SENG1050

# Internet Communications

## Week 05 HTML5

## Lecture Plan



### Weekly program (lectures)

- ✓ Week 1 – The Internet, Protocols, TCP/IP, Email, HTTP
- ✓ Week 2 – HTML basics
- ✓ Week 3 – XML and DTD
- ✓ Week 4 – CSS
- ❑ **Week 5 – More HTML with CSS**
- ❑ Week 6 – Revision and Midterm
- ❑ Week 7 – XSLT
- ❑ Week 8 – JavaScript
- ❑ Week 9 – More JavaScript and User Interface
- ❑ Week 10 – Encoding, Compression and Information Retrieval
- ❑ Week 11 – Security and Encryption
- ❑ Week 12 – Ethics and Course review

## Key concepts from last lecture



- How CSS can separate semantics and visual formatting
- CSS facilitate smart, effective and efficient management, of sites
- Syntax of CSS
- CSS Selectors:

  *selector* { *property*$_1$: *value*$_1$; *property*$_2$: *value*$_2$; … }
- Different types of selectors:
  - Element selector, ID selector, Class Selector
- Common CSS properties and their values
  - ***font-style, font-weight, font-family, font-size***
  - ***text-decoration, text-align, text-indent***
  - ***list-style-type, list-style-image***

## Key concepts from last lectures



- RGB Color Scheme
- Background management with CSS properties
  - ***background-image, background-repeat, background-position***
- Margin, borders and paddings

- CSS Images: floating
- CSS Validation

## Week 05 Outline



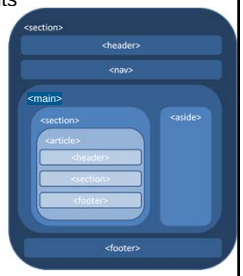
**More HTML with CSS**

- ❑ HTML5 Structural Elements
- ❑ Image Maps
- ❑ Tables
- ❑ Tables with CSS
- ❑ CSS positioning
- ❑ Page Layout with CSS

## HTML5 Structural Elements



- AKA: Semantic structural elements
- Not intended to replace ***div*** completely rather to use along to structure the web page in a more meaningful manner



```
<body>
  <section>
    <header>
      <h1>Big Header: HTML5 Semantic Structure</h1>
    </header>
    <nav>
      <a href="#home"> home </a>
      <a href="#about"> about </a>
    </nav>
    <main>
      <aside style="float:right; width:20%">
        <h2> aside </h2>
        <p> The aside element is used to represent content that is tangibly related to the content
surrounding it, but could be considered separate. </p>
      </aside>
      <section>
        <article>
          <header>
            <h2> article 1 header SENG1050 </h2>
          </header>
          <section>
            <h3> section 1 </h3>
            <p> This is the first section.</p>
          </section>
          <footer>
            article 1 footer SENG1050
          </footer>
        </article>
      </section>
    </main>
    <footer>
      &copy; Copyright Newcastle University 2017
    </footer>
  </section>
</body>
```

7


```
<body>
  <section>
    <header>
      <h1>Big Header: HTML5 Semantic Structure</h1>
    </header>
    <nav>
      <a href="#home"> home </a>
      <a href="#about"> about </a>
    </nav>
    <main>
      <aside style="float:right; width:20%">
        <h2> aside </h2>
        <p> The aside element is used to represent content that is tangibly related to the content
surrounding it, but could be considered separate. </p>
      </aside>
      <section>
        <article>
          <header>
            <h2> article 1 header SENG1050 </h2>
          </header>
          <section>
            <h3> section 1 </h3>
            <p> This is the first section.</p>
          </section>
          <footer>
            article 1 footer SENG1050
          </footer>
        </article>
      </section>
    </main>
    <footer>
      &copy; Copyright Newcastle University 2017
    </footer>
  </section>
</body>
```


8

## HTML5 Structural Elements

9

- **Header Element:** ***<header>***
  - block display; contains the headings of either a web page document or an area in the document such as a section or article
  - should be used as a container for introductory content or set of navigational links.
  - You can have several ***<header>*** elements in one document.
- **Nav Element:** ***<nav>***
  - block display; contains a section of navigation hyperlinks
  - Notice that NOT all links of a document should be inside a ***<nav>*** element. The ***<nav>*** element is intended only for major block of **navigation links**.

## HTML5 Structural Elements

10

- **Main Element:** ***<main>***
  - block display; contains main page content
  - The content inside the ***<main>*** element should be unique to the document.
  - There must not be more than one ***<main>*** element in a document.
  - The ***<main>*** element must NOT be a descendent of an ***<article>, <aside>, <footer>, <header>***, or ***<nav>*** element.
- **Footer Element**
  - block display; contains the footer content of a web page or specific area (such as a section or article) on a web page
  - A ***<footer>*** element will typically contain information about its containing element.
    - e.g. the author of the document, copyright information, links to terms of use, contact information, etc.
  - You can have several ***<footer>*** elements in one document.

## HTML5 Structural Elements

11

- **Aside Element:** ***<aside>***
  - block display; contains a sidebar, a note, or other tangential content
  - defines some content aside from the content it is placed in.
  - The aside content should be related to the surrounding content.
- **Section Element:** ***<section>***
  - contains a "section" of a document, such as a chapter or topic or any other sections of the document
  - block display
- **Article Element:** ***<article>***
  - contains an independent entry, such as a blog posting, comment, or e-zine article that could stand on its own (self-contained)
  - block display
  - should be possible to distribute it independently from the rest of the site.

## HTML5 Structural Elements

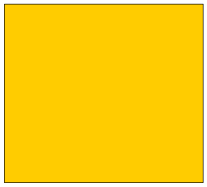
12

| | |
|---|---|
| ***<details>*** | Defines additional details that the user can view or hide |
| ***<figcaption>*** | Defines a caption for a ***<figure>*** element |
| ***<figure>*** | Defines self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| ***<mark>*** | Defines marked or highlighted text |
| ***<progress>*** | Defines the progress of a task |
| ***<summary>*** | Defines a visible heading for a ***<details>*** element |
| ***<time>*** | Defines a date/time |

## HTML5 Structural Elements

13

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">
  <figcaption>Fig.1 - A view of the <mark>pulpit rock</mark> in Norway.</figcaption>
</figure>
```

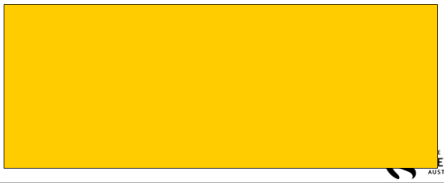


## HTML5 Structural Elements

14

```
<details>
<summary>Copyright 1999-2014.</summary>
  <p> - by Refsnes Data. All Rights Reserved.</p>
  <p>All content and graphics on this web site are the property of the company Refsnes Data.</p>
</details>
<p><b>Note:</b> The details tag is currently only supported in Opera, Chrome, and in Safari 6.</p>
```

Hide/show



## HTML5 Structural Elements

15

Downloading progress:

```
<progress value="22" max="100">
</progress>
<p>Finishing at <time>10:00</time>.</p>
<p><strong>Note:</strong> The progress tag is not supported in Internet Explorer 9 and earlier versions.</p>
```

With javascript


## HTML5 Audio Element

16

- The *<audio>* tag defines sound, such as music or other audio streams.
- Currently, there are 3 supported file formats for the *<audio>* element:
  - MP3, Wav, Ogg
  - Not all formats are supported by all browsers

```
<audio controls>
        <source src="horse.ogg" type="audio/ogg">
        <source src="horse.mp3" type="audio/mpeg">
        Your browser does not support the audio element.
</audio>
```


## HTML5 Video Element

17

- The *<video>* tag specifies video, such as a movie clip or other video streams.
- Currently, there are 3 supported video formats for the *<video>* element:
  - MP4, WebM, and Ogg:
  - Not all formats are supported by all browsers

```
<video width="320" height="240" controls>
        <source src="movie.mp4" type="video/mp4">
        <source src="movie.ogg" type="video/ogg">
        Your browser does not support the video tag.
</video>
```

## HTML5 Source Element

18

- The *<source>* tag is used to specify multiple media resources for media elements, such as *<video>* and *<audio>*.
- The *<source>* tag allows you to specify alternative video/audio files which the browser may choose from, based on its media type or codec support.
- Attributes:
  - *src*= Specifies the URL of the media file
  - *type*= Specifies the MIME type of the media resource

```
<video width="320" height="240" controls>
        <source src="movie.mp4" type="video/mp4">
        <source src="movie.ogg" type="video/ogg">
        Your browser does not support the video tag.
</video>
```

## HTML5 Embed Element

19

- The *<embed>* tag defines a container for an external application or interactive content (a plug-in).
- Many web browsers have supported the *<embed>* tag for a long time. However, the *<embed>* tag has not been a part of the HTML 4 specification.

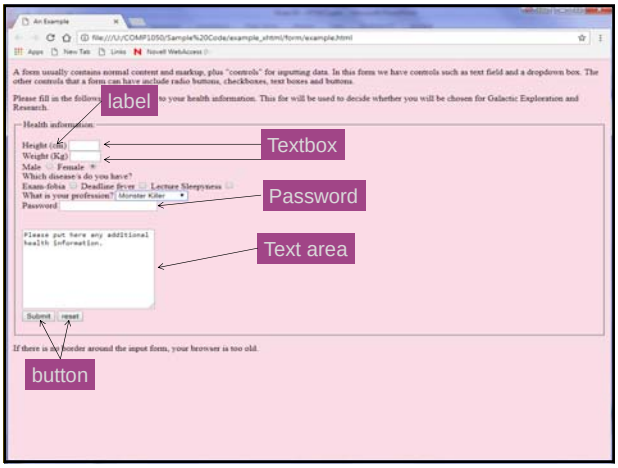
```
<embed src="helloworld.swf">
```

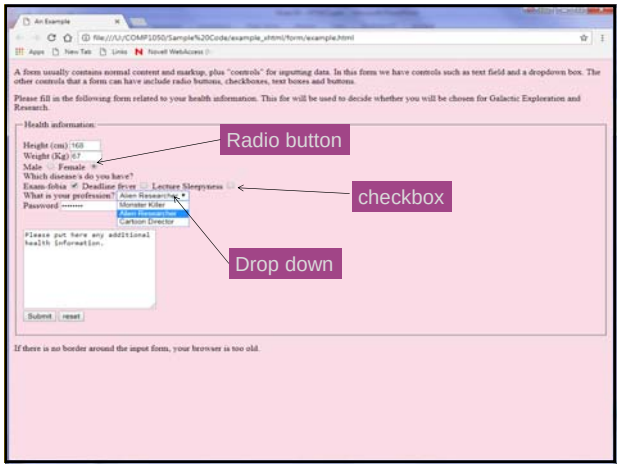
https://www.w3schools.com/tags/tag_embed.asp

## HTML Form Element

20

- A form usually contains normal content and markup, plus “controls” for inputting data
  - Controls are buttons, text boxes, checkboxes, radio buttons, menus, …
  - Can also place labels on those controls
  - A specific button on the page is clicked to submit the data



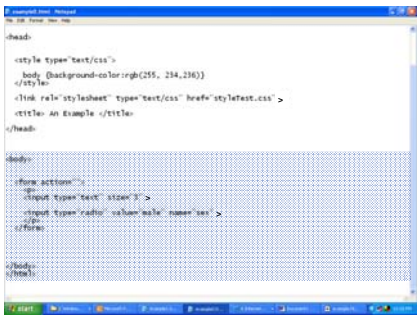

## HTML Form Element

23

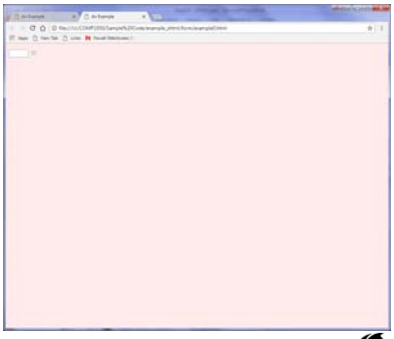
```
<form>
  <input >
  <input >
</form>

<form>
  <input type="text" size="3" name="height">
  <input type="radio" value="male" name="sex">
  <input type="radio" value="female" name="sex">
</form>
```

24



25



26

## <input> Tag

- Most controls are specified by **<input>**
- **<input type = " ">**
- **type = "text |password**
  - **| submit | reset | button**
  - **size = "*number*"**
    - Width of the control field
    - In pixels, except for **text** and **password** where it means "number of characters"
  - **maxlength = "*number*"**
    - The maximum number of characters allowed in **text** and **password**
    - Can be greater than **size**

27

## <input> Tag

- Most controls are specified by **<input>**
- **<input type = " ">**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button**

**name = "*name*"**
The name for the data of this input
**value = "*value*"**
The (initial) value (and label) for the input
Optional except for **type="radio"** and **type="checkbox"**
Cannot be used with **type="file"**

<input type="radio" name="gender" value="male">
<input type="radio" name="gender" value="female">

28

## <input> Tag

- Most controls are specified by **<input>**
- **<input type = " ">**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button**

**name = "*name*"**
The name for the data of this input
**value = "*value*"**
The (initial) value (and label) for the input
Optional except for **type="radio"** and **type="checkbox"**
Cannot be used with **type="file"**

<input type="checkbox" name="disease" value="Exam-fobia">
<input type="checkbox" name="disease" value="Deadline-fever">
<input type="checkbox" name="disease" value="Lecture nosia">

29

## <input> Tag

- Most controls are specified by **<input>**
- **<input type = " ">**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button**

**checked = "checked"**
For **radio** and **checkbox** only
Specifies that the button is initially on

<input type="radio" name="gender" value="male" **checked="checked"**>
<input type="radio" name="gender" value="female">
<input type="checkbox" name="disease" value="Exam-fobia" **checked="checked"**>
<input type="checkbox" name="disease" value="Deadline-fever">
<input type="checkbox" name="disease" value="Lecture nosia">

30

## <input> Tag

- Most controls are specified by **<input>**
- **<input type = " ">**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button | image"**

**submit or reset button**
**button:** mostly used with a JavaScript to activate a script
**image:** an image as a submit button

<input type="submit" value="Submit">
<input type="reset" value="reset">

<input type="button" value="Click me" onclick="msg()">

## <input> Tag

31

- Most controls are specified by **<input>**
- **<input type = “ ”>**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button | image”**

**submit or reset button**
**button:** mostly used with a JavaScript to activate a script
**image: an image as a submit button**

<input type="image" src="img_submit.gif" alt="submit" width="48" height="48">
**src = "*url*"**
For **image** only
Has an **alt** attribute – **USE IT!**



## <input> Tag

32

- Most controls are specified by **<input>**
- **<input type = “ ”>**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button | image”**
  - **| file**

Defines a file-select field and a "Browse..." button (for file uploads)

<input type=“file”>



## <input> Tag

33

- Most controls are specified by **<input>**
- **<input type = “ ”>**
- **type = "text |password**
  - **| checkbox | radio**
  - **| submit | reset | button | image”**
  - **| file**
  - **| hidden "**

Defines a hidden input field

<input type="hidden" name="country" value="Norway">

http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_input_type_hidden



## <button>

34

Use <input> to create buttons

- **<button>**
  - Defines a clickable button
  - You can put content (text/image) inside a button
  - Wraps around markup to make the entire area act as a button
- **name = "*name*"**
- **type = "submit | reset | button"**
  - Which **<input>** type the button acts like

<button type=“button” onclick=“alert(‘Hello world’)”> Click Me ! </button>



## textarea

35

- **<textarea>**
  - A multi-line text control
  - The initial value is the content
- **name = "*name*"**
- **cols = "*number*"**
  - Specifies the visible width (in characters)
- **rows = "*number*"**
  - Specifies the visible height (in characters)
- <textarea rows="10" cols="30">Please put here any additional health information.</textarea>

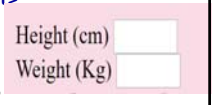



## Labels

36

- Can associate a label with any control

```
<label for="id">Label text</label>
…
<control id="id">
```



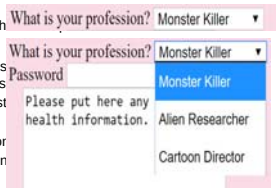

```
<label for=“height”>Height(cm)</label>
<input type=“text” size=“3” id=“height”>

<label for=“weight”>Weight(Kg)</label>
<input type=“text” size=“3” id=“weight”>
```

## Drop down

```
<label for="profession"> What is your profession?</label>
<select name="profession" id="profession">
    <option value="monster killer" selected="selected">Monster Killer</option>
    <option value="alien researcher">Alien Researcher</option>
    <option value="cartoon">Cartoon Director</option>
</select><br/>
```

- **<select>**
  - Container for a multiple-choice menu: can only contain **<option>** tags
  - **multiple = "multiple"**
    - Allows the user to select more th
  - **size = "*number*"**
    - How many different options are vis
    - **size="1"** acts as a drop down lis
    - Higher values act as a scrolling list
- **<option>**
    - One option within a **<select>** cor
    - The text between the begin and en
  - **selected = "selected"**
    - Specifies if this option is initially selected
  - **value = "*text*"**
    - The return value if this option is selected



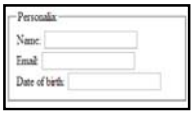



## Grouping with the fieldset tag



- <fieldset> : to group the different kinds of fields
  - <legend> : The name of the group

```
<form>
 <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text"><br>
    Email: <input type="text"><br>
    Date of birth: <input type="text">
 </fieldset>
</form>
```






## Forms and CGI



- CGI = Common Gateway Interface
- Defines how form data is passed to and from a server-side script for dynamic processing
- Other server-side scripting languages (JSP, ASP, …) use similar mechanisms



## Forms and CGI



- Allows Web pages with simple interaction
  1. User **enters** data into form
  2. Browser "**submits**" form to **server**
  3. **Server** processes data and responds
  - Response = update internal records, dynamically generate Web page, …
  - Processing on server performed by CGI scripts, or JSP/ASP pages, or …



## Forms and CGI



- **action = "*url*"**
  - The URL of the server page which will process the form
- **method = "get** | **post"**
  - The HTTP request used to submit the form to the server
- **get**
  - The form data is appended in the URL in name/value pairs
  - ***url?input1=value1&input2=value2&…***
  - Never use it to send sensitive data
  - Size limitation (3000 Characters)
  - Results can be bookmarked
- **post**
  - Form data is sent inside the body of HTTP request
  - No size limitation
  - Results can not be bookmarked



```
<form action="demo_form_method.asp" method="get">
  First name: <input type="text"
name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
</form>

<p>Click on the submit button, and the input will
be sent to a page on the server called
"demo_form_method.asp".</p>
```

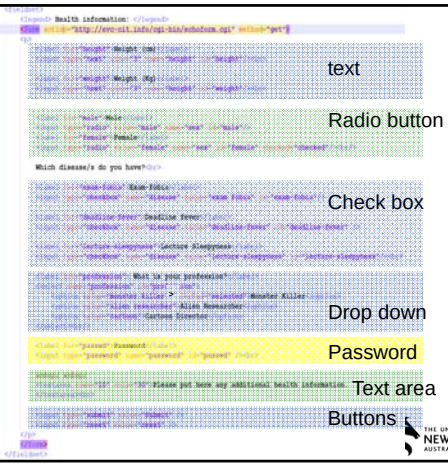
## HTML Form: sending the data

43

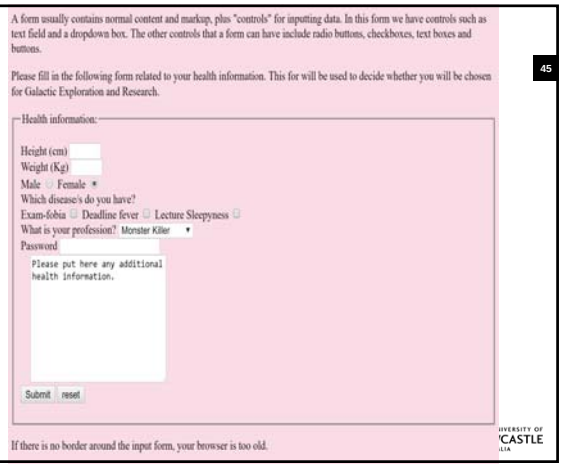
<form action="http://www.cs.tut.fi/cgi bin/run/~jkorpela/echo.cgi" method="get">

*Or email the form to your email*

<form action="mailto:your@email.com" method="post">




44




45

## <input> more in HTML5

- **color** Defines a colour picker
- **date** Defines a date control (dd/mm/yyyy)
- **datetime** | **datetime-local** Defines a date and control
- **month** Defines month and year control
- **week** Defines week and year control
- **time** Defines control for entering a time

46

## <input> more in HTML5

- **search** Defines a text field for entering a search string
- **tel** Defines a field for entering a telephone number
- **number** Defines a field for entering number
- **range** Defines a slider control for entering a number
- **email** Defines a field for an e-mail address
- **url** Defines text field for entering a URL

47

## <input> more in HTML5

48

```
<form action="demo_form.asp">
   Guest Name: <input type="text" name="guestname"><br>
   Date Of Birth: <input type="date" name="dob"><br>
   Date and Time Of Flight: <input type="datetime-local" name="dtarr"><br>
   Favorite Color: <input type="color" name="favcolor"><br>
   Month of Joining: <input type="month" name="depmonth"><br>
   Deposit Amount: <input type="number" name="nod"><br>
   Expertise level: <input type ="range" min ="1" max="10" step ="1" value ="5"/><br>
   Email: <input type="email" name="email"><br>
   <input type="submit" value="Submit">
</form>
```

## Image Maps

49

- An image with clickable areas.
- Can be effective for linking to related data from an HTML page
  - Example: Map of Australia could link to a page containing information on each State
- Simplest type is a Single Image Link – whole image is the anchor

```
<a href="world.html">
  <img src="world.jpg" alt="The World" >
</a>
```

## Image Maps

50

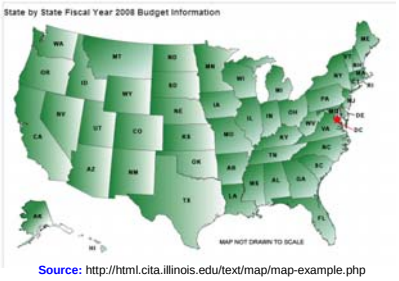
```
<p> <a href="http://www.nasa.gov">
  <img src="nasa.jpg" alt="NASA" width="42" height="42">
  </a>
   The National Aeronautics and Space Administration is
  the United States government agency that is
  responsible for the civilian space program as well as
  for aeronautics and aerospace research.
</p>
```

The National Aeronautics and Space Administration is the United States government agency that is responsible for the civilian space program as well as for aeronautics and aerospace research.

## Image Maps

51

- More interesting/useful if different parts of an image link to different information




Source: http://html.cita.illinois.edu/text/map/map-example.php

## Image Maps

52

- Main tags are:

```
<img  ..  >

<map …>
       <area … >
       <area … >
</map>
```

## (Client-side) Image-maps

53

- A way of adding hyperlinks from an area in an image
- `<img src="url" usemap="#label" >`
- `<map name="label">`
  - Contains a description of the areas of the image and their links
  - `name="label"` must match `usemap="#label"`

54

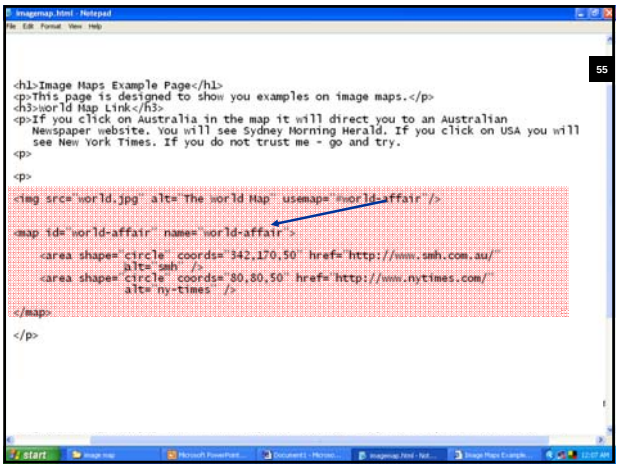
### Image Maps Example Page

This page is designed to show you examples on image maps.

**World Map Link**

If you click on Australia in the map it will direct you to an Australian Newspaper website. You will see Sydney Morning Herald. If you click on USA you will see New York Times. If you do not trust me - go and try.

This picture has been taken from the web.




## (Client-side) Image-maps

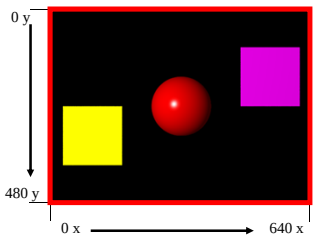
56

- **<area>** describes an area of the image-map
- **shape="rect | circle | poly | default"**
  - The shape of the area (**default** = entire image)
- **coords="*values*"**
  - A comma-separated list of pixel coordinates that define the area relative to the image size
- **href="*url*"**
  - The URL to link to for this area
- **download="URL"**
  - The target will be downloaded when user clicks

## (Client-side) Image-maps

57

```
<map name="objects">
     <area shape="rect"
        coords="29,240,175,384"
        href="yellowexp.html"/>
     <area shape="circle"
        coords="318,238,70"
        href="redexp.html"/>
     <area shape="rect"
        coords="466,94,611,240"
        href="purpleexp.html"/>
</map>
```




## Image-maps

58

- Client-side image-maps may not work in all browsers
- Server-side image-maps will always work
  - Are more flexible in what they can do
  - Browser sends the positional information to server about where the user clicks within an image
  - But they are MUCH less efficient
  - Done using CGI scripts
  - Requires **<img>** tag to be inside an anchor tag and <img> must include the **ismap** attribute

```
<a href= "wwwnav.map">
      <img src="... navbar.gif" border="0"
      ismap="ismap" alt="King County Navigation Bar">
</a>
```

## Image-maps

59

- Image-maps are not always the best solution
  - Why download a LARGE image when a simple list of hyperlinks will do?
  - People can turn images off → no image-maps
  - You should always provide a text alternative
  - It is possible to slice the image in rectangular areas and piece it back together using **<table>** and **<a>** – efficient, works on more browsers, but semantically nasty

## HTML5 inline SVG

60

- What is SVG?
  - Scalable Vector Graphics
  - Vector-based graphics for the web
  - Not lose any quality if they are zoomed or resized

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

<svg width="300" height="200">
 <polygon points="100,10  40,198  190,78  10,78  160,198"
 style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>

</body>
</html>
```

## HTML Table

61

- Tables are used on web pages to organize tabular information

| Name | Birthday | Phone |
|---|---|---|
| Jack | 5/13 | 857-555-5555 |
| Sparky | 11/28 | 303-555-5555 |

- Composed of rows and columns – similar to a spreadsheet.
- Each individual table cell is at the intersection of a specific row and column.

## Tables

62

- A table can contain
  - **<caption>…</caption>** – text to display as the table's caption
  - **<thead>…</thead>** – rows to display as the table's header
  - **<tfoot>…</tfoot>** – rows to display as the table's footer
  - **<tbody>…</tbody>** – rows to display as the table's body (the content of the table)

## Tables

63

- A table row is a tag **<tr>…</tr>**
- Each row contains possibly many
  - **<th>…</th>** – table heading cells
  - **<td>…</td>** – table data cells
- Given cells can "span" multiple rows or columns
  - **colspan** = number of columns spanned
  - **rowspan** = number of rows spanned
- Cells can contain any body tags
  - Even nested tables

64


## HTML colspan Attribute

65

```
<table border="1">
  <tr>
    <td colspan="2">
    Birthday List</td>
  </tr>
  <tr>
    <td>James</td>
    <td>11/08</td>
  </tr>
  <tr>
    <td>Karen</td>
    <td>4/17</td>
  </tr>
</table>
```

| Birthday List | |
|---|---|
| James | 11/08 |
| Karen | 4/17 |

## HTML rowspan Attribute

66

```
<table border="1">
  <tr>
    <td rowspan="2">This spans two rows</td>
    <td>Row 1 Column 2</td>
  </tr>
  <tr>
    <td>Row 2 Column 2</td>
  </tr>
</table>
```

| This spans two rows | Row 1 Column 2 |
|---|---|
| | Row 2 Column 2 |

```
<h3>Now let's pretty it up a bit</h3>
<table border="0" style="table-layout:auto;empty-cells:hide">
  <caption>Example Table Caption</caption>
  <thead style="background-color:lightblue">
    <tr><td colspan="3">Header for table</td></tr>
  </thead>
  <tfoot class="footer" style="background-color:lightgreen">
        <tr><td colspan="3">Some footer information</td></tr>
  </tfoot>
  <tbody style="color:navy; background-color:lightgrey;">
    <tr>
      <th>Header</th>
      <td>Top-left Cell</td>
      <td>Top-right Cell</td>
    </tr>
    <tr>
       <th>Header</th>
      <td>Bottom-left Cell</td>
      <td>Bottom-right Cell</td>
    </tr>
    <tr><td rowspan="2" style="background-color:yellow">Double <br />Cell</td><td></td><td></td></tr>
    <tr><td></td><td></td></tr>
  </tbody>
</table>
```

Now let's pretty it up a bit

Example Table Caption

| Header for table | | |
|---|---|---|
| Header | Top-left Cell | Top-right Cell |
| Header | Bottom-left Cell | Bottom-right Cell |
| Double Cell | | |
| Some footer information | | |



## Tables and CSS



- CSS adds complex visual formatting to HTML tables
  - http://www.w3schools.com/css/css_table.asp
- Properties are passed down from **`<table>`** to **`<tbody>`** to **`<tr>`** to **`<td>`**

| Company | Contact | Country |
|---|---|---|
| Alfreds Futterkiste | Maria Anders | Germany |
| Berglunds snabbköp | Christina Berglund | Sweden |
| Centro comercial Moctezuma | Francisco Chang | Mexico |
| Ernst Handel | Roland Mendel | Austria |
| Island Trading | Helen Bennett | UK |
| Königlich Essen | Philip Cramer | Germany |
| Laughing Bacchus Winecellars | Yoshi Tannamuri | Canada |
| Magazzini Alimentari Riuniti | Giovanni Rovelli | Italy |
| North/South | Simon Crowther | UK |
| Paris spécialités | Marie Bertrand | France |
| The Big Cheese | Liz Nixon | USA |
| Vaffeljernet | Palle Ibsen | Denmark |

## Tables and CSS



- **`width`** and **`height`** suggest the *minimum* size for a row or column, but not necessarily the maximum

```
table {
    width: 100%;
}
```

sets the width of the table to 100%, and the height of the th elements to 50px:

```
th {
    height: 50px;
}
```

- **`border:`** used to specify table border.

specifies a black border for table, th, and td elements:

```
table, th, td {
    border: 1px solid black;
}
```

## Tables and CSS



- **`border-style:`**
  - As before, except…
  - **`hidden`** = **`none`** but overrides adjacent styles
  - **`inset`**/**`outset`** = **`ridge`**/**`groove`** if **`collapse`**

## Tables and CSS



```
<style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style>
```

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

## Tables and CSS



- **`empty-cells:`**
  - **`show`** | **`hide`**

| Peter | Griffin |
|---|---|
| Lois | |

| Peter | Griffin |
|---|---|
| Lois | |

- **`table-layout:`**
  - **`auto`** – adjust the width of each column to accommodate the content of all cells
  - **`fixed`** – set the width of each column by the cells of the first row only

table-layout: auto:

100000000000000000000000000 10000000

table-layout: fixed:

100 0000000000000000000000000

## Tables and CSS

73

- **`overflow:`**
  - What to do when **`table-layout: fixed`**
  - **`visible | hidden | scroll | auto`**

table-layout: fixed and overflow hidden

| 100 | 10000000 |
|---|---|

table-layout: fixed and overflow scroll

| 1 | 10000000 |
|---|---|

## Tables and CSS

74

- **`Changing the Width of columns:`**
- <col style="width:40%">
  <col style="width:30%">
  <col style="width:30%">

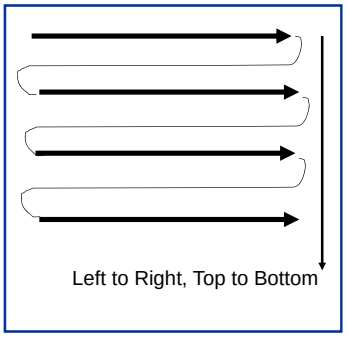
## CSS Positioning

75

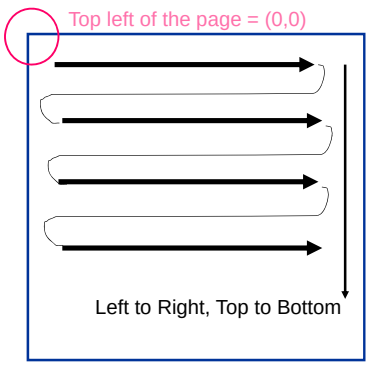
- CSS provides three “positioning schemes” to control the placement of blocks…
  - http://www.w3.org/TR/CSS2/visuren.html#positioning-scheme

1. Normal flow – blocks are positioned in sequence, allowing for size and margins
2. Floats – the block is shifted to the left or right margin and other content flows around it
3. Absolute – the block is positioned relative to the “page” and has no interaction with other blocks

## Normal Flow – no “positioning”




## Normal Flow – no “positioning”




## Normal Flow

This is a paragraph to which I have set the width.

If the next paragraph fits next to it on the right, it will line up.

## Normal Flow

This is a paragraph to which I have set the width.

However, if the second paragraph is too wide to fit the container, it will shift down.

## Normal Flow

This is a paragraph to which I have set the width.

However, if the second paragraph is too wide to fit the container, it will shift down.

This is the basic principle of Normal Flow

## Float

HTML

```
<div>
<p> This is the normal…</p>
<p class="float">This text is floated right.</p>
```

CSS

```
.float {float:right;}
```

This is the normal flow of a document; from top to bottom, left to right. When the floated text is added, it moves to the top right corner of the containing element, in this case the . Normal text flows around the floated text.

**This text is floated right.**

## Absolute

HTML

```
<div>
<p> This is the normal… <span class="abs"> This text is absolutely positioned.</span> …top to bottom…</p>
</div>
```

CSS

```
.abs {position: absolute;
      top: 40px;
      left: 80;}
```

This is the normal flow of a document; from top to bottom, left to right. When you add the absolutely positioned text, it moves to the coordinates you set based on the top left corner of the containing element, in this case the . Normal text flows over the absolutely positioned text. There is no gap where the text is taken from.

This text is absolutely positioned.

This text is relatively positioned.

## Relative

HTML

```
<div>
<p> This is the normal… <span class="rel"> This text is relatively positioned. </span> … from top to bottom…</p>
</div>
```

CSS

```
.rel {position: relative;
      top: -50px;
      left: -150px}
```

This is the normal flow of a document; from top to bottom, left to right. When you add the relatively positioned text, it moves to the coordinates you set based on the top left corner of the containing element, in this case the . Normal text flows as normal, but a gap is left where the relative text used to be, and the text overlaps with the newly positioned relative text if they are in the same area.

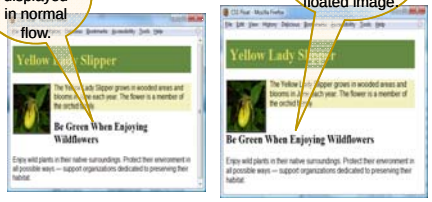
## Clear Property

84

- Useful to "clear" or terminate a float
- Values are left, right, and both

The h2 text is displayed in normal flow.

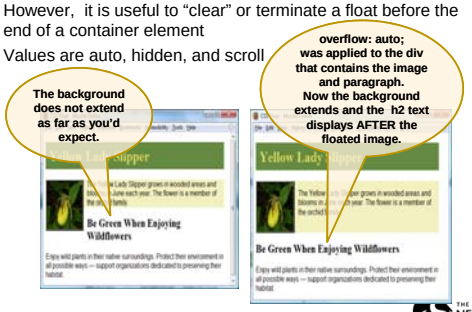
clear: left; was applied to the h2. Now the h2 text displays AFTER the floated image.



## Overflow Property

85

- Intended to configure the display of elements on a Web page.
- However, it is useful to "clear" or terminate a float before the end of a container element
- Values are auto, hidden, and scroll




## CSS Debugging Tips

86

- Manually check syntax errors
- Use W3C CSS Validator to check syntax errors
  - http://jigsaw.w3.org/css-validator/
- Configure temporary background colors
- Configure temporary borders
- Use CSS comments to find the unexpected
  /* the browser ignores this code */
- Don't expect your pages to look exactly the same in all browsers!
- Be patient!

## Checking browser support

87

- CanIUse.com
- They test almost every single one of the new features and the test the compatibility a long way back, version-wise, and on every mainstream browser.

## Summary

88

- What are the semantics structural elements in HTML5
  - *<header>, <footer>,<nav>, <main>, <article>,<section>,<aside>,*
  - *<details>, <figure>, <figcaption>,<mark>,<progress>,<summary>,<time>*
- HTML5 *<audio>,<video>* elements
- HTML5 *<form>* element
  - Different types of inputs in form element
  - Methods for submitting a form
  - New attributes in *<form>* element
  - New input types for *<form>* element

## Summary

89

- Image maps be effective for linking to related data from an HTML page
- How to use tables to organize your data
- What are the components of HTML table: ***thead, tbody, tr, td, th, tfoot, caption***
- How to design complex tables: ***rowspan, colspan***
- How CSS can be used to make tables more interesting and exciting
- CSS provides different "positioning schemes" to control the placement of blocks: fixed, relative, absolute

## References

90

- **Web Development and Design Foundations with HTML5 (6e)**
  *By Terry Felke-Morris*
  - *Chapter 6: Section 6.11*
  - *Chapter 8, 9*
- http://www.w3schools.com/html/html5_intro.asp