



HAZZA INSTITUTE
OF TECHNOLOGY



MERN – ES6 + React

Module 01 : HTML Basics

Outline

Module 01

- ▶ HTML Basics
- ▶ Explaining Tagging concept
- ▶ List creation and display
- ▶ Divs and Spans creation
- ▶ Attributes and information display

HTML Basics

Client-side development

► What is HTML?

- HTML stands for **Hyper Text Markup Language**
- **HTML is not a programming language**
- HTML describes the **structure** of a Web page
- HTML consists of a series of **elements**
- HTML elements tell the browser **how to display the content**

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

  <h1>My First Heading</h1>
  <p>My first paragraph.</p>

</body>
</html>
```

HTML Basics

Client-side development

► What is an HTML Element?

- An HTML element is defined by a **start tag**, some **content**, and an **end tag**:
- The HTML element is everything from the start tag to the end tag:

`<tagname>` Content goes here... `</tagname>`

Start tag	Element content	End tag
<code><h1></code>	My First Heading	<code></h1></code>
<code><p></code>	My first paragraph.	<code></p></code>
<code>
</code>	<i>none</i>	<i>none</i>

Note: Some HTML elements have no content (like the `
` element). These elements are called empty elements. Empty elements do not have an end tag!

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

  <h1>My First Heading</h1>
  <p>My first paragraph.</p>

</body>
</html>
```

HTML Basics

Client-side development

► What are an HTML Attributes ?

- **All** HTML elements can have attributes
- Attributes provide **additional information** about elements
- Attributes are always specified in the **start tag**
- Attributes usually come in name/value pairs like: **name="value"**

```

```

Examples

- The **href** attribute of **<a>** specifies the URL of the page the link goes to
- The **src** attribute of **** specifies the path to the image to be displayed
- The **width** and **height** attributes of **** provide size information for images
- The **style** attribute is used to add styles to an element, such as color, font, size, and more
- The **alt** attribute of **** provides an alternate text for an image

Evolution of HTML

Browser WAR?

Year	HTML Version
1993	HTML 1.0 - Developed by Tim Berners-Lee to link document
1995	HTML 2.0 - Developed by Internet Engineering Task Force RFC to include stylized text and tables
1996	CSS 1
1997	HTML 3.2 – Developed by W3C and included browser specific features
1997	HTML 4.0 – A move back to normalizing the pages across platforms.
1998	CSS 2
1999	HTML 4.01 – Introduced different document types
2012	HTML 5 - Back to HTML plus multimedia and semantic tags

Creating and Editing Your Files

1. **Decide how you will organize your files**
2. **Decide on a naming convention**
 - dash-names, CamelCase
 - No spaces, Consistent capitalization
3. **Decide on an editor**
 - Windows (*Notepad*, Notepad++, Sublime, **VS Code**)
 - Mac (*TextEdit*, TextWrangler, Sublime, **VS Code**)

Getting Started

- 1. Open your editor**
- 2. Select Save or Save As and name your file.**
You may need to create a new folder first
- 3. Add Doctype, head, and body tags**
- 4. Save File (Ctrl-S or Command-S)**
- 5. Open in browser**

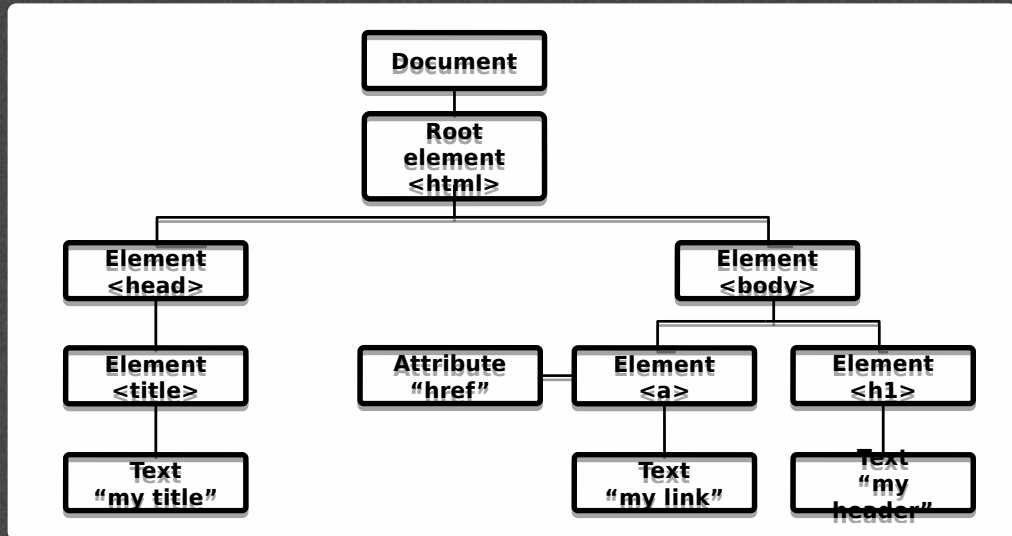
Document Object Model

Writing clean code

The Document Object Model (DOM)

- **Basis of HTML5 is “*New features should be based on HTML, CSS, the DOM, and JavaScript...*”**
- **DOM provides common tree-like structure that all pages should follow**
- **Computer Scientists love trees (the mathematical kind) because you can test them.**

HTML is built on the DOM



Adapted from w3Schools.com

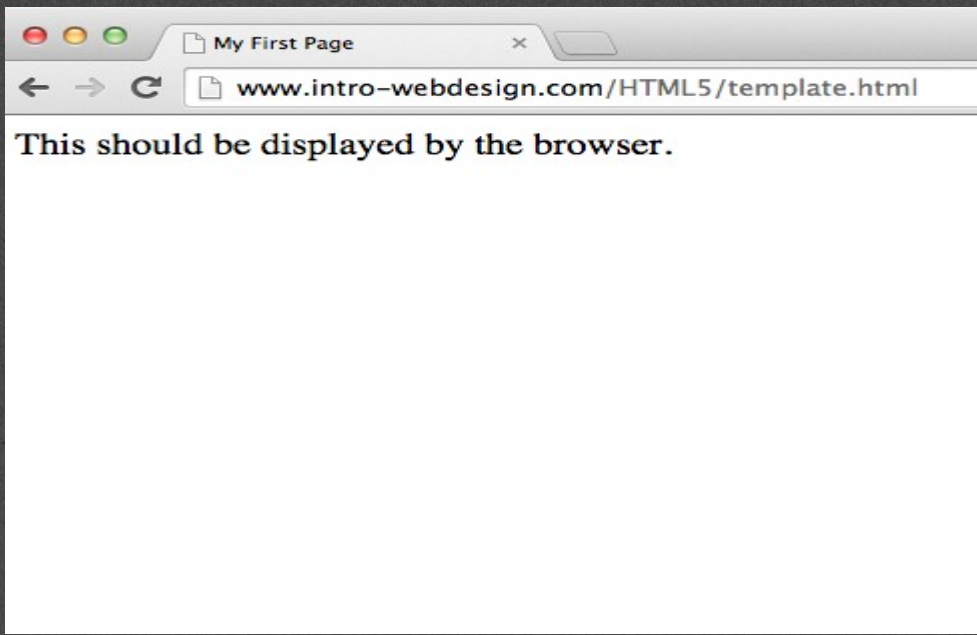
Three parts of a well-formed document

- **Doctype**
 - Version of HTML that you will be using
- **Head**
 - Metadata
- **Body**
 - Displayable content

Example

- **Example: template.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My First Page</title>
</head>
<body>
  This should be displayed by the browser.
</body>
</html>
```



Display

- One of the most important attributes of an element is its display. The two most common are **block** and **inline**
 - **block (can take width and height)**
 - *Newline is inserted before and after, e.g. it “Takes up” whole width*
 - **inline (can not take width and height)**
 - *Only uses as much space as needed to contain the element.*

Common Tags

- **Headings (block)**
 - `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`
 - **These tags have `syntax` and `semantics`**
- **Paragraphs (block)**
 - `<p> </p>`
 - **Should only contain inline elements**
- **Divs (block)**
 - `<div>...</div>`
 - **Generic section that is larger than a paragraph**

More tags

- **Ordered lists**

```
<ol>
  <li> Item One </li>
  <li> Item Two </li>
</ol>
```

```
<li> Item One </li>
<li> Item Two </li>
</ul>
```

- **Unordered lists**

```
<ul>
```

```
<br>
```

Attributes

- **Attributes provide additional information about an element**
- **Always specified in the start tag**
- **Attributes come in name/value pairs**

Images

- **Images (inline)**
 - ``
- **Images rarely work the first time**
 - **Show a broken link, too big, too small, etc.**
- **Save yourself heartache and size/carefully name your picture before you use it.**

More Attributes

- **As you learn the tags, you learn their specific attributes. Some apply to any tag**
 - **class - applies special properties to groups of elements**
 - **id - specifies a unique id to one element on the page**
 - **style - specifies a certain visual style (avoid this one!!!)**
 - **accesskey - a shortcut key to activate an element**
 - **tabindex - the order elements will come into focus using the tab key.**

Semantic HTML5 Tags

Making the most of the new tags

How to Design

- **The most important step in web design is the *design*.**
- **You need a clear picture of what you want to create, before you can begin coding.**
- **Think wireframes !!!**

How to Design

`<header>`

`<section>`

`<article>`

`<aside>`

`<footer>`

Using Semantic Tags

- In the beginning (insert dramatic music of your choice...) there was `div`
- `<div>` was a way to group related content together
- Divs almost always had special classes/ids associated with them
 - `<div class = "header">...</div>`
 - `<div class = "section">...</div>`
 - `<div class = "footer">...</div>`

<header>

- A group of introductory or navigational aids: title, navigation links, etc.

```
<header>  
  <h1>This is the Title</h1>  
  <h2>The author is Colleen</h2>  
</header>
```

- Not to be confused with <head> or the different headings.

<nav>

- A section of the page that links to other pages or to parts within the page.

```
<nav>  
  <ul>  
    <li><a href="#overview">Overview</a></li>  
    <li><a href="#history">History</a></li>  
    <li><a href="#development">Development</a>  
  </ul>  
</nav>
```

- Often found in the <header> tag

<footer>

- A section that contains info such as copyright data, related documents, and links to social media

```
<footer>  
  &copy; 2015 by Colleen van Lent<br>  
  <a href="http://www.intro-webdesign.com/HTML5">Introduction to  
  HTML5</a>  
</footer>
```

- Typically at the bottom of the page, but not required.

<figure>

- More semantics than . Can include:

```
<figure>  
    
  <figcaption> A sunset over Lake Erie. Taken in  
  Ashtabula Ohio</figcaption>  
</figure>
```

Other New Tags

- **Structural Elements**
 - article, aside, main, menuitem, summary, section
- **Form Elements**
 - datalist, keygen, output
- **Input Types**
 - color, date, email, list
- **Graphics Elements**
 - canvas, svg
- **Media Elements**
 - audio, embed, source, track, video

Hyperlinks

Creating a linked document

Links

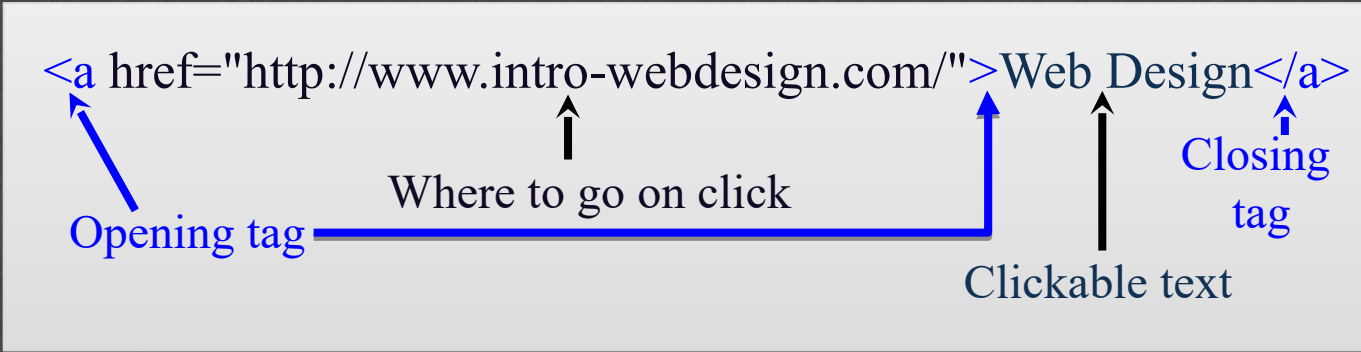
- **Links are what make the Web a web.**
- **The interlinked nature of the web leads to the “knowledge” that search engines appear to have.**

Anchor links

```
<a href="http://www.umich.edu">University of Michigan</a>
```

- The `<a>` tag stands for *anchor link*
- Needs a hyper-reference AND content
 - **href**: reference to location of new content
 - **content**: the “clickable” part (text or image)

Absolute reference



Relative References

```
<a href="page2.html">Second Page</a>
```



Link to a local file in the same folder

```
<a href="docs/page2.html">Second Page</a>
```



Link to a local file in a different folder called “docs”

```
<a href="#history">History section</a>
```



Link to a different location in the same file

Absolute vs Relative

- **When would you use absolute links?**
- **Are there any benefits to using local links?**
- **Your links should NEVER have folders that are specific to your computer**

C:\page2.html



Using Images as the Link

- The “clickable” component doesn’t have to be text.

```
<a href="http://www.redcross.org">  
  <img src = "imgs/redcross-logo.png"  
  alt = "Red Cross logo"/></a>
```

```
<a href="http://www.redcross.org">  
  <img src = "http://www.redcross.org/images/redcross-logo.png"  
  alt = "Red Cross logo"/>  
</a>
```

Targets

- **Anchors can take a target attribute**
 - **_self** - default action
 - **_blank** - open in new tab or window
 - **_top** and **_parent**

Summary

Module 01

- ▶ HTML Basics
- ▶ Explaining Tagging concept
- ▶ List creation and display
- ▶ Divs and Spans creation
- ▶ Attributes and information display