HTML/XML

XHTML Authoring
Creating Tables

Table: An arrangement of horizontal rows and vertical columns. The intersection of a row and a column is called a cell.

Used for:
- displaying tabular data in rows and columns
- controlling the layout and appearance of a Web page
Creating Tables

- Creating tables requires organisation.

- All tables begin with the `<table>` tag and end with a `</table>` tag.

- In between those tags are three other tags to be aware of:
  - `<tr>` defines a horizontal row
  - `<td>` defines a data cell within that row
  - `<th>` specifies a data cell as a table heading (generally, a table heading cell is formatted as centered and bold).
Creating Tables

```html
<html>
<head>
<title>Table</title>
</head>

<body>
<table>
<tr><th>Column 1</th> <th>Column 2</th></tr>
<tr><td>Row 2</td> <td>Row 2</td></tr>
<tr><td>Row 3</td> <td>Row 3</td></tr>
</table>
</body>
</html>
```
Creating Tables

- Formatting Tables - Attributes

  - **width** (default: *to fit*)
    Width of table or cells (in pixels or as a % of the page)

  - **border** (default: 0)
    Width of the border (in pixels - `<table>` tag)
Creating Tables

- Formatting Tables - Attributes

  - `cellspacing` (default: depends on browser)
    Space between cells (in pixels - `<table>` tag).

  - `cellpadding` (default: depends on browser)
    Space between border and content (in pixels - `<table>` tag)
Creating Tables

- Formatting Tables - Attributes

  - **valign (default: center)**
    Vertical alignment of cell contents (**top**, **bottom**, or **baseline**)

  - **align (default: <td> left, <tr> center)**
    Horizontal alignment of cell contents (**left**, **right**, **center**, or **justify**)
```
<html>
<head>
<title>My first formatted table</title>
</head>

<body>

<table border="1">
<tr><th> Column 1 </th> <th> Column 2 </th></tr>
<tr><td align="center"> Row 2 </td><td align="center"> Row 2 </td></tr>
<tr><td align="center"> Row 3 </td><td align="center"> Row 3 </td></tr>
</table>

</body>
</html>
```
Advanced Tables

- The **colspan** and **rowspan** attributes are used to create complex tables:

```html
<html>
<head>
<title>My first formatted table</title>
</head>
<body>
<table border="1">
<tr><th colspan="2">Column 1 and Column 2 are combined</th></tr>
<tr><td>Row 2</td><td rowspan="2">Row 2 and Row 3 are combined</td></tr>
<tr><td>Row 3</td></tr>
</table>
</body>
</html>
```
Every strictly conforming XHTML document must begin with an XML declaration:

```xml
<?xml version="1.0" encoding="utf-8"?>
```
XML Declaration

- The version attribute gives the version of XML that the document is based on – the current supported value is “1.0”

- The encoding attribute defines the encoding in which the text is stored on a computer.
To conform to the XHTML 1.0 standard, the root element of the document must be `<html>`.

The root element of the document must contain an `xmlns` declaration for the XHTML namespace.

The namespace for XHTML is defined to be `http://www.w3.org/1999/xhtml`. An example root element might look like:

```html
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```
The **xmlns** attribute indicates the URL identifying the *namespace* location – a resource containing a general definition of the language.

The value for XHTML is fixed.
Three "flavours" of XHTML 1.0

- XHTML 1.0 is specified in three "flavours".

- You specify which of these variants you are using by inserting a line at the beginning of the document.

- Each variant has its own DTD - which sets out the rules and regulations for using HTML in a succinct and definitive manner.
XHTML 1.0 Strict

- Use this when you want really clean structural mark-up, free of any markup associated with layout.

- Use this together with CSS to get the font, color, and layout effects you want.
XHTML 1.0 Transitional

- Many people writing Web pages for the general public to access used this flavor of XHTML 1.0.

- The idea was to take advantage of XHTML features including style sheets but nonetheless to allow small adjustments to your markup for the benefit of those viewing your pages with older browsers which couldn’t understand style sheets.
XHTML 1.0 Frameset

- Use this when you want to use Frames to partition the browser window into two or more frames.
DOCTYPE Declarations

- There must be a DOCTYPE declaration in the document prior to the root element.

- The public identifier included in the DOCTYPE declaration must reference one of the three DTDs using the respective Formal Public Identifier.
DOCTYPE Declarations

XHTML 1.0 Strict:

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```
DOCTYPE Declarations

XHTML 1.0 Transitional:

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```
DOCTYPE Declarations

XHTML 1.0 Frameset:

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```
Character Encoding

- We also need to specify the character encoding for the HTML document.

- Common values:
  - UTF-8 - Character encoding for Unicode
  - ISO-8859-1 - Character encoding for the Latin alphabet

- In theory, any character encoding can be used, but no browser understands all of them. The more widely a character encoding is used, the better the chance that a browser will understand it.

- To view all available character encodings, look at IANA character sets.
Charset

- Define the your character set by using the `charset` attribute.

- Do this using a `<meta>` tag nested within the `<head>` tag:

  ```html
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  ```
Put it all together...

```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <title>XHTML</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body> </body>
</html>
```

© Nina Bresnihan, School of Computer Science & Statistics, TCD, 2016
Entities

Special Characters

- Symbols such as +, -, %, and & are used frequently.
- Not all Web browsers display these symbols correctly.
- HTML uses a little computer shorthand to tell the browser how to interpret these symbols.
- [www.w3.org/TR/REC-html40/sgml/entities.html#h-24.2.1](http://www.w3.org/TR/REC-html40/sgml/entities.html#h-24.2.1) - Contains a complete list of the characters supported by HTML.

© Nina Bresnihan, School of Computer Science & Statistics, TCD, 2016
## Entities

<table>
<thead>
<tr>
<th>Char</th>
<th>Code</th>
<th>Description</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>©</td>
<td>©</td>
<td>copyright</td>
</tr>
<tr>
<td>®</td>
<td>®</td>
<td>registered trademark</td>
</tr>
<tr>
<td>\²</td>
<td>²</td>
<td>superscript2</td>
</tr>
<tr>
<td>\³</td>
<td>³</td>
<td>superscript3</td>
</tr>
<tr>
<td>/</td>
<td>´</td>
<td>acute accent/fada</td>
</tr>
<tr>
<td>\</td>
<td>`</td>
<td>grave accent</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>hash sign</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>percent sign</td>
</tr>
</tbody>
</table>
Entities

Special Characters

- Probably the most important of these special characters is the non-breaking space:

    &nbsp;

- Used to insert a space inside a HTML document.

- Also used to create an empty cell in a table.
Even More Tags

**Meta Tags**
- Contained within the `<head>` tag.
- Can use as many `<meta>` tags as you want in your page.
- Doesn’t appear in the document.
- Used to:
  - identify the page’s author,
  - identify keywords used for searching,
  - a brief description to appear in search results
  - give commands to the browser
Even More Tags

- **Meta Tags**

- **Improve Searching**
  - Search Engines add the content of your Web pages to their indexes
  - When a potential visitor enters a search phrase, the search engine checks its index to find that word and returns any pages that include that word.
Even More Tags

- Meta Tags

- Improve Searching
  - You can use the `<meta>` tag to include product names, geographic locations, industry terms, and synonyms.

- There are three `<meta>` tags that work to help improve your chances of being found by a search engine
  - Keywords
  - Description
  - Author
Even More Tags

- **Meta Tags**
  
- **Improve Searching**
  - *Keywords* – words that you feel people might use to search for your web page, or synonyms for words in your document.
  
  *Description* – usually a paragraph of information about your page. Some search engines use this description to describe your page; other search engines use the first few lines of text in your document.
  
  *Author* – This is your opportunity to shine. Just in case someone is searching for your name, they will find it if you enter that information into the `<meta>` tag.
Even More Tags

- **Meta Tags**

- Improve Searching
  - Meta information for search engines comes in pairs: name and contents.

```html
<html>
<head><title>Your HTML Page</title>
<meta name="keywords" content="keywords, that people, might, use, to, search, for, your, page" />
<meta name="description" content="a brief paragraph describing your document" />
<meta name="author" content="your name" />
</head>
<body>Insert your document here</body>
</html>
```
**Even More Tags**

- **Meta Tags**
  - Refresh and Redirect (not all browsers)
    - Replace one page with another
    - Redirect a link
    - Include a splash page (please don’t)
    - You can use meta information to force the page to change within a given time span.
    - Can use the refresh tag to refresh the same page – use if there are frequent updates

```
<meta http-equiv="refresh" content="time in seconds; URL=URL for new page" />
```
Lab Assignment

- Create the following html table:

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00-11:00</td>
<td>Contextual Media (Games Development)</td>
<td>Programming for Digital Media (Programming Lab)</td>
<td>Audio, Video &amp; Sensor Technologies (Moving Image for Digital Media Applications)</td>
<td>Authoring for Digital Media (HTML5/CSS/jQuery)</td>
<td>Audio, Video &amp; Sensor Technologies (Audio Technologies)</td>
</tr>
<tr>
<td>11:00-12:00</td>
<td>Programming for Digital Media (Client Side Programming)</td>
<td>Image Processing and 3D Modelling (Visual Computing)</td>
<td>Contextual Media (Cultural and Critical Studies)</td>
<td>Image Processing and 3D Modelling (Graphic Design)</td>
<td>Contextual Media (Interactive Narratives)</td>
</tr>
<tr>
<td>12:00-13:00</td>
<td></td>
<td></td>
<td>Authoring for Digital Media (HTML5/CSS/jQuery)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>