HTML5 and CSS3

Chapter 1:
Web Page Building Blocks

At the end of this topic you will learn:

- A Basic HTML Page
- Semantic HTML: Markup with Meaning
- Markup: Elements, Attributes, and Values
- A Web Page's Text Content
- Links, Images, and Other Non-Text Content
- File Names
- URLs

Introduction

- A web page is primarily made up of three components:
  - **Text context**
    - *Text context that appears on the page to inform the visitor*
  - **References to other files**
    - *Load items that links to other HTML pages, assets and style sheets.*
  - **Markup**
    - *HTML elements which describe text content and make the reference work.*

Semantic HTML: Markup with Meaning

- HTML is the **information** that called **markup** which describe the meaning of the content called **semantic**.
- HTML focus on semantic describe the **content's meaning**, not it display.
- HTML5 goes further; it eliminates some presentational elements and focus on **semantic value**.
- The semantic **significant** because:

  - Improved accessibility and interoperability
  - Improved search engine optimization (SEO)
  - Lighter code and faster pages
  - Easier code maintenance and styling

Markup: Elements, Attributes, and Values

- HTML has three principal markup component include **elements**, **attributes** and **values**.

- ELEMENTS – Little labels that describe the difference parts of a Web page. It will be **non-empty** and **empty** elements.

- ATTRIBUTES and VALUE – Attributes contain **information** about the content in the document thus contain **value** to produce the result.

Markup: Elements

- Elements can be **non-empty** which contain text and/or other elements, or they can be **empty**.

- A **non-empty element** consists of an **opening tag** *(the element's name and attributes, if any, enclosed in less than or greater than signs),* **the content**, and a **closing tag***(a forward slash followed by the element's name, again enclosed in greater than and less than signs)*

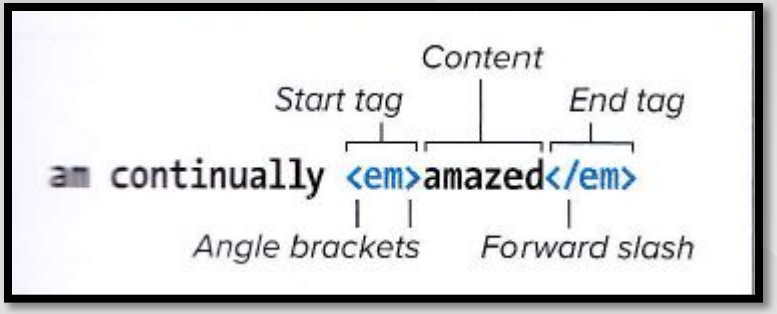


Figure 1: Non-empty element show the start tag and end tag surround the text the element describes. The word "amazed" is emphasized using element

Markup: Elements

- An **empty** element looks like a **combination opening and closing tag**, with an initial less than sign, the element's name followed by any attributes it may have, a space, a forward slash, and the final greater than sign.

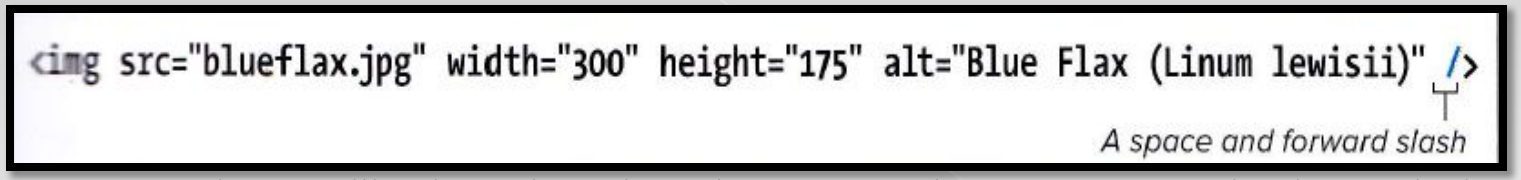


Figure 2: Empty elements, like ”img” shown here, do not surround any text content. They have a single tag which serves both to open and close the element. In HTML the final slash is optional. The > is required to complete the element

Markup: Attributes and Values

- Attributes contain information about the **data in the document**, as opposed to being that data itself . In HTML5, an attributes value's may optionally be enclosed in quotation marks as follow:

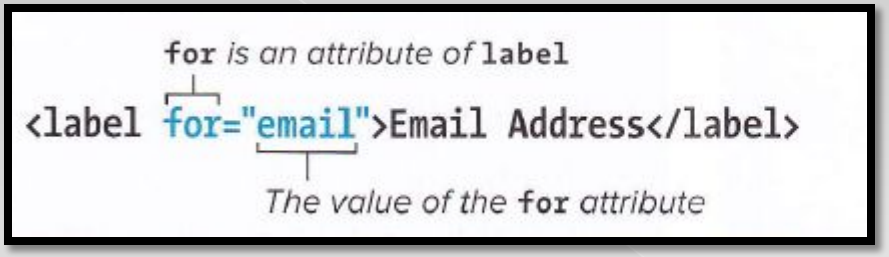


Figure 3: The element with a simple attribute-value pair. Attributes are always located inside an element's opening tag.

Markup: Attributes and Values

- Some elements, like ”img” shown here, can take one or more attributes, each with its own value. The order is not important. Separate each attribute-value pair from the next with a space.

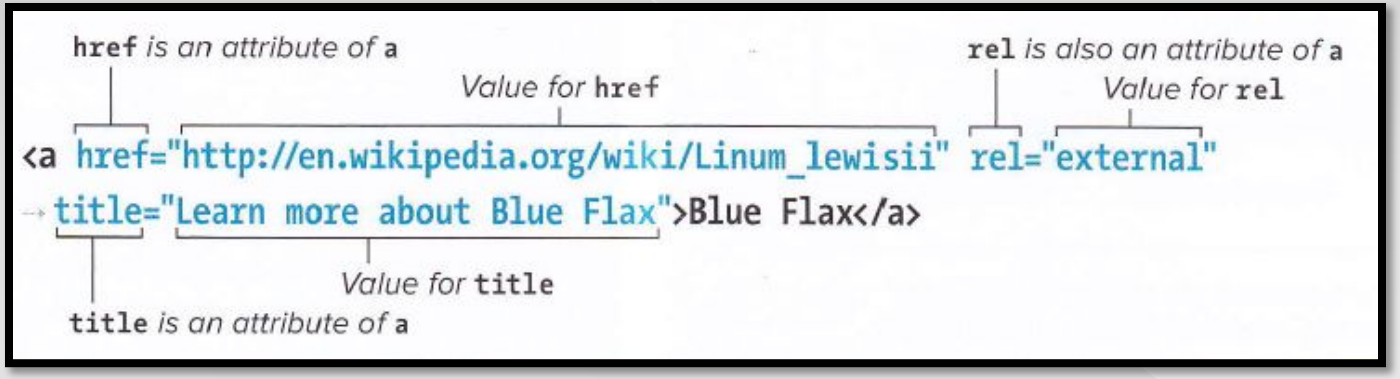


Figure 4

Markup: Attributes and Values

- Some attributes can accept any value at all, others are more limited. In figure 5, some attributes only accepts specific values. For example, the **media** attribute in the **link** element can be set to **all, screen** or **print**, among others, but you can't just make up a value for it.

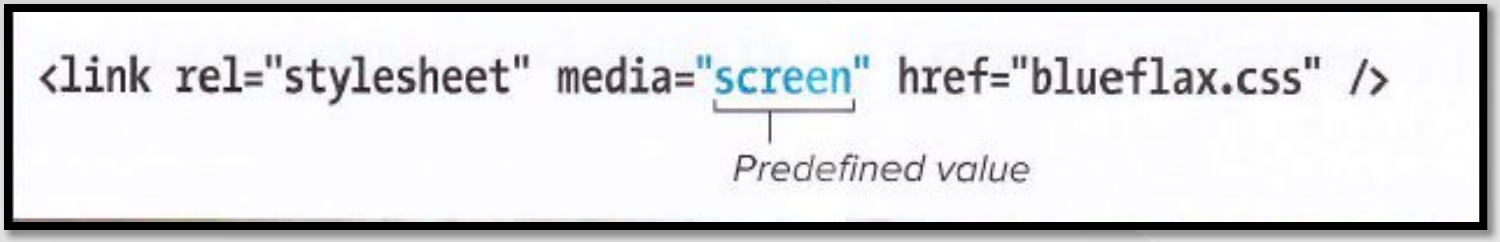


Figure 5

Parents and Children

- Is the elements contain another
- Any elements contain in the child element are considered descendants of the outer, parent element.

```
<article>
   <h1>The Ephemeral Blue Flax</h1>
   <img src="blueflax.jpg"... />
   <p>... continually <em>amazed</em> ... delicate <a ...>Blue Flax</a> ...</p>
</article>
```

Figure 7: The example of parent and children elements

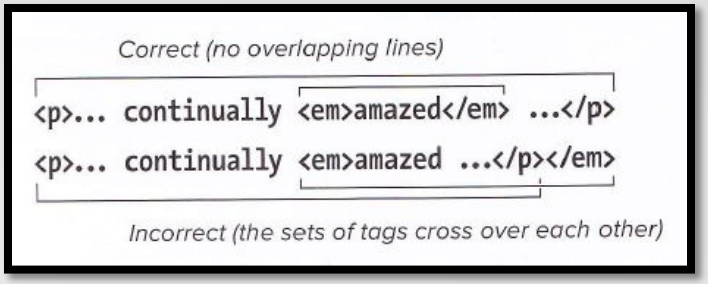


Figure 8: Elements must be properly nested. If **"p"** and **"em"** is open it must close **"em"** before **"p"**

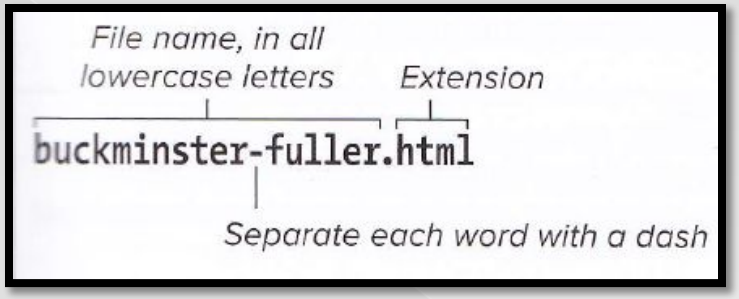
File Names

- Web page has a file names identifies itself to users, users visitors, and to users visitors' Web browser.

- Help users organize their files, make it easier for their visitors to find and access the pages, and ensure that their browsers view the pages correctly.

- File names must be assign with three concepts and rules include:

  - Use **Lowercase** File Names
  - Separate Words with a **Dash**
  - Use the Proper **Extension**



URLs

- URLs is stand for **Uniform Resource Locator** is a fancy name for address.
- Contain **information** about where a file is and what a browser should do with it.
- First part of URL called **scheme.** It tells the browser how to deal with a file that is about to open. *Example:* ***http (Hypertext Transfer Protocol), ftp (file transfer protocol, mailto and file.***
- Second part of URL called **server name** followed by **path** that leads to the file and the name itself.

Figure 10: URLs

Absolute and Relative URLs

- URLs can be either **absolute** and **relative.**
- Absolute URL show the entire path to the file, including the **scheme**, the **server name**, the complete **path** and the **file name** itself.
- An absolute URL is analogues to a complete **street address** including name, street and number, city, state, zip code and country.
- Relative URL describes the **location** of the desired file with reference to the location of the file that contains the URL reference itself.
- The relative URL for a file that is in the same directory as the current page (that is, the one containing the URL in question) is simply the file name and extension.

Absolute and Relative URLs

Absolute URLs vs. Relative URLs

File name	Absolute URL (can be used anywhere)	Relative URL (only works in youarehere.html)
index.html	www.site.com/web/index.html	index.html
image.gif	www.site.com/web/images/image.gif	images/image.gif
data.html	www.site.com/info/data.html	../info/data.html
homepg.html	www.remote.com/pub/homepg.html	*(none: use absolute)*
info.html	www.remote.com/bcn/info.html	*(none: use absolute)*

Figure 11: Absolute vs Relative URL

THE END

THANK YOU FOR YOUR ATTENTION

Q &A