Introduction to HTML

Dr. Vivek Nallur
vivek.nallur@ucd.ie

University College Dublin

What HTML lets you do

- Insert images using the **<img>** tag
- Create links with the **<a>** tag
- Create lists with the **<ul>, ** and **<li>** tags
- Create headings with **<H1>, <H2>, …, <H6>**
- Define metatdata with **<meta>** tag
- And much more…

Elements

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

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

QUICK TOUR OF HTML

Why a quick tour?

- HTML5 contains many structural and presentation elements – too many to completely cover.
- Rather than comprehensively cover all these elements, this presentation will provide a quick overview of the most common elements.
- You will discover other useful elements during independent study and assignments.

Headings

`<h1>`, `<h2>`, `<h3>`, etc

HTML provides six levels of heading (**h1, h2, h3**, …), with the higher heading number indicating a heading of less importance.

Headings are an essential way for document authors use to show their readers the structure of the document.

Headings

The browser has its own default styling for each heading level.

However, these are easily modified and customized via CSS.

```
<html>
<body>
    <h1>This heading is of size H1</h1>
    <h2>This heading is of size H2</h2>
    <h3>This is slightly smaller: H3</h3>
    <h4> Even smaller: H4 </h4>
    <h5>Still smaller: H5 </h5>
    <h6>Smallest heading: H6 </h6>
</body>
</html>
```

headings.html

Headings

Be semantically accurate

In practice, specify a heading level that is semantically accurate.

Do not choose a heading level because of its default presentation

- e.g., choosing <h3> because you want your text to be bold and 16pt

Rather, choose the heading level because it is appropriate

- e.g., choosing <h3> because it is a third level heading and not a primary or secondary heading

Paragraphs

`<p>`

Paragraphs are the most basic unit of text in an HTML document.

Notice that the **<p>** tag is a container and can contain HTML and other **inline HTML elements**

inline HTML elements refers to HTML elements that do not cause a paragraph break but are part of the regular “flow” of the text.

An Example

```
<html>
    <body>
        <h3>This is a small heading</h3>
        <p>And this is a paragraph. Most
        text inside an HTML document
        will be inside paragraphs.
        Many other HTML elements will
        also be inside the paragraph tag.
        </p>
    </body>
</html>
```

paragraph.html

Divisions

This ** ** tag is also a container element and is used to create a logical grouping of content

- The element has no intrinsic presentation.
- It is frequently used in contemporary CSS-based layouts to mark out sections.

div.html

Empty Tags

Most HTML elements look like `<body><p>…</p></body>`

But some elements cannot have other elements nested inside them. They are called empty elements.

`<br />` ➔ Inserts a line break

`<hr />` ➔ Inserts a horizontal line

`<img>` ➔ Inserts an image

`<input>` ➔ Inserts an input control like text box, checkbox, etc.

`<link>` ➔ Defines relationships between documents (e.g., to load CSS)

`emptytag.html`

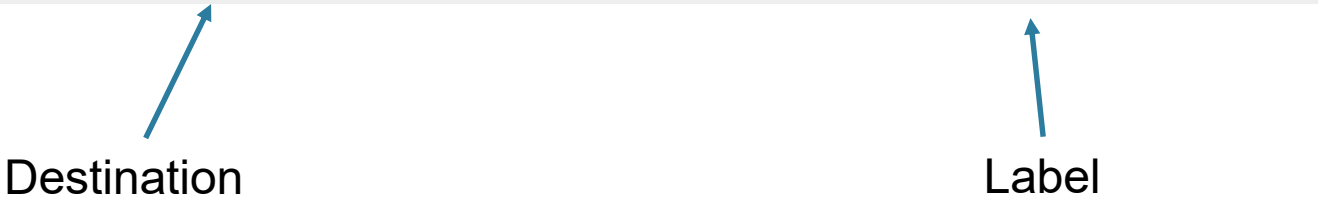
Links

<a>

Links are created using the **<a>** element (the “a” stands for anchor).

A link has two main parts: the destination and the label.

```
<a href="http://www.yoururl.com">Link text</a>
```



link.html

Types of Links

You can use the anchor element to create a wide range of links:

- Links to external sites (or to individual resources such as images or movies on an external site).
- Links to other pages or resources within the current site.
- Links to other places within the current page.
- Links to particular locations on another page.
- Links that are instructions to the browser to start the user's email program.
- Links that are instructions to the browser to execute a Javascript function.

Different link destinations

```
<html>
    <body>
    <p>This piece of text has a
    link to <a href="http://localhost/headings.html">headings page</a>


    <p>This is a relative link to <a href="emptytag.html">the
    tags page</a>

    <p>This is a link to my <a href="mailto:vivek.nallur@ucd.ie">email
    address</a>

    </body>
</html>
```

differentlinks.html

URL Absolute Referencing

For external resources

When referencing a page or resource on an external site, a full **absolute reference** is required: that is,

- the protocol (typically, http://),
- the domain name,
- any paths, and then finally
- the file name of the desired resource.

URL Relative Referencing

An essential skill

We also need to be able to successfully reference files within our site.

This requires learning the syntax for so-called **relative referencing**.

When referencing a resource that is on the same server as your HTML document, then you can use briefer relative referencing. If the URL does not include the “http://” then the browser will request the current server for the file.

URL Relative Referencing

If all the resources for the site reside within the same **directory** (also referred to as a **folder**), then you can reference those other resources simply via their filename.

However, most real-world sites contain too many files to put them all within a single directory.

For these situations, a relative pathname is required along with the filename.

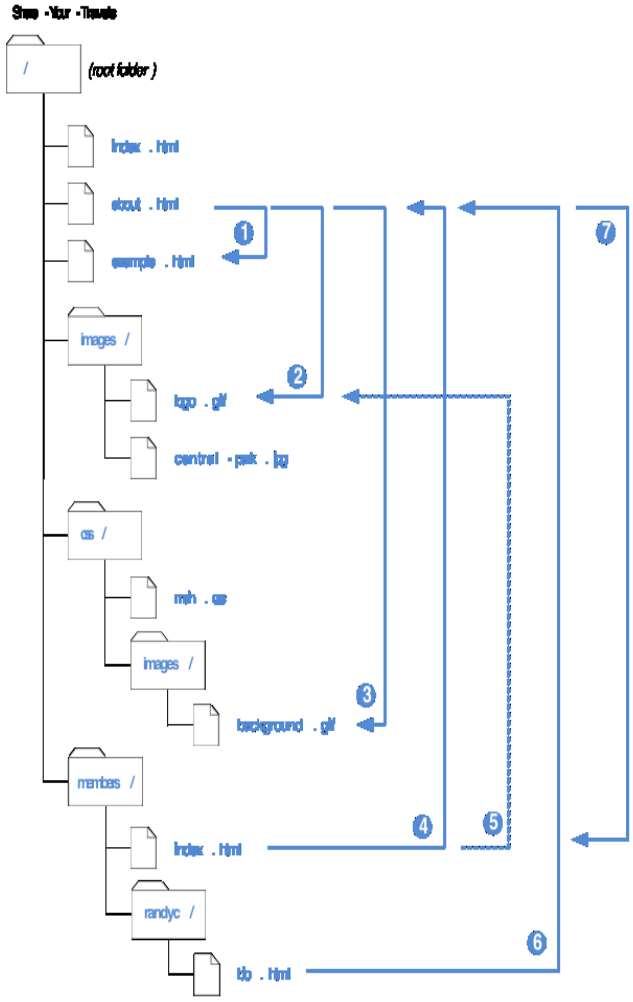
The **pathname** tells the browser where to locate the file on the server.

Pathnames

Pathnames on the web follow Unix conventions.

- Forward slashes (“/”) are used to separate directory names from each other and from file names.
- Double-periods (“..”) are used to reference a directory “above” the current one in the directory tree.

URL Relative Referencing



Relative Link Type	Example
Same Directory To link to a file within the same folder, simply use the file name.	To link to example.html from about.html (in Figure 2.18), use: `<a href="example.html">`
Child Directory To link to a file within a subdirectory, use the name of the subdirectory and a slash before the file name.	To link to logo.gif from about.html, use: `<a href="images/logo.gif">`
Grandchild/Descendant Directory To link to a file that is multiple subdirectories *below* the current one, construct the full path by including each subdirectory name (separated by slashes) before the file name.	To link to background.gif from about.html, use: `<a href="css/images/background.gif">`

Parent/Ancestor Directory
Use "../" to reference a folder *above* the current one. If trying to reference a file several levels above the current one, simply string together multiple "../".
	To link to about.html from index.html in members, use:
	`<a href="../about.html">`
	To link to about.html from bio.html, use:
	`<a href="../../about.html">`

URL Relative Referencing

Sibling Directory

Use "../"to move up to the appropriate level, and then use the same technique as for child or grandchild directories.

To link to logo.gif from index.html in members, use:

```
<a href="../images/about.html">
```

To link to background.gif from bio.html, use:

```
<a href="../../css/images/background.gif">
```

Root Reference

An alternative approach for ancestor and sibling references is to use the so-called **root reference** approach. In this approach, begin the reference with the root reference (the "/") and then use the same technique as for child or grandchild directories.

Note that these will only work on the server! That is, they will not work when you test it out on your local machine.

To link to about.html from bio.html, use:

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

To link to background.gif from bio.html, use:

```
<a href="/css/images/background.gif">
```

Default Document

Web servers allow references to directory names without file names. In such a case, the web server will serve the default document, which is usually a file called index.html (apache) or default.html (IIS). **Again, this will only generally work on the web server**.

To link to index.html in members from about.html, use either:

```
<a href="members">
```

Or

```
<a href="/members">
```

Inline Text Elements

Do not disrupt the flow

Inline elements do not disrupt the flow of text (i.e., cause a line break).

HTML5 defines over 30 of these elements.

e.g., <a>,
, ,

Images

While the **<img>** tag is the oldest method for displaying an image, it is not the only way.

For purely decorative images, such as background gradients and patterns, logos, border art, and so on, it makes semantic sense to keep such images out of the markup and in CSS where they more rightly belong.

But when the images are content, such as in the images in a gallery or the image of a product in a product details page, then the <img> tag is the semantically appropriate approach.

Images

```
<html>
    <body>
    <p>This is a piece of text which has an image inside it. In this
    case, it happens to be my photo.
        <img src="images/viveknallur-photo.jpg" alt="Vivek's Photo"
        title="Very serious Vivek" width="60px" height="60px" />
    </p>
    </body>
</html>
```

NOTE: You can use both absolute/relative referencing for the image

image.html

Figure and Figure Captions

```
<html>
    <body>
    <p>This is a photo of Kennedys, which is a fairly famous
    pub in Dublin
        <br />
        <figure>
            <img src="images/kennedys.jpg" alt="Kennedys Pub"
            title="Kennedys Pub" width="450px" height="450px" />
            <figcaption>Kennedys Pub in the city centre</figcaption>
        </figure>
    </p>
    </body>
</html>
```

figure.html

Figure and Figure Captions

Note however …

The **<figure>** element should **not** be used to wrap every image.

For instance, it makes no sense to wrap the site logo or non-essential images such as banner ads and graphical embellishments within <figure> elements.

Instead, only use the <figure> element for circumstances where the image (or other content) has a caption and where the figure is essential to the content but its position on the page is relatively unimportant.

Attributes

- An attribute provides additional information about the specific element
- It is always specified in the opening tag

```
<img src="images/kennedys.jpg" alt="Kennedys Pub"
title="Kennedys Pub" width="450px" height="450px" />
```

attributes

```
<p>This is a relative link to <a href="emptytag.html">the
tags page</a>
```

Lists

HTML provides three types of lists

Unordered lists. Collections of items in no particular order; these are by default rendered by the browser as a bulleted list.

Ordered lists. Collections of items that have a set order; these are by default rendered by the browser as a numbered list.

Definition lists. Collection of name and definition pairs. These tend to be used infrequently. Perhaps the most common example would be a FAQ list.

Lists Example

```
<html>
    <body>
        <p>Now we shall create some unordered lists:
        <ul>
            <li> List item 1</li>
            <li> List item 2</li>
            <li> List item 3</li>
            <li> List item 4</li>
        </ul>
        And then some ordered lists:
        <ol>
            <li> List item 1</li>
            <li> List item 2</li>
            <li> List item 3</li>
            <li> List item 4</li>
        </ol>
    </body>
</html>
```

Italics & Bold OR Strong & Emphasize

The `<i>` element was for italic text, now it is also for alternate text, such as foreign words, technical terms, or inline stage directions

The `<b>` element was for bold text, now it is also used as a stylistic offset such as keywords in a document of product names

The `<em>` was for emphasis, now it is used for words or sentences you would pronounce differently

Lastly, the `<strong>` element is for something with strong emphasis, it represents importance

Blockquote

The usage of the <`blockquote`> element for one thing – marking up a section of your webpage that is quoted from another source.

```
<html>
<body>
    <p>This is a paragraph with some text.
    <blockquote>Give a man a fish, you feed him once.
    Teach him how to fish and you feed him for life</blockquote>
    Again we have some random text just to illustrate the difference.
</body>
</html>
```

`blockquote.html`

Character Entities

These are special characters for symbols for which there is either no way easy way to type in via a keyboard (such as the copyright symbol or accented characters) or which have a reserved meaning in HTML (for instance the “<” or “>” symbols).

They can be used in an HTML document by using the entity name or the entity number.

e.g.,   and ©

Character Entities Example

```
<html>
<body>
    <p>This is how you write tags: &langle;   &rangle;  <br />
    <p>This text is copyrighted &copy;Vivek
</body>
</html>
```

entities.html

HTML SEMANTIC ELEMENTS

HTML5 Semantic Elements

Why are they needed?

One substantial problem with modern, pre-HTML5 semantic markup:

most complex web sites are absolutely packed solid with elements.

Unfortunately, all these elements can make the resulting markup confusing and hard to modify.

Developers typically try to bring some sense and order to the chaos by using id or class names that provide some clue as to their meaning.

Header and Footer

<header> <footer>

Most web site pages have a recognizable header and footer section.

Typically the **header** contains

- the site logo
- title (and perhaps additional subtitles or taglines)
- horizontal navigation links, and
- perhaps one or two horizontal banners.

Header and Footer

<header> <footer>

The typical footer contains less important material, such as

- smaller text versions of the navigation,
- copyright notices,
- information about the site's privacy policy, and
- perhaps twitter feeds or links to other social sites.

Header and Footer

Both the HTML5 <header> and <footer> element can be used not only for *page* headers and footers, they can also be used for header and footer elements within other HTML5 containers, such as <article> or <section>.

```
<header>
   <img src="logo.gif" alt="logo" />
   <h1>Fundamentals of Web Development</h1>
   ...
</header>
<article>
   <header>
       <h2>HTML5 Semantic Structure Elements </h2>
        <p>By <em>Randy Connolly</em></p>
        <p><time>September 30, 2012</time></p>
   </header>
   ...
</article>
```

Navigation

`<nav>`

The **<nav>** element represents a section of a page that contains links to other pages or to other parts within the same page.

Like the other new HTML5 semantic elements, the browser does not apply any special presentation to the <nav> element.

The <nav> element was intended to be used for major navigation blocks, presumably the global and secondary navigation systems.

Navigation

```
<header>
     <img src="logo.gif" alt="logo" />
     <h1>Fundamentals of Web Development</h1>
     <nav role="navigation">
          <ul>
               <li><a href="index.html">Home</a></li>
               <li><a href="about.html">About Us</a></li>
               <li><a href="browse.html">Browse</a></li>
          </ul>
     </nav>
</header>
```

Articles and Sections

<article> <section>

The **<article>** element represents a section of content that forms an independent part of a document or site; for example, a magazine or newspaper article, or a blog entry.

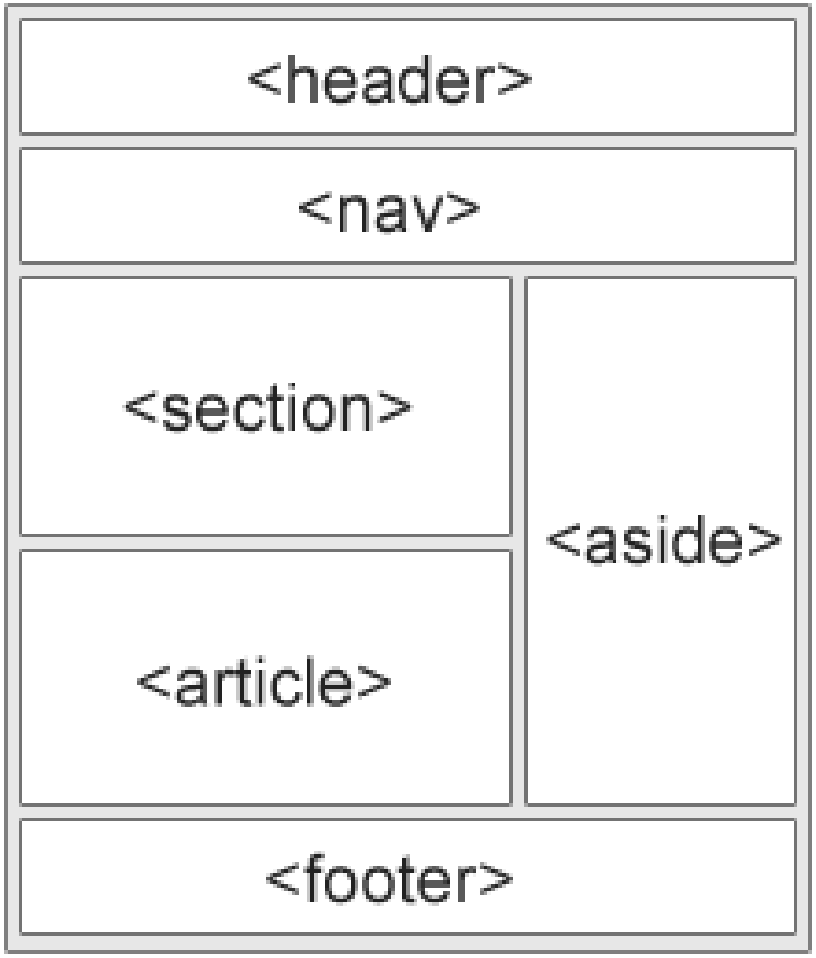
The **<section>** element represents a section of a document, typically with a title or heading. **<section>** is a much broader element,

Aside

<aside>

The **<aside>** element could thus be used for sidebars, pull quotes, groups of advertising images, or any other grouping of non-essential elements.

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time


<header>
<nav>
<section>
<aside>
<article>
<footer>

That’s all, folks!

Questions?