



# DEEPer

Intro & HTML  
Week 1 Session 1

# TODAY'S SESSION

2



01

**Introduction**

02

**HTML**

# Intro

---

# 01

# Lectures

- Two half days weekly
- 9:30 – 12:30 Monday & Thursday
- 10 minute breaks each hour - flexible
- Lectures will feature;
  - Traditional lecture-style slides
  - Quizzes
  - Live-coding delivered by lecturer
- PDF of exported slides will be shared in Slack
- Recordings will be made available for review shortly after lectures

# Project

- One single project developed throughout the course
- Check-in type application
- Iterated upon each week to involve new learning
- Domain can be determined by you
- Most workbooks will include generic requirements for functionality, with scope to fit your individual projects

# Workbooks

- PDF delivered weekly via Slack
- Contain a minimum of one of;
  - Standalone exercises related to the week's content – e.g. additional reading and tasks
  - New requirements provided for the project

# Mentoring & Support

- Sessions held weekly in small groups, with a course representative
- Opportunity to raise issues related to both the course and the content
- Encouragement for peer-support – helping each other is one of the best ways to solidify understanding

# Development Environment Walkthrough

---

- Next we will demonstrate the development environments
- This section may prove useful to revisit in the recording



02

# HTML

---

Hypertext Markup Language

# What is HTML

- Markup language used to form the structural content of a web page
- A wide range of **tags**, each with their own semantics
- Pieced together to form a single document hierarchy
- Tags exist for all common elements of a web page, such as headings, images, links and body text
- For now, all HTML we write will be added to a file with a `.html` extension

# What is HTML

- Browsers apply default styling to all HTML elements
- Some styling can vary wildly between browsers
- Styling comes later, for now we only care about structure

# HTML Syntax

```
<!-- HTML Comment -->
```

```
<tag-name attribute="value">
```

```
    Some content
```

```
</tag-name>
```

```
<self-closing-tag attribute="value">
```

# Page Foundations

- Most HTML documents begin with an identical structure
- Each of the following tags appears once within a single document
- Like **many** topics in this course, understanding this syntax is the important part, not remembering it

`<!DOCTYPE html>` – Informs the browser what type of document follows

`<html></html>` – Root element in the HTML document's hierarchy

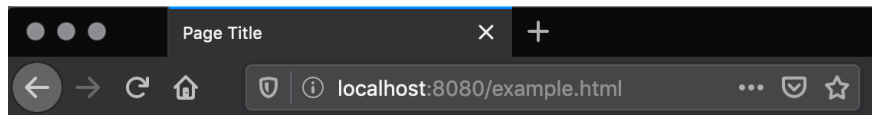
`<head></head>` – Container for document metadata, generally not visible in the page

`<title></title>` – The title that appears in the browser tab and bookmarks

`<body></body>` – Container for the visible content of an HTML document

# Page Foundations

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



Hello World

# Headings

- Denote page or section headings and subheadings
- Six heading tags exist - h1-h6, ranked by relative importance
- A single page should contain only one h1, but can contain multiple h2-h6 elements
- Levels should not be skipped when forming a heading hierarchy
- Headings should be selected based on their position in the hierarchy, not how they appear (e.g. their size)

# Headings – Tags

`<h1>Heading 1</h1>`

`<h2>Heading 2</h2>`

`<h3>Heading 3</h3>`

`<h4>Heading 4</h4>`

`<h5>Heading 5</h5>`

`<h6>Heading 6</h6>`

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6



# Headings – Example

```
...  
<body>  
  <h1>Animal Directory</h1>  
  <h2>Invertebrates</h2>  
    <h3>Octopus</h3>  
    <h3>Lobster</h3>  
  
  <h2>Vertebrates</h2>  
    <h3>Mammal</h3>  
    <h3>Fish</h3>  
</body>  
...
```

## Animal Directory

### Invertebrates

Octopus

Lobster

### Vertebrates

Mammal

Fish

# Text Content – Paragraphs

```
<p>This is some regular body text...</p>
```

This is some regular body text...

# Text Content – Unordered lists

- Milk
- Cheese
  - Blue cheese
  - Feta

```
<ul>
  <li>Milk</li>
  <li>Cheese
    <ul>
      <li>Blue cheese</li>
      <li>Feta</li>
    </ul>
  </li>
</ul>
```

# Text Content – Ordered lists

1. Mix flour, baking powder, sugar, and salt.
2. In another bowl, mix eggs, milk, and oil.
3. Stir both mixtures together.
4. Fill muffin tray 3/4 full.
5. Bake for 20 minutes.

```
<ol>
  <li>Mix flour, baking powder, sugar, and salt.</li>
  <li>In another bowl, mix eggs, milk, and oil.</li>
  <li>Stir both mixtures together.</li>
  <li>Fill muffin tray 3/4 full.</li>
  <li>Bake for 20 minutes.</li>
</ol>
```

# Text Content – Quotes

Words can be like X-rays, if you use them properly—they'll go through anything.

*Brave New World*

```
<blockquote cite="https://www.huxley.net/bnw/four.html">
  <p>
    Words can be like X-rays, if you use them properly—they'll go
    through anything.
  </p>
  <cite>Brave New World</cite>
</blockquote>
```

# Text Content – Links

## Mozilla Web Docs

*<!-- Open in same tab -->*

```
<a href="https://developer.mozilla.org/">
```

Mozilla Web Docs

```
</a>
```

*<!-- target of \_blank opens in new tab -->*

```
<a href="https://developer.mozilla.org/" target="_blank">
```

Mozilla Web Docs

```
</a>
```

# Text Content – Links

Need to contact us?

```
<a href="#contact-us">Need to contact us?</a>
```

```
<a href="https://example.com/my-page#contact-us">Need to contact us?</a>
```

```
<!-- Other content... -->
```

```
<h2 id="contact-us">Contact Us</h2>
```

# Text Content – Formatting/Tone

- A number of text elements exist to provide formatting or tone to text
- Each element has default styling within browsers
- When selecting which tag to use, consider the semantics not the styling



# Text Content – Formatting/Tone

Remember, elements should be selected for their semantic meaning **not their default styling!**

```
<!-- <strong/> - indicates importance/seriousness/urgency -->  
<p>Remember, elements should be selected for their semantic  
meaning <strong>not their default styling!</strong></p>
```

# Text Content – Formatting/Tone

Jimmy, go to bed *now!*

`<!-- <em/> - indicates stress emphasis -->`

`<p>Jimmy, go to bed <em>now!</em></p>`

# Text Content – Formatting/Tone

Search results for "salamander":

Several species of **salamander** inhabit the...

*<!-- <mark/> - used to highlight text for notation purposes -->*

`<p>Search results for "salamander":</p>`

`<p>Several species of <mark>salamander</mark> inhabit the...</p>`

# Text Content – Example

## Fish

Fish live in water, **they cannot breathe air**. Examples of fish include;

- Carp
- Trout
- Salmon

```
<h3>Fish</h3>
```

```
<p>
```

Fish live in water, **they cannot breathe air**.

Examples of fish include;

```
</p>
```

```
<ul>
```

```
<li>Carp</li>
```

```
<li>Trout</li>
```

```
<li>Salmon</li>
```

```
</ul>
```

# Media - Images

```

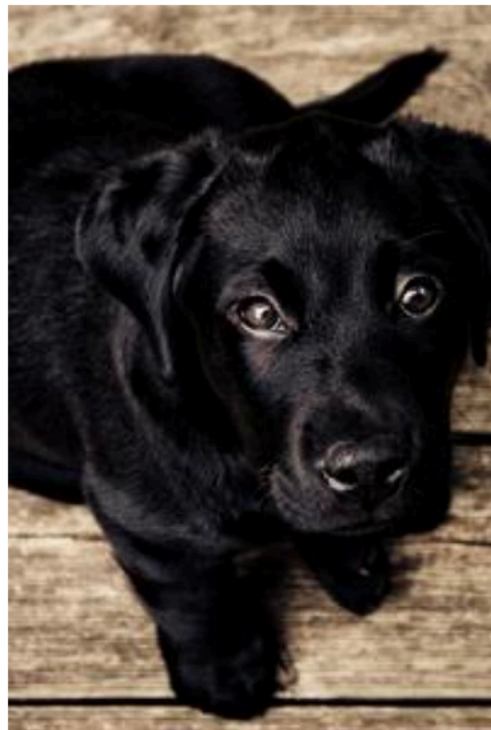
```

- The mandatory **src** attribute contains the path to the image file
- The **alt** attribute contains a textual description of the image, used for screen readers or if the image fails to load. Not mandatory, **but very important!**



# Media - Figures

```
<figure>
  
  <figcaption>A small black puppy</figcaption>
</figure>
```



A small black puppy

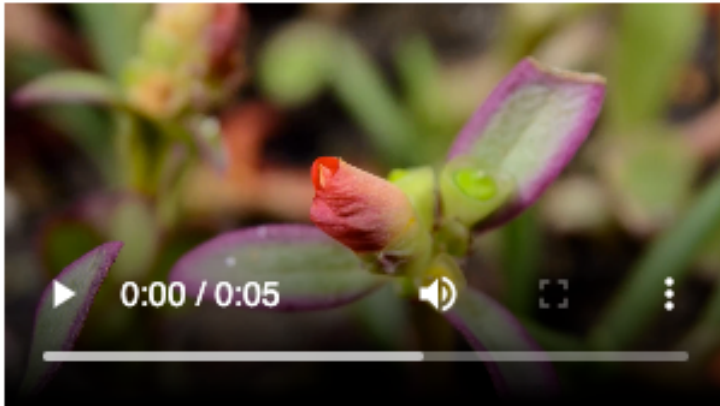
# Media - Videos

```
<video controls width="250">  
  <source src="/path/to/video.webm" type="video/webm">
```

```
  <source src="/path/to/video.mp4" type="video/mp4">
```

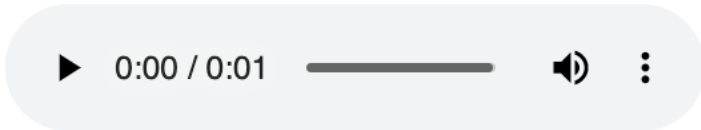
Sorry, your browser doesn't support embedded videos.

```
</video>
```



# Media - Audio

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





# Tables

```
<table>
  <thead>
    <tr>
      <th>Forename</th>
      <th>Surname</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Joe</td>
      <td>Bloggs</td>
    </tr>
    <tr>
      <td>Jane</td>
      <td>Doe</td>
    </tr>
  </tbody>
</table>
```

- **<thead/>** – wraps the header row(s)
- **<tbody/>** – wraps the body row(s)
- **<tfoot/>** – wraps the footer row(s)
- **<tr/>** – wraps a single table row's cells
- **<td/>** – a single body cell
- **<th/>** – a single header cell

Forename	Surname
Joe	Bloggs
Jane	Doe

# Tables - Colspan

```
<table>
  <thead>
    <tr>
      <th colspan="2">2 Column Heading</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Column 1</td>
      <td>Column 2</td>
    </tr>
  </tbody>
</table>
```

## 2 Column Heading

2 Column Heading	
Column 1	Column 2

# Tables - Example

## Invoice

Item	Quantity	Price (£ each)	Price (£ total)
Pencil	10	0.25	2.50
Eraser	8	0.50	4.00
Total			£6.50

*<!-- Elements to consider -->*

`<table>` `<thead>` `<tbody>` `<tfoot>` `<th colspan>` `<td colspan>`

# Tables - Example

```
<table>
  <thead>
    <tr>
      <th colspan="4">Invoice</th>
    </tr>
    <tr>
      <th>Item</th>
      <th>Quantity</th>
      <th>Price (£ each)</th>
      <th>Price (£ total)</th>
    </tr>
  </thead>
  ...
```

## Invoice

Item	Quantity	Price (£ each)	Price (£ total)
Pencil	10	0.25	2.50
Eraser	8	0.50	4.00
Total			£6.50

# Tables - Example

```
...  
<tbody>  
  <tr>  
    <td>Pencil</td>  
    <td>10</td>  
    <td>0.25</td>  
    <td>2.50</td>  
  </tr>  
  <tr>  
    <td>Eraser</td>  
    <td>8</td>  
    <td>0.50</td>  
    <td>4.00</td>  
  </tr>  
</tbody>  
...
```

Invoice

Item	Quantity	Price (£ each)	Price (£ total)
Pencil	10	0.25	2.50
Eraser	8	0.50	4.00
Total			£6.50

# Tables - Example

```
...  
<tfoot>  
  <tr>  
    <td colspan="3">Total</td>  
    <td>£6.50</td>  
  </tr>  
</tfoot>  
</table>
```

Invoice

Item	Quantity	Price (£ each)	Price (£ total)
Pencil	10	0.25	2.50
Eraser	8	0.50	4.00
Total			£6.50

# Forms

```
<form action="search-results.html" method="GET">  
  ...  
</form>
```

- The **<form/>** tag wraps all inputs of a single form
- **action** determines the URL which processes the form's submission
- **method** sets which HTTP method to submit with, usually **GET** or **POST**
  - **GET** – submits form values through the URL, can be distributed or bookmarked but must not contain sensitive data
  - **POST** – submits data via the request body

# Forms – Controls

```
<input type="text" name="username">
```

- The **<input>** tag renders an input or control field
- **name** is used as the key for the entered value in the form's submission
- **type** is the type of form control to render
- HTML5 introduced a large number of contextual input types



# Forms – Common Control Types

```
<input type="text" name="username">  
<input type="checkbox" name="termsAgreed" checked>  
<input type="email" name="email">  
<input type="file" name="profilePicture">  
<input type="radio" name="favouriteColour" value="red">
```



No file selected.



# Forms – Common Control Types

```
<textarea placeholder="Enter a bio" name="bio"></textarea>  
<select name="colour">  
  <option value>--Select a colour--</option>  
  <option value="red">Red</option>  
  <option value="blue">Blue</option>  
  <option value="green">Green</option>  
</select>
```

Enter a bio

--Select a colour--

# Forms – Checkboxes

- In many cases we want to be able to select multiple values from checkbox fields and submit their values as a collection
- e.g. selecting toppings to include on a pizza
- This is achieved by adding [] to the end of the name attribute

```
Ham <input type="checkbox" name="toppings[]" value="ham">  
Pepperoni <input type="checkbox" name="toppings[]" value="pepperoni">  
Sweetcorn <input type="checkbox" name="toppings[]" value="sweetcorn">
```

Ham ☒ Pepperoni ☐ Sweetcorn ☒



localhost:8080/example.html?toppings[]=ham&toppings[]=sweetcorn

# Forms - Labels

- Every form control should have a corresponding `<label/>`
- The label's `for` attribute pairs with the field's `id`
- Wrapping radio and checkbox inputs in labels allows the label to be clicked

```
<label for="username">Username:</label>  
<input id="username" type="text" name="username">
```

**Username:**

# Forms - Buttons

- Buttons are rendered using the `<button/>` tag
- Buttons should have a `type` attribute
  - `submit` – submits the form the button exists within
  - `button` – does not submit a form, used for custom interactions

```
<button type="submit">Log In</button>
```



LOG IN

# Forms - Buttons

- Buttons are rendered using the `<button/>` tag
- Buttons should have a `type` attribute
  - `submit` – submits the form the button exists within
  - `button` – does not submit a form, used for custom interactions

```
<button type="submit">Log In</button>
```



LOG IN

# Forms - Validation

- HTML5 introduces some basic native form validation
- Specialist input types such as email include automatically applied validation rules
- Adding the **required** attribute to a field marks it as mandatory
- Examples of other attributes;
  - **pattern=" [A-Za-z] {3} "**
  - **minlength="6"**
  - **maxlength="20"**
- It is possible to write CSS rules that only target valid or invalid fields, more on this in the CSS section

# Forms – Login Example

```
<form action="/my/application/login" method="POST">
  <div>
    <label for="username">Username:</label>
    <input id="username" type="text" name="username">
  </div>

  <div>
    <label for="password">Password:</label>
    <input id="password" type="password" name="password">
  </div>

  <button type="submit">Log In</button>
</form>
```

Username:

Password:

LOG IN



# Forms – Example

## Create a Book

Book Title:

Summary:

Released?

## Genres

Horror ☐

Fantasy ☐

Biography ☐

CREATE BOOK

```
<!-- Elements to consider -->
<form action method>
<label for>
<input type="text|checkbox|radio" name id>
<textarea name id>
<select name id>
<option value>
<button type>
<div>
```

# Forms – Example

```
<form action="/path/to/create-book" method="POST">
  <h1>Create a Book</h1>

  <div>
    <label for="title">Book Title:</label>
    <input id="title" type="text" name="title">
  </div>

  <div>
    <label for="summary">Summary:</label>
    <textarea name="summary"></textarea>
  </div>

  <div>
    <label for="summary">Released?</label>
    <select name="colour">
      <option value="yes">Yes</option>
      <option value="no">No</option>
    </select>
  </div>
  ...
</form>
```

## Create a Book

Book Title:

Summary:

Released?

Yes ▼

## Genres

Horror ☐

Fantasy ☐

Biography ☐

CREATE BOOK

# Forms – Example

```

...
<div>
  <h2>Genres</h2>
  <label for="genreHorror">
    Horror
    <input
      id="genreHorror"
      type="checkbox"
      name="genre[]"
      value="horror"
    >
  </label>
  <label for="genreFantasy">
    Fantasy
    <input
      id="genreFantasy"
      type="checkbox"
      name="genre[]"
      value="fantasy"
    >
  </label>
  <label for="genreBiography">
    Biography
    <input
      id="genreBiography"
      type="checkbox"
      name="genre[]"
      value="biography"
    >
  </label>
</div>

<button type="submit">Create Book</button>
</form>

```

## Create a Book

Book Title:

Summary:

Released?

## Genres

Horror ☐

Fantasy ☐

Biography ☐

CREATE BOOK

# Layout Elements

- Many elements exist with the purpose of building a page layout, or sectioning other elements
- Similarly to typography, there are both generic elements and contextual elements

# Layout Elements

- **<article/>** - wraps independent content, e.g. a newspaper article or forum post
- **<header/>** - wraps a header section, typically including an h1-h6 but may also include other elements like author or logo
- **<nav/>** - wraps a section which defines any kind of navigation
- **<section/>** - wraps a thematic group of content in a page, usually including a **<heading/>**
- **<footer/>** - wraps footer information for the nearest sectioning element
- **<div/>** - block-level generic wrapper used to group elements when there is no suitable specific element, usually for styling purposes
- **<span/>** - generic wrapper used to group inline elements, such as to wrap word(s) in text for styling purposes

# Layout Elements

```
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#">Link 1</a></li>
        <li><a href="#">Link 2</a></li>
      </ul>
    </nav>
    <h1>
      Joe Bloggs' Blog
    </h1>
  </header>

  <section>
    <h2>Posts</h2>
    <article>
      <header>
        <h3>Article Title</h3>
      </header>
      <p>content</p>
    </article>

    <aside>
      Author info
    </aside>
  </section>

  <footer>
    Copyright Info
  </footer>
</body>
```

- Link 1
- Link 2

Joe Bloggs' Blog

Posts

Article Title

content

Author info

Copyright Info

# Common Attributes

- **id** – a unique identifier for an element, must not appear in a single page more than once
- **class** – space separated list of class labels for an element, used mainly for selecting elements for styling purposes (see CSS)
- **tabindex** – set as an integer indicating the order in which elements can be navigated to when using the tab key. If set to -1, element cannot be tabbed to
- **data-\*** - any custom attributes which have no meaning to HTML, but hold data to be used in JavaScript
- **spellcheck** – indicates whether a given element should be checked for spelling errors
- **title** – produces a tooltip which shows on hover

# Code Style

- When writing HTML, like any code, we should be mindful of formatting
- Producing neat, consistently formatted code helps both your colleagues and your future self
- Some general rules with HTML;
  - Use lower-case tags and attribute names (`<p>` not `<P>`)
  - Generally one element per line
  - When nesting elements, indent the child (distance defined by the project's standard, typically 2 or 4 spaces)
- All blocks of whitespace regardless of size are collapsed to a single space (within `<pre>` is an exception)



# Code Style – Simple Example

```
<!-- Bad -->
<div><h1>My Heading</h1>
<h2>My Subheading</h2>

<p>This is a paragraph of long text</p>

<ul><li>List Item </li><li>List Item </li><li>List Item </li></ul>
</div>
```

```
<!-- Good -->
<div>
  <h1>My Heading</h1>
  <h2>My Subheading</h2>

  <p>This is a paragraph of long text</p>

  <ul>
    <li>List Item </li>
    <li>List Item </li>
    <li>List Item </li>
  </ul>
</div>
```

- How neat your code is creates a lasting impression, particularly when looking for work! Get into the habit early