Brief Intro to HTML
Origins and Evolutions of HTML

- HTML was defined with SGML
- Original intent of HTML: General layout of documents that could be displayed by a wide variety of computers

Philosophies behind HTML5

- Specifying current browser behaviour to ensure interoperability
  - Clear specifications on error handling
- Not breaking the web (backward compatibility)
- Programmatic rather than theoretic (HTML5 is not XML)
- User > Web Designer > Browser Implementer > Standard Theorists
- Aiming at easier authoring of Web Applications
HTML Basic Syntax

- Elements are defined by *tags* (markers)
  - Tag format:
    - Opening tag: `<name>`
    - Closing tag: `</name>`
  - The opening tag and its closing tag together specify a container for the *content* they enclose
  - Not all tags have content
  - If a tag has no content, its form is `<name />`
  - The container and its content together are called an *element*
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>My First Heading</h1>
  My first paragraph.
</body>
</html>
• If a tag has *attributes*, they appear between its name and the right bracket of the opening tag
  – eg. `<table border="1" cellspacing="1">...
</table>`

• Comments:
  `<!-- ignore stuff here... -->`

• Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs

• Tags are suggestions to the browser, can be ignored (even if they are recognized by the browser)
Logical Markup versus Display Markup

- One of the ideas behind the web and (other places such as great typesetting languages like LaTeX) is the separation between what sort of information it is, and how it should be displayed
  - allows the same source information to be easily communicated in multiple environments
  - eg “this is a heading” or “this should be emphasised” should not determine how it should be displayed
    - computer browser
    - mobile phone
    - interface for partially sighted (eg. read out)
HTML Document Structure

• Every HTML5 document should begin with a DOCTYPE declaration:
  ```html
  <!DOCTYPE html>
  ```

• `<html>`, `<head>`, `<title>`, and `<body>` are required in every document

• The whole document must have `<html>` as its root

• A document consists of a head and a body

• The `<title>` tag is used to give the document a title, which is normally displayed in the browser’s window title bar (at the top of the display)

• Document is a `tree` of elements

• Visible elements are on `<body>` branch
Basic Text Markup

- **Headings**
  - Six sizes, 1 - 6, specified with `<h1>` to `<h6>`
  - 1, 2, and 3 use font sizes that are larger than the default font size
  - 4 uses the default size
  - 5 and 6 use smaller font sizes

```html
<html>
  <head>
    <title> Headings </title>
  </head>
  <body>
    <h1> Aidan's Airplanes (h1) </h1>
    <h2> The best in used airplanes (h2) </h2>
    <h3> "We've got them by the hangarful" (h3) </h3>
    <h4> We're the guys to see for a good used airplane (h4) </h4>
    <h5> We offer great prices on great planes (h5) </h5>
    <h6> No returns, no guarantees, no refunds, all sales are final (h6) </h6>
  </body>
</html>
```
Highlighting and Special Characters

• Font Styles and Sizes (can be nested)
  – Boldface - <b>
  – Italics - <i>
  – Larger - <big>
  – Smaller - <small>
  – Monospace - <tt>

• The <big> sleet <big> in <big> <i> Crete </i><br /> lies </big> completely </big> in </big> the street

<table>
<thead>
<tr>
<th>Character</th>
<th>Entity</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
<td>Ampersand</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
<td>Double quote</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
<td>Single quote (apostrophe)</td>
</tr>
<tr>
<td>$\frac{1}{4}$</td>
<td>¼</td>
<td>One quarter</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>½</td>
<td>One half</td>
</tr>
<tr>
<td>$\frac{3}{4}$</td>
<td>¾</td>
<td>Three quarters</td>
</tr>
<tr>
<td>$\deg$</td>
<td>°</td>
<td>Degree</td>
</tr>
<tr>
<td>(space)</td>
<td> </td>
<td>Nonbreaking space</td>
</tr>
</tbody>
</table>
The HTML5 Way

- Document is a *tree* of elements
- No need to have `<html>`, `<head>`, and even `<body>`
  - No need to close `<p>` elements
  - This is valid HTML5 code according to
    ```html
    <!DOCTYPE html>
    <html lang="en">
      <head>
        <meta charset="utf-8"/>
        <title> Our first document </title>
      </head>
      <body>
        <p>
          Greetings from your Webmaster!
        </p>
      </body>
    </html>
    ```
So what are these two terms all about? **Graceful Degradation** is a widely used term which ideology is basically using the latest technologies first, and then fix anything that needs fixing for older browsers. We do this on a daily basis: most of us code for Firefox first, then fix Internet Explorer. That is Graceful Degradation in the practice.

**Progressive Enhancement** refers to the habit of building first for the less capable, outdated browser and then enhance for the latest technologies. We, too, use this on a daily basis. For example, most of the times we code a website we start with the markup and then apply an external CSS file where we add all the styling. That is Progressive Enhancement in the practice.

Both technologies usually go hand in hand and have been part of the ways we do things for years. It’s just the terms that are not that well-known. And now, both of these practices need to evolve due to the new languages that are approaching.

Images

- **GIF** (Graphic Interchange Format)
  - 8-bit color (256 different colors)
- **JPEG** (Joint Photographic Experts Group)
  - 24-bit color (16 million different colors)
- Both use compression, but JPEG compression is better
- **Portable Network Graphics (PNG)**
  - Relatively new
  - Should eventually replace both gif and jpeg
Images

• Images are inserted into a document with the `<img />` tag with the `src` attribute

```html
<!DOCTYPE html>
<html>
<body>

<h2>Spectacular Mountain</h2>
<img src="pic_mountain.jpg" alt="Mountain View" style="width:304px;height:228px;"/>

</body>
</html>
```
Hypertext Links

- Hypertext is the essence of the Web!
- A link is specified with the `href` (**hypertext reference**) attribute of `<a>` (the anchor tag)
  - The content of `<a>` is the visual link in the document
  - Note: Relative addressing of targets is often easier to maintain and more portable than absolute addressing
  - You can link to elements in the same document, use an id attribute:
    - `<H2 id="Link">Link to me!</H2>`
    - `<a href="#Link">linking...</a>`
  - The `href` can be any file, not just html.
<html>
<head>  
<title> Links </title> 
</head> 
<body> 
<h1> Aidan's Airplanes </h1> 
<h2> The best in used airplanes </h2> 
<h3> "We've got them by the hangarful" </h3> 
<h2> Special of the month </h2> 
<p>  
1960 Cessna 210 <br />  
<a href = "C210data.html"> Information on the Cessna 210 </a> 
</p> 
</body> 
</html>
Lists

- **Unordered lists**
  - The list is the content of the `<ul>` tag
  - List elements are the content of the `<li>` tag
  - Use `<ol>` for *ordered* lists
  - Type attribute can change from numbers to letters etc.

```html
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
  <li> Cessna Skyhawk </li>
  <li> Beechcraft Bonanza </li>
  <li> Piper Cherokee </li>
</ul>
```

Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee
Lists (continued)

- **Definition lists (for glossaries, etc.)**
  - List is the content of the `<dl>` tag
  - Terms being defined are the content of the `<dt>` tag
  - The definitions themselves are the content of the `<dd>` tag

```html
<h3> Single-Engine Cessna Airplanes </h3>
<dl>
  <dt> 152 </dt>  <dd> Two-place trainer </dd>
  <dt> 172 </dt>  <dd> Smaller four-place airplane </dd>
  <dt> 182 </dt>  <dd> Larger four-place airplane </dd>
  <dt> 210 </dt>  <dd> Six-place airplane - high performance </dd>
</dl>
```
Tables

- A table is a matrix of cells, each possibly having content
  - The cells can include almost any element
  - Some cells have row or column labels and some have data
  - A table is specified as the content of a `<table>` tag
  - A `border` attribute in the `<table>` tag specifies a border between the cells
  - The `border` attribute can be set to a number, which will be the border width
  - Without the `border` attribute, the table will have no lines
  - Tables are given titles with the `<caption>` tag, which can immediately follow `<table>`
  - Each row of a table is specified as the content of a `<tr>` tag
  - The row headings are specified as the content of a `<th>` tag
  - The contents of a data cell is specified as the content of a `<td>` tag
Tables (continued)

• The `cellspacing` attribute of `<table>` is used to specify the distance between cells in a table.

  The `cellpadding` attribute of `<table>` is used to specify the spacing between the content of a cell and the inner walls of the cell.

<table>
<thead>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
<tr>
<td>John</td>
<td>Doe</td>
<td>80</td>
</tr>
</tbody>
</table>
Tables for layout?

Tables have frequently been used to layout a webpage.

Why is this not a great idea?

<table cellspacing = "50">
  <tr>
    <td> Colorado is a state of ... </td>
    <td> South Dakota is somewhat... </td>
  </tr>
</table>
Typical Layout in HTML or XHTML

- Layout in XHTML created named div(isions) and applied styles to them
- HTML5 has named structural elements.

Layout in HTML5

```html
1. <!doctype html>
2. <html>
3. <head>
4.   <title>Page title</title>
5. </head>
6. <body>
7.   <header>
8.     <h1>Page title</h1>
9.   </header>
10. <nav>
11.   <!-- Navigation -->
12. </nav>
13. <section id="intro">
14.     <!-- Introduction -->
15. </section>
16. <section>
17.     <!-- Main content area -->
18. </section>
19. <aside>
20.     <!-- Sidebar -->
21. </aside>
22. <footer>
23.     <!-- Footer -->
24. </footer>
25. </body>
26. </html>
```
The new structural elements

- `<header>`
The header element contains introductory information to a section or page.

- `<nav>`
The nav element is reserved for a section of a document that contains links to other pages or links to sections of the same page.

- `<section>`
The section element represents a **generic document or application section**. It acts much the same way a `<div>` does by separating off a portion of the document.

- `<article>`
The article element represents a portion of a page which can stand alone such as: a blog post or a forum entry.

- `<aside>`
Aside, represents content related to the main area of the document. Usually expressed in sidebars that contain elements like related posts, tag clouds.

- `<footer>`
The footer element is for marking up the footer of, not only the current page, but each section contained in the page.
Document outlining algorithm

It produces outline summaries of Web pages based on how they are marked up.

1. Horses for sale
   1. Mares
      1. Pink Diva
      2. Ring a Rosies
      3. Chelsea’s Fancy
   2. Stallions
      1. Korah’s Fury
      2. Sea Pioneer
      3. Brown Biscuit

Online Outliner Testing Tool: http://gsnedders.html5.org/outliner/
What elements affects outline?

- Two things in the mark-up affect the outline of a Web page:
  - Implicit: heading content (h1 to h6 and hgroup)
    - The heading elements have a rank
      - <h1> has the highest rank, and
      - <h6> has the lowest rank.
  - Explicit: sectioning content (section, article, aside and nav).
    - Explicit outline also needs headings to be useful.
    - The basic rule is: the first HTML heading element (one of <h1>, <h2>, <h3>, <h4>, <h5>, <h6>) defines the heading of the current section.
  - Relative ranking matters only within a section; the structure of the sections determines the outline, not the heading rank of the sections.

Validation

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets or to find broken links, there are other validators and tools available.
HTML 5 Validation

http://html5.validator.nu/

Validator.nu (X)HTML5 Validator (Highly Experimental)

Validator Input

- Address

- Show Image Report
- Show Source

- Validate

About this Service • More options
We will give just a brief overview of common aspects of HTML5. Many references can be found in texts and on-line the official standards see also:

- W3C Editor’s Draft

- Books
  - HTML5 for Web Designers, by Jeremy Keith, Published by A Book Apart.
  - [http://html5demos.com/](http://html5demos.com/)
  - Introducing HTML5 (Second Edition), By Bruce Lawson, Remy Sharp
    Published Oct 18, 2011 by New Riders.

- HTML5 Tutorials
  - 15 Useful HTML5 Tutorials
  - HTML and the Future of the Web
More References

- *Note these may include non-XHTML features!*
- Tutorials (check for XHTML compatibility)
  - *Getting Started with HTML*, by Dave Ragget
    [http://www.w3.org/MarkUp/Guide/](http://www.w3.org/MarkUp/Guide/)
  - *More Advanced Features*, by Dave Ragget
    [http://www.w3.org/MarkUp/Guide/Advanced.html](http://www.w3.org/MarkUp/Guide/Advanced.html)
- Reference
  - HTML & XHTML: The Complete Reference
- Reading
  - *Adding a Touch of Style*, by Dave Ragget
    [http://www.w3.org/MarkUp/Guide/Style.html](http://www.w3.org/MarkUp/Guide/Style.html)