

# Server Side Web Development

## Lecture 2 – HTML Refresher

Lecturer: Mr Daryl Parker ([dparker@ait.ie](mailto:dparker@ait.ie))

- Web pages are written in HyperText Markup Language.
- With HTML
  - formatting information is included in text documents.
  - hyperlinks are also embedded in the document as text.
- HTML 4.01 - W3C Recommendation 24 Dec 1999.
  - Available at
  - <http://www.w3.org/TR/html401>
- [HTML5 now released.]
  - <http://www.w3.org/TR/html5/>



Covering HTML4  
only

# HTML Example

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Introduction to HTML</title>
  </head>
  <body bgcolor="#FFFFFF" text="#000000">
    <h2>HTML</h2>
    Web pages are written in HyperText Markup Language.
    With HTML,
    <ul>
      <li>formatting information is included in text documents.
      <li>hyperlinks are also embedded in the document as text
    </ul>
    HTML documents are viewed using Web browsers such
    as Chrome.
  </body>
</html>
```

## HTML

Web pages are written in HyperText Markup Language. With HTML,

- formatting information is included in text documents.
- hyperlinks are also embedded in the document as text

HTML documents are viewed using Web browsers such as Chrome.

- Web pages are written in HyperText Markup Language.
- With HTML
  - formatting information is included in text documents.
  - hyperlinks are also embedded in the document as text.
- HTML 4.01 - W3C Recommendation 24 Dec 1999.
  - Available at
  - <http://www.w3.org/TR/html401>
- [HTML5 now released.]
  - <http://www.w3.org/TR/html5/>

- Examples of elements in HTML are
  - paragraphs, lists hypertext links, tables etc. etc.
- Elements appear in an HTML document as
  - <tagname> content </tagname>



- Sometimes end tags can be omitted.
  - Depends on DOCTYPE
    - <br> correct for HTML, <br /> correct for XHTML
- Sometimes there is no content (e.g. line break <br>)
- We have already seen <html>, <title>, <body> ....

# HTML Attributes

- Elements have associated properties called attributes, which may have values.
- Attribute/value pairs appear before the “>” of the start tag.
- Attribute value pairs are separated by spaces and can appear in any order.
- Attribute values can be delimited using either single or double quotes.

```
<body bgcolor="#FFFFFF" text="#000000">
```

# HTML Attributes

- If the attribute value contains only alphanumeric, hyphens, periods, underscores and colons, the quotes can be omitted. **XHTML requires quotes always**
- The HTML specification recommends always using quotes.

**Safest is to always use quotes!**

# HTML Global Attributes

- A set of attributes that can be used on **any** HTML element.

Attribute	Description
<u>accesskey</u>	Specifies a shortcut key to activate/focus an element
<u>class</u>	Specifies one or more classnames for an element (refers to a class in a style sheet)
<u>contenteditable</u>	Specifies whether the content of an element is editable or not
<u>data-*</u>	Used to store custom data private to the page or application
<u>dir</u>	Specifies the text direction for the content in an element
<u>draggable</u>	Specifies whether an element is draggable or not
<u>dropzone</u>	Specifies whether the dragged data is copied, moved, or linked, when dropped
<u>hidden</u>	Specifies that an element is not yet, or is no longer, relevant
<u>id</u>	Specifies a unique id for an element
<u>lang</u>	Specifies the language of the element's content
<u>spellcheck</u>	Specifies whether the element is to have its spelling and grammar checked or not
<u>style</u>	Specifies an inline CSS style for an element
<u>tabindex</u>	Specifies the tabbing order of an element
<u>title</u>	Specifies extra information about an element
<u>translate</u>	Specifies whether the content of an element should be translated or not

- Comments start with special `<!--` and end with `-->` tags
- Everything between these tags is considered a comment even over multiple lines.

```
<!-- this is a comment -->
<head>
|   <title>Introduction to HTML</title>
</head>
<!-- this is also a comment
over multiple lines -->
<body bgcolor="#FFFFFF" text="#000000">
```

- Highlight and **CTRL – SHIFT – C** in Eclipse will put in comments. (**Most IDE's with have similar shortcuts**)

- Colours are represented as six digit hexadecimal numbers preceded by # character.
  - E.g #FFFFFF, #09AB4E
- These are actually three 2 digit numbers representing the amount of Red, Green and Blue in the colour.
- There are also a set of well known colour names
  - [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)
- Lots of online (and IDE integrated) tools to help choose/visualise HEX colour values
  - [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)

- The `<body>` tag defines the document's body.
- The `<body>` element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.
- Attributes
  - **background** - The value of this attribute is a URL that designates an image.
  - **color** – The color of the background for the document.
  - **text** - This attribute sets the foreground color for text
  - **alink** – The color of the active hypertext link
  - **link** - This attribute sets the color of text marking unvisited hypertext links
  - **vlink** - This attribute sets the color of text marking visited hypertext links

# HTML Div Element

- The <div> tag defines a division or a section in an HTML document.
- The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript
- Attributes
  - align – Specifies the alignment of the content inside the <div> element. Valid values (left, right, center, justify)

```
<div style="background-color:lightblue">
    <h3>This is a heading</h3>
    <p>This is a paragraph.</p>
</div>
```

**Tip:** The <div> element is very often used together with CSS, to layout a web page.

- Headings
  - There are six levels of headings from h1 to h6.
  - <h1>Text of heading</h1>
  - Start and end tag are required
- Paragraphs
  - <p> Text of paragraph </p>
  - End tag is optional
- Line Break has no end tag.
  - <br>
- Horizontal Rule (line) has no end tag.
  - <hr>

**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

**This is heading 6**

## **HTML**

HTML is a language for describing web pages.

## **css**

CSS defines how to display HTML elements.

# HTML Ordered Lists

- The `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.
- Use the `<li>` tag to define list items.

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

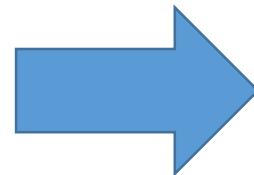
1. Coffee  
2. Tea  
3. Milk

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

50. Coffee  
51. Tea  
52. Milk

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

I. Coffee  
II. Tea  
III. Milk

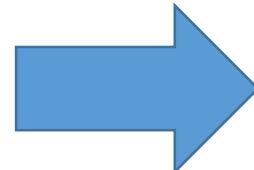


# HTML Unordered Lists

- The `<ul>` tag defines an unordered (bulleted) list.
- Use the `<li>` tag to define list items.

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<ul type="square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



```
<ul type="circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

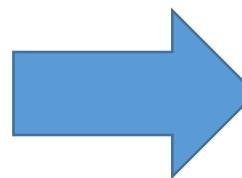
- Coffee
  - Tea
  - Milk
- 
- Coffee
  - Tea
  - Milk
- 
- Coffee
  - Tea
  - Milk

# HTML Images

- The `<img>` tag defines an image in an HTML page.
- The `<img>` tag has two required attributes: `src` and `alt`.
- **Note:** Images are not technically inserted into an HTML page, images are linked to HTML pages. The `<img>` tag creates a holding space for the referenced image.
- **Tip:** To link an image to another document, simply nest the `<img>` tag inside `<a>` tags.

```

```



```
<p>
An image that is a link:
<a href="https://www.w3schools.com">

width="42" height="42" border="0">
</a>
</p>
```

An image that is a link:



**The Image is actually the link**

- A connection from one Web resource to another.
- The link starts at the “source” anchor and points to the “destination” anchor, which may be any Web resource (e.g., an image, a video clip, a sound bite, a program, an HTML document, an element within an HTML document, etc.).
- The default behavior associated with a link is the retrieval of another Web resource.
- This is activated by clicking on the link (associated text or image.)

- The `<a>` tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the `<a>` element is the **href** attribute, which indicates the link's destination.
- By default, links will appear as follows in all browsers:
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red

# HTML Links (Anchor Tags)

```
<h2 id="top">Top of page!</h2>

<a href="anotherFile.html">Text for the link to file in same dir</a>

<a href="dir1/anotherFile.html">Text for the link to file in different dir</a>

<a href="http://java.sun.com/jdk/1.6/">JDK Download Page</a>

<a href="http://java.sun.com/jdk/1.6/"> <img src = "pic.gif"></a>

<a href="#top">Link to page element with id top</a>

<a href="https://www.w3schools.com" target="_blank">Visit W3Schools.com!</a>

<a href="mailto:someone@example.com?Subject=Hello" target="_top">Send Mail</a>

<a href="tel:+4733378901">+47 333 78 901</a>
```

- The `<table>` tag defines an HTML table.
- An HTML table consists of the `<table>` element and one or more `<tr>`, `<th>`, and `<td>` elements.
  - The `<tr>` element defines a table row
  - The `<th>` element defines a table header
  - The `<td>` element defines a table cell.
- A more complex HTML table may also include `<caption>`, `<col>`, `<colgroup>`, `<thead>`, `<tfoot>`, and `<tbody>` elements.

**Note:** Tables should not be used for page layout!

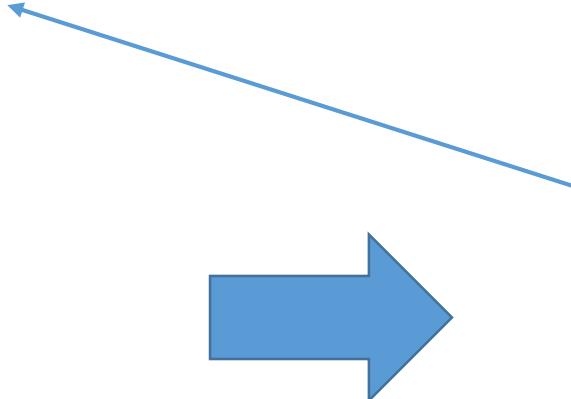
Historically some Web authors misused tables in HTML as a way to control their page layout, the recommended approach is to use CSS.

# HTML Tables

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>

<table>
    <tr>
        <th>Month</th>
        <th>Savings</th>
    </tr>
    <tr>
        <td>January</td>
        <td>$100</td>
    </tr>
    <tr>
        <td>February</td>
        <td>$80</td>
    </tr>
</table>

</body>
</html>
```



Month	Savings
January	\$100
February	\$80

**With the style applied to table, th, td**

Month	Savings
January	\$100
February	\$80

**Without the style**

- Write a Web page of your choice in HTML that must include:
  - headings
  - paragraphs
  - lists (ordered and unordered)
  - links
  - a table
- Look up the HTML specification to see how a table is constructed in HTML.

[https://www.w3schools.com/tags/tag\\_table.asp](https://www.w3schools.com/tags/tag_table.asp)

# HTTP Forms

- An HTML form is a section of a document containing:
  - normal content,
  - markup,
  - special elements called controls (checkboxes, radio buttons, menus, etc.)
  - and labels on those controls.

- Users generally complete a form by:
  - modifying its controls (entering text, selecting menu items, etc.)
  - before submitting the form to an agent for processing (e.g., to a Web server, to a mail server, etc.)
- See ch. 17 of HTML 4.01 Specification [here](#)

- HTML forms are used support data entry on the client side (browser).
- On the client side, all data entered is packed into a HTTP message and a (GET or) POST method is invoked.
- This data is sent to the Web server.
- The data is handled either by a form processing agent such as a Servlet or Grails controller.

# HTML Forms Example

```
<!DOCTYPE HTML>
<html>
<head>
    <title>Important Decision</title>
</head>
<body bgcolor="#FFFFFF">
    <form method="post" action="user/create">
        <p>Enter your name:<input type="text" name="name" size=40></p>
        <p>Email Address :<input type="text" name="emailAddress" size=40></p>
        <p>Tel:<input type="text" name="telNumber" size=20></p>
        <p>Which of the following have you heard of</p>
        <p>
            <input type="checkbox" name="linus" value="Linus"> Linus Torvalds<br>
            <input type="checkbox" name="larry" value="Larry"> Larry Page<br>
            <input type="checkbox" name="bill" value="Bill"> Bill Gates<br>
        </p>
        <p>What do you want to download</p>
        <select name=software>
            <option> Git
            <option> Windows 42
            <option> Google Chrome
        </select>
        <p> Additional Information:</p>
        <p><textarea name="message" rows=7 cols=50></textarea></p>
        <p><input type="submit" value="Submit">
        <p><input type="reset" value="Reset and clear">
    </form>
</body>
</html>
```

# HTML Forms Example

Enter your name:

Email Address :

Tel:

Which of the following have you heard of

- Linus Torvalds
- Larry Page
- Bill Gates

What do you want to download

▾

Additional Information:

- Controls occur inside a <FORM>...</FORM> element
- Users interact with forms through named controls.
- The name of a control is given by the “name” attribute.
- A control has an initial value given by the “value” attribute.
- A control has a current value, defined by the user interaction.

# HTML Form Control Types



- buttons
- checkboxes
- radio buttons
- menus
- text input
- file select
- hidden controls
- object controls

- <form **method**=“post” **action**= “user/create”>
- Attributes
  - method = <get|post> (use post)
  - action = <form-processing-agent>

- <input type="text" name="emailAddress" size=40>
- Attributes
  - type = <text | password | checkbox | radio | submit | reset | file | hidden | image | button>
  - name= <control-name>
  - value=<initial-value>
  - placeholder=short hint that describes the expected value of an <input> element
  - required=Specifies that an input field must be filled out before submitting the form
  - size

# HTML Form Input Types

`<input type="text">` defines a **one-line text input field**

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```



First name:

Last name:

# HTML Form Input Types

`<input type="password">` defines a **password field**:

```
<form>  
    User name:<br>  
    <input type="text" name="username"><br>  
    User password:<br>  
    <input type="password" name="psw">  
</form>
```



User name:

Joe

User password:

••••

# HTML Form Input Types

`<input type="radio">` defines a **radio button**.

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```



- Male
- Female
- Other

# HTML Form Input Types

<input type="checkbox"> defines a **checkbox**.

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```



- I have a bike
- I have a car

# HTML Form Input Types

<input type="button"> defines a **button**:

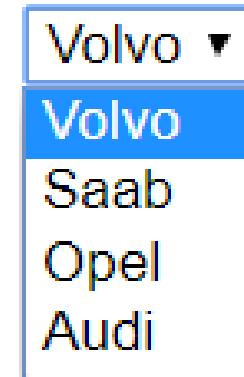
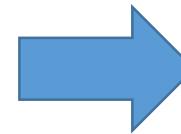
```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```



Click Me!

- The `<select>` element is used to create a drop-down list.
- The `<option>` tags inside the `<select>` element define the available options in the list.

```
<select>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```



The image shows a visual representation of a dropdown menu. On the left, there is a blue rectangular button with the word "Volvo" in white and a small downward arrow. To the right of this button is a vertical list of four items, each in a light blue box: "Volvo", "Saab", "Opel", and "Audi". The word "Volvo" is also in white, indicating it is the selected option.

# HTML Form Submission

- “The method attribute of the FORM element specifies the HTTP method used to send the form to the processing agent. This attribute may take two values:
- get: With the HTTP "get" method, the form data set is appended to the URI specified by the action attribute (with a question-mark ("?") as separator) and this new URI is sent to the processing agent.

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



Enter your details!

First name:

Last name:

<http://localhost/PHPEX1Soln/display.php?fname=Daryl&lname=Parker>

# HTML Form Submission

- post: With the HTTP "post" method, the form data set is included in the body of the form and sent to the processing agent."

```
<form action="display.php" method="post">
    <label>First name:</label> <input type="text" name="fname"><br/>
    <label>Last name:</label><input type="text" name="lname"><br/>
    <input type="submit" value="Submit">
</form>
```

Enter your details!

First name: Daryl  
Last name: Parker  
Submit



Form Data Set (in HTTP Request body)  
fname=Daryl&lname=Parker

<http://localhost/PHPEX1Soln/display.php>

# HTML Form Exercise

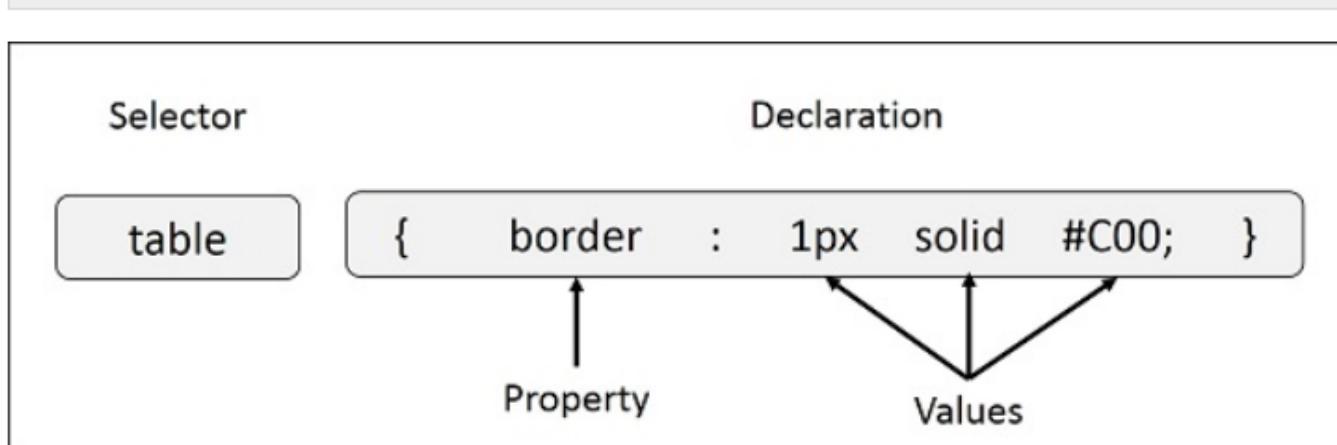


- Design two forms for your Web application.
- One form should be used to enter data.
- The other form should be used to support a query of the data.

# CSS Basics

- A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document.
- A style rule is made of three parts –
  - **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like `<h1>` or `<table>` etc.
  - **Property** – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be `color`, `border` etc.
  - **Value** – Values are assigned to properties. For example, `color` property can have value either `red` or `#F1F1F1` etc.

```
selector { property: value }
```



# CSS Selectors

## The Type Selectors

- This selector uses selects those elements matching the tag type. For example to give a color to all level 1 headings. i.e. Tag <h1>

```
h1 {  
    color: #36CFFF;  
}
```

## The Universal Selectors

- Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

```
* {  
    color: #000000;  
}
```

This rule renders the content of every element in our document in black.

## The Descendant Selectors

- Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to `<em>` element only when it lies inside `<ul>` tag.

```
ul em {  
    color: #000000;  
}
```

## The Class Selectors

- You can define style rules based on the `class` attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {  
    color: #000000;  
}
```

This rule renders the content in black for every element with `class` attribute set to `black` in our document.

## The Class Selectors Cont.

- You can make the class selector a bit more specific. For example –

```
h1.black {  
    color: #000000;  
}
```

This rule renders the content in black for only `<h1>` elements with class attribute set to ***black***.

**NOTE:** You can apply more than one class selector to given element. Consider the following example –

```
<p class = "center bold">  
    This para will be styled by the classes center and bold.  
</p>
```

## The ID Selectors

- You can define style rules based on the ***id*** attribute of the elements. All the elements having that ***id*** will be formatted according to the defined rule.

```
#black {  
    color: #000000;  
}
```

This rule renders the content in black for every element with ***id*** attribute set to ***black*** in our document.

Similar to class selectors they can be made more specific. For example -

```
h1#black {  
    color: #000000;  
}
```

This rule renders the content in black for only **<h1>** elements with ***id*** attribute set to ***black***.

# CSS Selectors - Multiple Style Rules



- You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example –.

```
h1 {  
    color: #36C;  
    font-weight: normal;  
    letter-spacing: .4em;  
    margin-bottom: 1em;  
    text-transform: lowercase;  
}
```

Here all the property and value pairs are separated by a **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, its recommended to keep them in separate lines.

# CSS Selectors – Grouping Selectors



- You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example –

```
h1, h2, h3 {  
    color: #36C;  
    font-weight: normal;  
    letter-spacing: .4em;  
    margin-bottom: 1em;  
    text-transform: lowercase;  
}
```

- This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.
- You can combine the various *id* selectors together as shown below –

```
#content, #footer, #supplement {  
    position: absolute;  
    left: 510px;  
    width: 200px;  
}
```