## Links

Links are things that we click on that redirect us to another page. Usually, links are in text format, but you can also use an image as a link.

The tag used for links is the **<a>** tag, which stands for the **anchor**tag. Similar to images, links also need to have an attribute that tells the browser where the link will send someone to. For links, this is called the **href** attribute, short for hypertext reference.

**Possible values for the href attribute are:**

* An absolute URL - points to another website (like href="http://www.example.com/default.html")
* A relative URL - points to another file within a website (like href="default.html")
* An anchor URL - points to an anchor inside a page (like href="#top")

Example:

<a href="http://www.google.com">Click here to go to Google</a>

<a href="www.google.com"> <!-- will this link work? -->

<img src="">

</a>

## Lists

How we think of lists in HTML is a little different from how we think of them in our day to day life. It is any collection of elements that are of the same type. The most common use for lists in HTML are for navigation links.

* Home
* About
* Contact Us

There are two types of HTML lists: **ordered lists** (lists that are numbered) and **unordered lists**. Ordered lists use the **<ol>**tag, and unordered lists use the **<ul>**tag. Both lists use the **<li>**tag to describe each **list item**.

Example:

<ul>

<li>

<a href="home.html">Home</a>

</li>

<li>

<a href="about.html">About</a>

</li>

<li>

<a href="contact\_us.html">Contact Us</a>

</li>

</ul>

## Tables

We will often find ourselves using tables to display information. Tables have many tags associated with them because they are made up of many different parts. They have:

* A table **head** <thead>, which contains **rows**<tr> and column **names**<th>.
* A table **body** <tbody>, which contains **rows**<tr> filled with table **data**<td>.

So the tags we need are:

**<table>, <thead>, <th>, <tbody>, <tr> and <td>**

Example:

<table>

<thead>

<tr>

<th>Name</th>

<th>Email</th>

<th>Phone number</th>

</tr>

</thead>

<tbody>

<tr>

<td>Sample Name</td>

<td>an\_email@gmail.com</td>

<td>555-5555</td>

</tr>

<tr>

<td>Another Name</td>

<td>another\_email@gmail.com</td>

<td>444-4444</td>

</tr>

</tbody>

</table>

copy

**Bonus**: Copy and Paste the HTML snippet above into one of your .html files and load it into your browser. Always be curious!

# Indentation Review

This is such an important concept to understand for the rest of your development life that we will be reviewing the last assignment together.

<!DOCTYPE html>

<html>

<head>

<title>Basic I</title>

</head>

<body>

<h1>What language do you love?</h1>

<p>I love HTML!</p>

</body>

</html>

By analyzing our HTML, we notice that we have an opening HTML tag on our second line that does not close till the end of our HTML document. This means every single element inside of those HTML tags should be indented over once as they are all children elements of the HTML tag. The same is applied for the title tag which is nested in the head element. Likewise, for the h1 tag and the p tag, they are both children elements of the body tag and grandchildren of the HTML tag. Now let's cover the second example.

<!DOCTYPE html>

<html>

<head>

<title>Basic II</title>

</head>

<body>

<table>

<thead>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

<th>Password</th>

</tr>

</thead>

<tbody>

<tr>

<td>Brendan</td>

<td>Stanton</td>

<td>brendanrocks@gmail.com</td>

<td>FakePassword123</td>

</tr>

</tbody>

</table>

<h1>Here is a list of my favorite things:</h1>

<ul>

<li>Food</li>

<li>Bandwidth</li>

<li>Coffee</li>

<li>Beach</li>

</ul>

</body>

</html>

Review this code and apply the parent, sibling and child relationships to understand this content a little easier.

Happy coding!