HTML/CSS

Introduction to markup language and stylesheet language.

Preface

Will write more on this. For now, know that HTML/CSS provides information on how to examine and edit source code for HTML and CSS on BookStack and other websites.

Introduction to HTML

Tags and Elements

HyperText Markup Language (HTML) labels and structures web content. The HyperText portion refers to links that connect web pages to other resources, while Markup Language mainly refers to the way content on a web page is identified. This form of identification is known as a **tag** and are distinguished by `<` and `>` with at least an *element name* in-between those characters. Personally, I think calling it the element name creates confusion, so from here on I'll refer to it as the tag's name. For most situations, a tag by itself isn't very useful. To put content on a webpage, there usually needs to be an *opening tag* and *closing tag* with a string of characters in between. The following is an example:

`<h1> Joshua Trees </h1>`

`<h1> ... </h1>`

This shows the opening tag `<h1>` and closing tag `</h1>` acting as a container for the content `Joshua Trees`. The opening tag, closing tag, and content is collectively known as an **element**. Notice how the closing tag has a `/` before the tag's name; all closing tags require this character. Note, the tag's name is case insensitive, however, it is universally agreed upon that developers keep it lowercase. This particular line of code is known as a **heading element**, which when compiled increase the size of the characters to make a heading, like so:

Joshua Trees

While a majority of elements follow this structure, there are a few exempted from this rule. Take for example, the **image element**

`<img src = "Images/Joshua Trees.jpg" />`

`<img src = "location_of_image_in_directory" />`

The code above displays the tag's name `img` followed by an attribute `src` and a value `Images/Joshua Trees.jpg` with a closing `/`. An **attribute** is a predefined variable that assigns a value to it in the form of strings, numbers, IDs, and classes (more on those in later chapters).

Notice that there is no second tab or content, these are categorized as **empty elements** and are only used with specific types of elements. Keep in mind that attributes can only found in empty elements and in the opening tags of regular elements. As you may have guessed, attributes affect the behavior of an element. For example, the code above was created in a local html file, and it displays an image of Joshua trees which the source `src` of that image is located in a directory.

Warning, you may omit the double quotation marks, but it is considered good practice to use them. You may write these elements on a HTML document, but they won't populate your webpage unless you use the basic outline of an HTML document.

The Basic Outline of an HTML Element

The following block of code shows the basic structure of an HTML document, along with the previous examples we used to show how it looks like. Notice how the elements are formatted in a nested like structure. We'll examine each line in the source code to understand each element.

```
<!DOCTYPE html>
<html>
   <head>
   </head>
   <body>
      <h1>Joshua Trees</h1>
      <img src = "Images/Joshua Trees.jpg" />
   </body>
```

```
</html>
```

Joshua Trees

Doctype Empty Element

`<! DOCTYPE html>`

`<! DOCTYPE version_of_html>`

This line of code states which version of HTML you are using, in this case we are using HTML5 which is written as `html` in the doctype empty element. A quick google search will help you how to declare older versions of HTML.

HTML Element

```
<html>

   ...

</html>
```

The HTML element lets the browser know that the code nested within the HTML element should be interpreted as HTML.

Head Element

```
<head>
   ...
</head>
```

The head element provides information about the website, like who's the creator what's the label and so forth.

Body Element

```
<body>
   ...
</body>
```

Lastly, the body element capsulates the code that will be display on the web browser.

HTML: Special Elements and Layout Elements