

# HTML: The Basics

Let's take a dive into HTML (Hyper Text Markup Language). HTML is the markup language that provides us a structured way to create web pages. When we use the tags available to us through HTML we can control how almost everything is presented to the user via their web browser. Your web browser it what interprets the tags to present what the end user sees.

## Useful Resources

[HTML Element/Tag Reference](#)

[HTML 5 Reference](#)

[Forms in HTML](#)

[Pixlr \\* Online photoshop clone](#)

---

1. HTML and its Tags
  2. Basic HTML page structure
  3. Site Organization
  4. Some common tags explained
    - Headings
    - Paragraphs
    - Lists
    - Links
    - Images
    - Tables
    - Divs
    - Spans
  5. Exercises
    - [Simple Travel Site](#)
  6. Homework
- 

## Hyper Text Markup Language

HTML is a markup language interpreted by the browser. It uses specific tags or elements to tell the browser how things should look when presented visually to the user. Lets take a look at some concepts behind how these tags work before we dive into actual tags.

## HTML and Its Tags

- Tags are surrounded by angle brackets.

```
<tag>some content</tag>
```

Notice the closing tag has the / character. This lets the browser know that this is the matching closing tag. Tags are almost always followed up with matching closing tags.

*Not all tags in these examples are real HTML tags. They are being used to explain the concepts.*

## HTML and Its Tags#

- Tags can also be nested inside each other.\*

```
<pizza>
  <topping>pineapple</topping>
  <topping>ham</topping>
  <topping>cheese</topping>
</pizza>
```

- Tags use something called attributes to add additional functionality or describe behavior.

*Some attributes related to styling have been deprecated over time. It's best practices to do styling through CSS (Cascading Style Sheets). Some common attributes like border and align are examples of attributes that have been deprecated to enforce best practices of controlling this with CSS.*

```
<tag attribute="value">some content</tag>
```

- Some tags are self closing. These tags are usually not intended to wrap content. The `<img>` tag is a

common self closing tag. As you can see there is still a `/` at the end of the tag. Notice the use of attributes as well.

```

```

## Basic HTML page structure

In this section we will take a look at the structure of an html file. We will discuss the basic tags need to have a well formed page for display on the web. From this point on we will be speaking in context of HTML5.

### HTML basic example

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>

</body>
</html>
```

Lets take a look at some of these tags/elements

#### **PRO TIP**

[https://developer.mozilla.org/en-US/docs/Web/HTML/Element/YOUR\\_TAG\\_HERE](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/YOUR_TAG_HERE) - Replacing `YOUR_TAG_HERE` with something like 'head', 'title', or any other tag/element you can think of should bring you to documentation on that specific tag/element

#### **Doctype**

`<!DOCTYPE html>` informs the browser which version of HTML will be used in this document. In this case we are declaring that we will be using HTML5. Doctype is special in that it's not really considered a tag but a declaration. What you really need to understand about this tag/declaration is that it should be present and the first line in your HTML file.

#### **html**

`<html>` is the tag that represents the root (top-level element) of an HTML document, so it is also referred to as the root element. All other elements must be descendants of this element. That being said, notice the matching closing `</html>` tag on the last line.

## **head**

`<head>` is that tag that contains descriptive information about your page like the pages title `<title>`. This is also where would include things like `<meta>` and style sheets. We will get into more of that stuff later in the curriculum. Notice the `<title>` tag has a closing tag, `</title>`, just like the `<html>` tag. I hope you are noticing the pattern.

## Recomended Reading

## **title**

`<title>` is the tag that defines the title of the document, shown in a browser's title bar or on the page's tab. It can only contain text, and any contained tags are ignored. It is followed by the closing `</title>` tag.

**body** `<body>` represents the content of an HTML document. There can be only one `<body>` element in an HTML document. This is where the things you want people to see would go. I am going to stop pointing out the closing tags. I think you got it by now.

# Get Organized



There are many opinions on how the files you create should be organized when building web applications. For the purposes of this course we will focus one of those opinions as outlined in this section. This will help things look familiar as we get into some of the later topics in the course.

## **Directory structure example**

*This is a basic example and can/will get more complicated as we progress through the curriculum. This is a great place to start and should get you through the next few lessons.*

```
root_directory
|
|   index.html
|   conat_us.html
|   ...
|
└── assets
    |
    └── images
        |
        |   image1.jpg
        |   image1.png
        |   ...
        |
        ├── javascripts
        |
        |   jquery.js
        |   slideshow.js
        |   ...
        |
        └── stylesheets
            |
            |   bootstrap.css
            |   custom.css
            |   ...
            |
```

## Whats going on here?

### **root\_directory**

Every project has a root directory. This is the top level folder where all files for the project will exist. This is where our .html files will live as well.

### **assets**

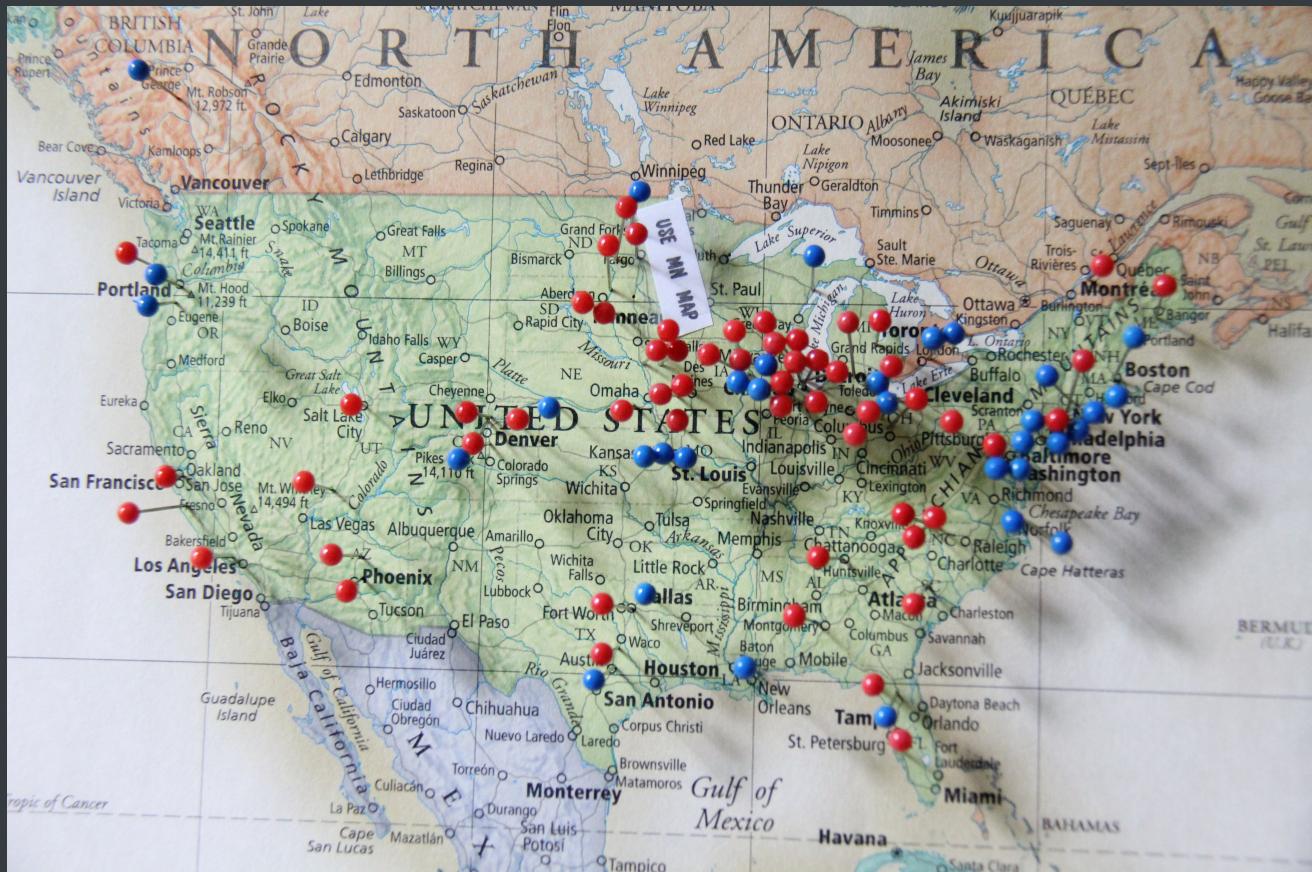
assets is a sub-directory under our root directory. This is the spot we save items we will use with our website. This can be images we intend to display, JavaScript we include for added functionality, or stylesheets we include to make things look cleaner.

## Popular HTML Tags



Let's take some time to cover some of the most common HTML tags you will use when creating web applications. This will be an interactive lesson so we can see the result of using these tags in a browser. We will build a simple travel page for our example.

## Travel Page Set Up



Before we move on, let's put together a page to try out all this HTML we're going to learn...

- Create a folder on your desktop call **html\_basics**
- Inside, create a file called **travel.html**
- Let's add the following HTML to the file so we have our basic page structure (next slide)

# Travel Page Set Up

In addition, let's go ahead and fill in the title for the site.

Now we're ready to explore more HTML tags!

```
<!DOCTYPE html>
<html>
<head>
    <title>My Travel Site</title>
</head>
<body>

</body>
</html>
```

## Headings

The HTML

—  
tags represent six levels of section headings.

is the largest and

is the smallest.

Let's add two headings to our site.

We will use the

and  
tags.

```
<h1>Welcome to my Travel Site</h1>
```

```
<h2>Here are some places I'd like to visit</h2>
```

Remember: this goes inside the !

## Paragraphs

The HTML `p` tag represents a paragraph of text.

Let's add a paragraph below our

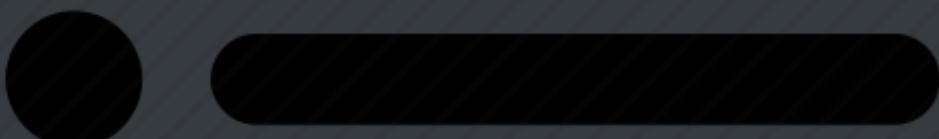
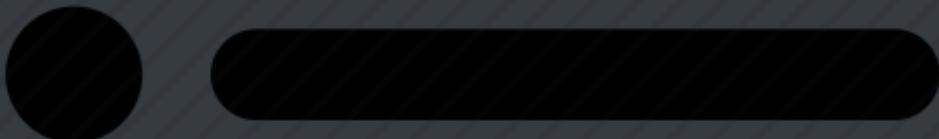
```
<h1>Welcome to my Travel Site</h1>
```

```
<p>This site is all about where I want to travel.</p>
```

```
<h2>Here are some places I'd like to visit</h2>
```

Lists

HTML has the ability to display lists. Lists are typically ordered or unordered. `<ol>` represents an ordered and `<ul>` represents an unordered list. Ordered lists are numbered and unordered lists use some type of bullet.



List items are represented by the `<li>` tag in both ordered and unordered lists.

# Lists

Go ahead and add either an ordered

or unordered

list of places you would like to travel. Add the list below our

```
<h1>Welcome to my Travel Site</h1>
<p>This site is all about where I want to travel.</p>

<h2>Here are some places I'd like to visit</h2>
<ul>
    <li>Spain</li>
    <li>Italy</li>
    <li>Aruba</li>
    <li>Alaska</li>
</ul>
```

## Links

Links are represented by the anchor tag ( `<a>` ). The anchor tag creates a hyperlink. A hyperlink is a link from one location or file, typically activated by clicking on a highlighted word or image on the screen. A hyperlink can take the user to other web pages, files, locations within the same page, email addresses, or any other URL. The text or element between the opening and closing `<a>` tag is what becomes the link.

### Basic anatomy of the anchor tag `<a>`

There are several attributes for an `<a>` tag. Let's go over a few of the common ones.

- `href` – The URL or destination hyperlink points to.
- `target` – Specifies where to display the linked URL. Below are the most common values used.
  - `_self`: Load the URL into the same browser window as the current one. This is the default behavior if the `target` attribute is not used.
  - `_blank`: Load the URL into a new browser window or tab based on how the user's browser is configured.

[Further Reading on the `<a>` tag](#)

### Basic anchor examples

```
<!-- external hyperlink opens in a new window -->
<a href="http://www.google.com" target="_blank">google.com</a>

<!-- external link opens in the same window -->
<a href="http://www.google.com">google.com</a>
```

```

<!-- Email link that opens users default email client -->
<a href="mailto:user@example.com">Email User</a>

<!-- link to local another file on the same site -->
<a href="my_html_file.html">go to my_html_file</a>

<!-- links to element on this page with id="page-section" -->
<a href="#page-section">Link to page section</a>

<!-- External link that opens in a new window/tab using an image instead of text -->
<a href="https://developer.mozilla.org/en-US/" target="_blank">
  
</a>

```

## Images

Images are placed in our HTML document using the `<img>` tag. The image tag is a self closing tag and **Doesn't** require a matching closing tag.

### Basic anatomy of the anchor tag `<img>`

There several attributes for an `<img>` tag. Lets go over a few of the common ones.

- `src` - The image URL. This attribute is mandatory for the `<img>` element.
- `alt` - This attribute defines description of the image. Users will see this text displayed if the image URL is wrong, the image is not in one of the supported formats, or if the image is not yet downloaded.
- `height` - The height of the image in pixels
- `width` - The width of the image in pixels

### *Discussion Topics*

*What happens to the image if only width or height is specified?*

*Further Reading on the `<img>` tag*

### Basic `<img>` examples

```

<!-- image with alt take providing a description -->


<!-- image with width and height set -->


<!-- Image used as a link -->
<a href="https://developer.mozilla.org/en-US/" target="_blank">
  
</a>

```

# Travel Page: Images

Lets add images of the flags that represent each place you would like to visit. These images came from Wikipedia

and the width has been set to 50.

```

<li>
  
  <a href="http://www.spain.info/en_US/" target="_blank">Spain</a>
</li>
<li>
  
  <a href="http://www.italia.it/en/home.html" target="_blank">Italy</a>
</li>
<li>
  
  <a href="http://www.arubatourism.com/" target="_blank">Aruba</a>
</li>
<li>
  
  <a href="https://www.travelalaska.com/" target="_blank">Alaska</a>
</li>

```

## Tables

The HTML `<table>` element represents tabular data. Think of the way a spreadsheet presents data. Tables allow you to arrange rows and columns to present data to the user. We are only going to scratch the surface on tables in this example so please explore the further reading link provided.

### Basic anatomy of a `<table>`

There several nested tags for the `<table>` tag that help build the rows and columns. Lets go over a few of the common ones.

- `<br>` – represents the start of a new row.
- `<td>` – represents the start of a new column within a row.
- `<th>` – represents a special header column. `<th>` should only be used in the first column in place of the standard `<td>`. Headers will appear bold.

*Further Reading on the `<table>` tag*

### Basic `<table>` example

```

<!-- basic table with columb headers -->
<table>
  <tr>
    <th>First name</th>
    <th>Last name</th>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>Doe</td>
  </tr>
</table>

```

# Travel Page: Tables

```
<!-- This is where our list ended -->

<table>
  <tr>
    <th>Destination</th>
    <th>Capital</th>
    <th>Official Language</th>
  </tr>
  <tr>
    <td>Spain</td>
    <td>Madrid</td>
    <td>Spanish</td>
  </tr>
  <tr>
    <td>Italy</td>
    <td>Rome</td>
    <td>Italian</td>
  </tr>
  <tr>
    <td>Aruba</td>
    <td>Oranjestad</td>
    <td>Dutch</td>
  </tr>
  <tr>
    <td>Alaska</td>
    <td>Juneau</td>
    <td>n/a</td>
  </tr>
</table>
```

## Divs

The HTML `<div>` element is a generic container for content flow. We are not going to use this in our travel site but you will see it as we go through the curriculum. It is mainly used to group elements for styling and positioning.

The `<div>` is a block-level element. Browsers typically display the block-level element with a newline both before and after the element.

[Further reading on block-level elements](#)

- Basic `<div>` example

```
<div>
  <p>Any kind of content here. Such as
  <p> or <table> tags. You name it!</p>
</div>
```

## Spans

The HTML `<span>` element is a generic container for content flow. Similar to the `<div>`, we are not going to use this in our travel site but you will see it as we go through the curriculum. It is mainly used for styling purposes.

The `<span>` is an inline element. An inline element occupies only the space bounded by the tags that define the inline element. Basically that means you can wrap it around a word or something similar and it won't add new lines above and below the span. Further reading on [inline tags](#)

- Basic `<span>` example

```
<p>Any kind of <span>content</span> here.</p>
```

## Homework

For homework we are going to create a simple page that displays your 5 favorite restaurants in your favorite city. You should use all the tags we used to create our travel site.

### Easy Challenge

[Reference for challenge](#)

Find some tags using the reference above to style some of the text on the page. This will help you get familiar with reading documentation. Learning on your own is a big part of being a successful developer.

### Medium Challenge

Make one of your favorite places to eat your own kitchen. Create another HTML page for your kitchen (e.g. `my_kitchen.html`). Make sure you link to your new page from the main page. Also make sure you can get back to your main page from your new page. This page should NOT open in a new window.

*[Link to an internal page example](#)*

```
<a href="my_kitchen.html">My Kitchen</a>
```

### Hard Challenge

[Reference for challenge](#)

Add a contact form to your page. This form does not need to function. Only worry about getting it to display on your page. Collect the user's name, email, and comment.

Copyright © 2013 – 2017 Tech Talent South. All rights reserved.