HTML

V1.100923

Quick recap

How did you find your session yesterday?

Give me one thing you took from yesterday’s session

Today’s to-do list

Today we are going to look at **HTML**

We will look at how to code basic pages and how to use Semantic HTML to create more effective pages.

Learning objectives

By the end of this session you will be able to:

- Grasp the concepts of HTML
- Understand the role in structuring web content
- Get hands on experience of writing HTML

Show of hands

Show of hands, who has **written code**?

And who has **Used HTML**?

What have you used?

HTML History

Young Sheldon Season 6 set in 1993.

This is his mainframe, a DEC Vax- 6000. Access via the internet was limited to text-based programs.

HTML History

- 1980 - HyperText Markup Language was developed Tim Berners-Lee while at CERN
- 1989 - Proposed using for a large number of collaborative institutes around the world
- 1990 - prototype browser NeXT launched
- 1991 - Open discussions on HTML across the internet

HTML History

- 1993 - Lynx & Mosaic browsers released
- 1994 - First WWW conference held showing HTML+
- 1994 - HTML 2 released
- 1994 - Netscape released
- 1994 - The World Wide Web Consortium formed
- 1995 - HTML3

HTML History

- 1995 - Microsoft’s Internet Explorer released
- 1995 - Netscape proposes frames
- 1995 - Style sheets for HTML documents
- 1996 - W3 consortium starts work on javascript
- 1996 - Work started on ‘Cougar’ which became HTML 4
- 1997 - HTML 4 released

HTML 4 to HTML 5

dms4all
Digital Marketing Skills4All

- HTML4 had limitations supporting multimedia events, interactive features and couldn’t meet developers demands
- 2008-2012 - HTML 5 gradually adopted
- HTML5 provided a more robust and versatile set of elements and features
- 2016 - HTML 5 finally issued as a W3C recommendation

HTML5 Key advancements

- New Semantic Elements
- Multimedia Support
- Form Enhancements
- Canvas & SVG
- Local Storage API
- HTML5 development is ongoing with new features and improvements in every version

HTML Principals

- Markup Elements and Tags
  - Enclosed in angle brackets **<tagname>**
  - Provides instructions to the browser on how to display
- Hierarchy and Nesting
  - Elements can be nested inside other elements to create hierarchical structure
  - **<h1>** and **<p>** are nested inside the **<header>** element

HTML Principals

- Elements and Content
  - Elements are identified by their content
  - Can contain content, eg text, multimedia or elements
  - **<p>** contains text and **<img>** contains multimedia
- Attributes
  - Additional information provided within the the opening tag **<img src=pic.jpg>**

HTML Principals

- Semantics
  - Semantic elements provide context and meaning. They help both developers and search engines
  - **<header>, <nav>, <article>, <footer>, <section>**
- Accessibility
  - Creation of content that is inclusive
  - Features like Semantics and Alt Text and proper heading structure enhance accessibility

HTML Principals

- Clean, Well-Structured Content
  - Improved readability, maintainability, and allows for collaboration
  - Proper indentation, consistent naming conventions and comments
- Progressive Enhancement & Forward compatibility
  - Content designed to work on all browsers and devices
  - Content created will continue to function with new versions

HTML Element Structure

1. Opening Tag

- An HTML element begins with an opening tag, which is enclosed in angle brackets, < and >
- The opening tag indicates the type of element and provides instructions to the web browser on how to render it.
- Example: <p>

HTML Element Structure

2. Content

- The content of an HTML element can be text, other elements, or multimedia like images, audio, or video.
- Content is placed between the opening and closing tags and is what users see on the webpage.

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

HTML Element Structure

3. Closing Tag

- Most HTML elements have a closing tag, which is identical to the opening tag but includes a forward slash (/) before the element name.
- The closing tag signifies the end of the element's content.
- Example: </p>

HTML Element Structure

Putting it all together, here is how an element looks with its structure

```
<element> Content goes here </element>
```

Examples HTML Elements

1. Headings (<h1> to <h6>)
   - Provides 6 levels of headings with <h1> being the highest and most important

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
```

Examples HTML Elements

2. Paragraphs <p>
    - Used to group and display blocks of text

```html
<p>This is a paragraph of text.</p>
```

Examples HTML Elements

3. Lists <ul> <ol> <li>
   - Unordered lists
   - Ordered lists
   - List items

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Examples HTML Elements

4. Links <a>
   - Used to create hyperlinks to other web pages and resources

```
<a href="https://www.example.com">Visit Example</a>
```

Examples HTML Elements

5. Images <img>
   - Used to display images
   - Does not need a closing tag (Void Element)

```
<img src="image.jpg" alt="A beautiful image">
```

Examples HTML Elements

6. Divisions <div>
   - Container element for grouping style content
   - Often used with CSS for layout

```
<div>
  <p>Content inside a div element.</p>
</div>
```

HTML Tags

- Instructions that tell web how to display content. They are enclosed in angle brackets **<** and **>.**
- Tags come in pairs: an opening tag and a closing tag. Opening tag tells browser where an element begins, and the closing tag marks where it ends.
- Example of an opening tag: <p>
- Example of a closing tag: </p>
- The content you want to format or structure is placed between these tags.

HTML Tags

<p> for paragraphs of text

<h1>, <h2>, ... <h6> for headings

<ul> for unordered lists

<ol> for ordered lists

<li> for list items

<a> for hyperlinks

<img> for images

<div> for divisions (used for layout)

<span> for inline styling

HTML Attributes

- HTML tags can have attributes that provide additional information about the element.
- Attributes are placed within the opening tag and are usually written as name-value pairs, separated by an equal sign (=).
- Attributes modify the behaviour or appearance of the element.
- Example of an attribute:

```
<a href="https://www.example.com">
```

HTML Attributes

Attribute	
href	Specifies the URL for hyperlinks (<a>)
src	Specifies the source URL for images (<img>)
alt	Provides alternative text for images (for accessibility (and SEO)).
class	Assigns a class name to an element for styling with CSS.
id	Provides a unique identifier for an element (usually for JavaScript or CSS).
style	Allows inline CSS styling for an element.
title	Adds a title or tooltip to an element (e.g., for <a> and <img>).

Nesting of Elements

- Proper nesting ensures that elements are opened and closed in the correct order
- They create hierarchy

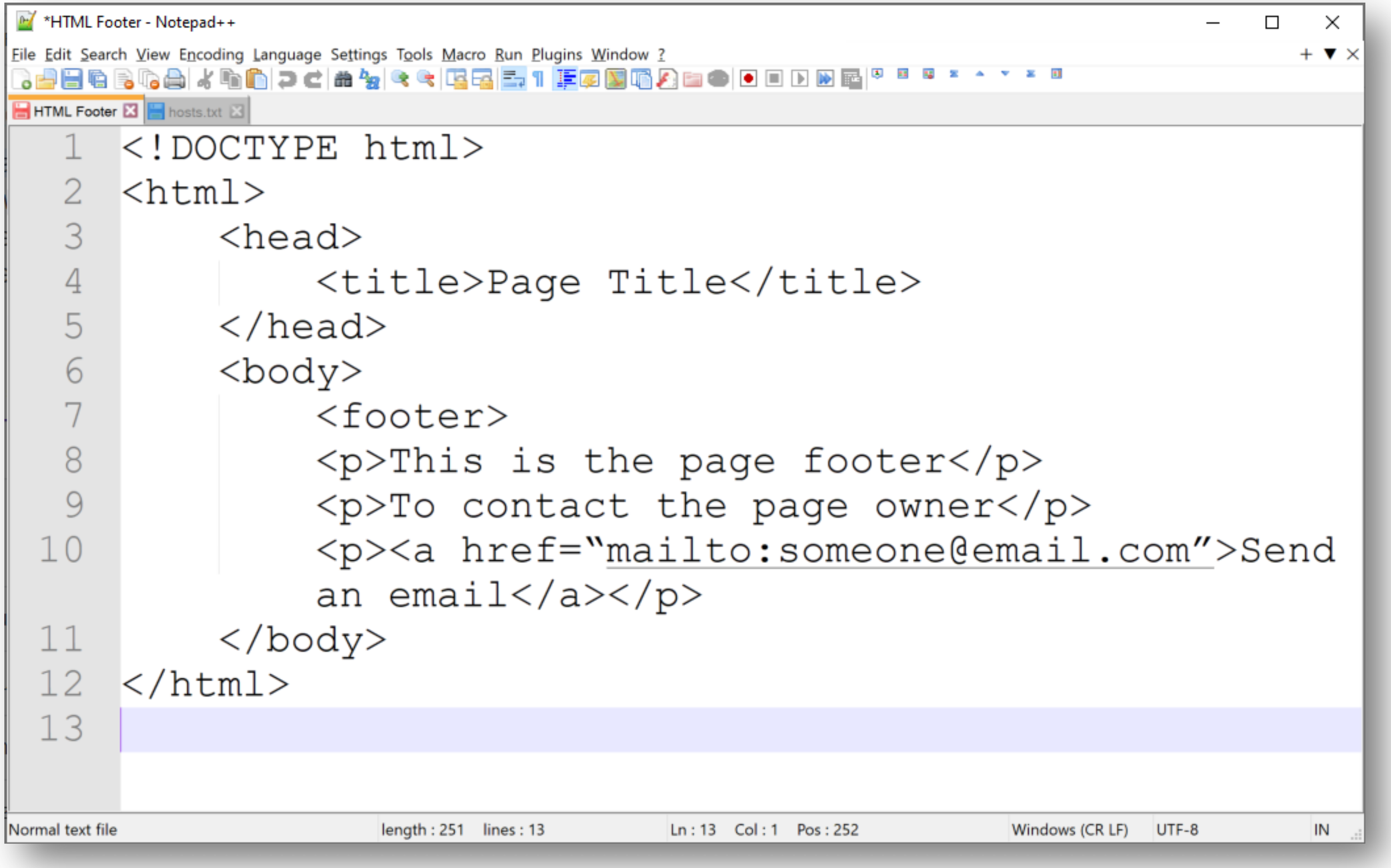
```
<div><p>This is a paragraph inside a division.</p></div>
```

Editors & Tools

1. HTML or text editor for writing and editing source code
   - Notepad++, Sublime Text, Visual Studio, WebStorm, Notepad
2. Browser for checking results
   - Chrome, Safari, Edge
3. Validator to check validity and syntax
   - validator.w3.org or W3schools.com

Notepad ++

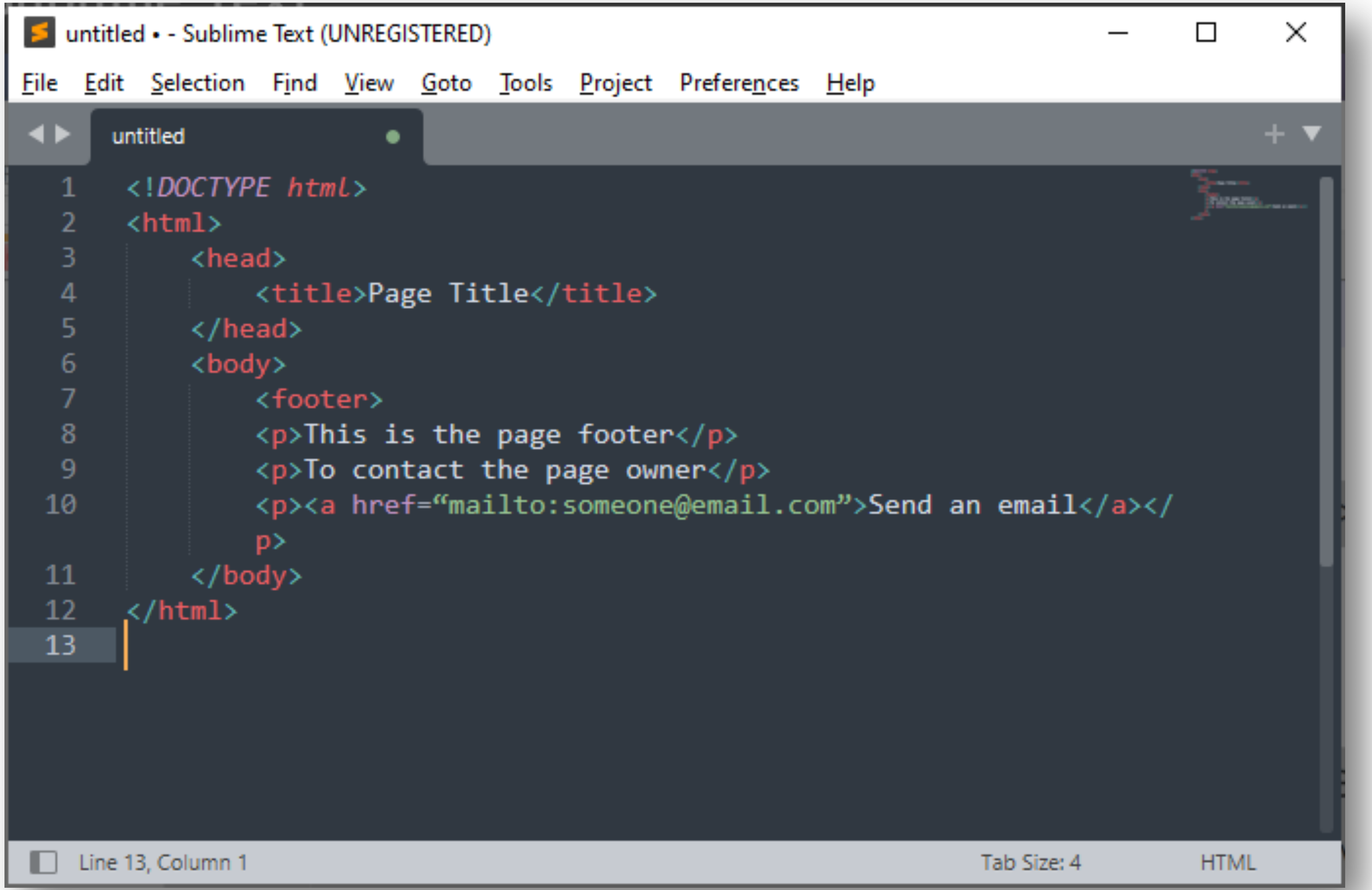
dms4all
Digital Marketing Skills4All


*HTML Footer - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
HTML Footer
hosts.txt

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <footer>
        <p>This is the page footer</p>
        <p>To contact the page owner</p>
        <p><a href="mailto:someone@email.com">Send
        an email</a></p>
    </body>
</html>
```

Normal text file
length : 251   lines : 13
Ln : 13   Col : 1   Pos : 252
Windows (CR LF)
UTF-8
IN

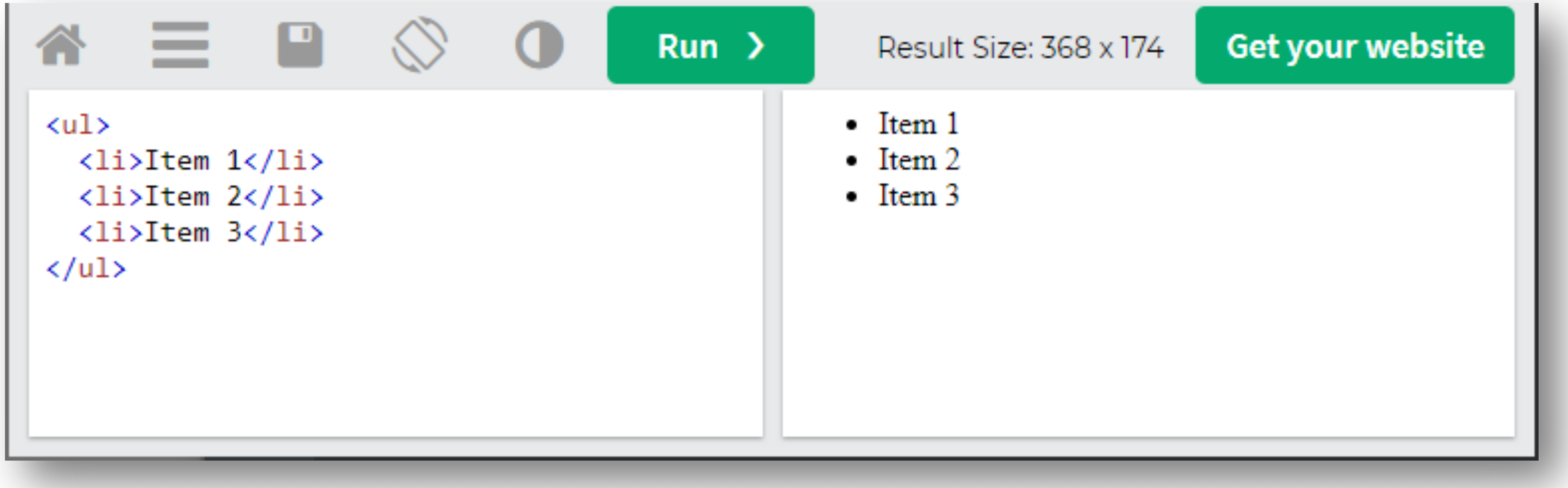
Sublime Text


untitled • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
untitled
Line 13, Column 1
Tab Size: 4
HTML

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <footer>
        <p>This is the page footer</p>
        <p>To contact the page owner</p>
        <p><a href=“mailto:someone@email.com”>Send an email</a></p>
    </body>
</html>
```

Validation tools

W3schools.com

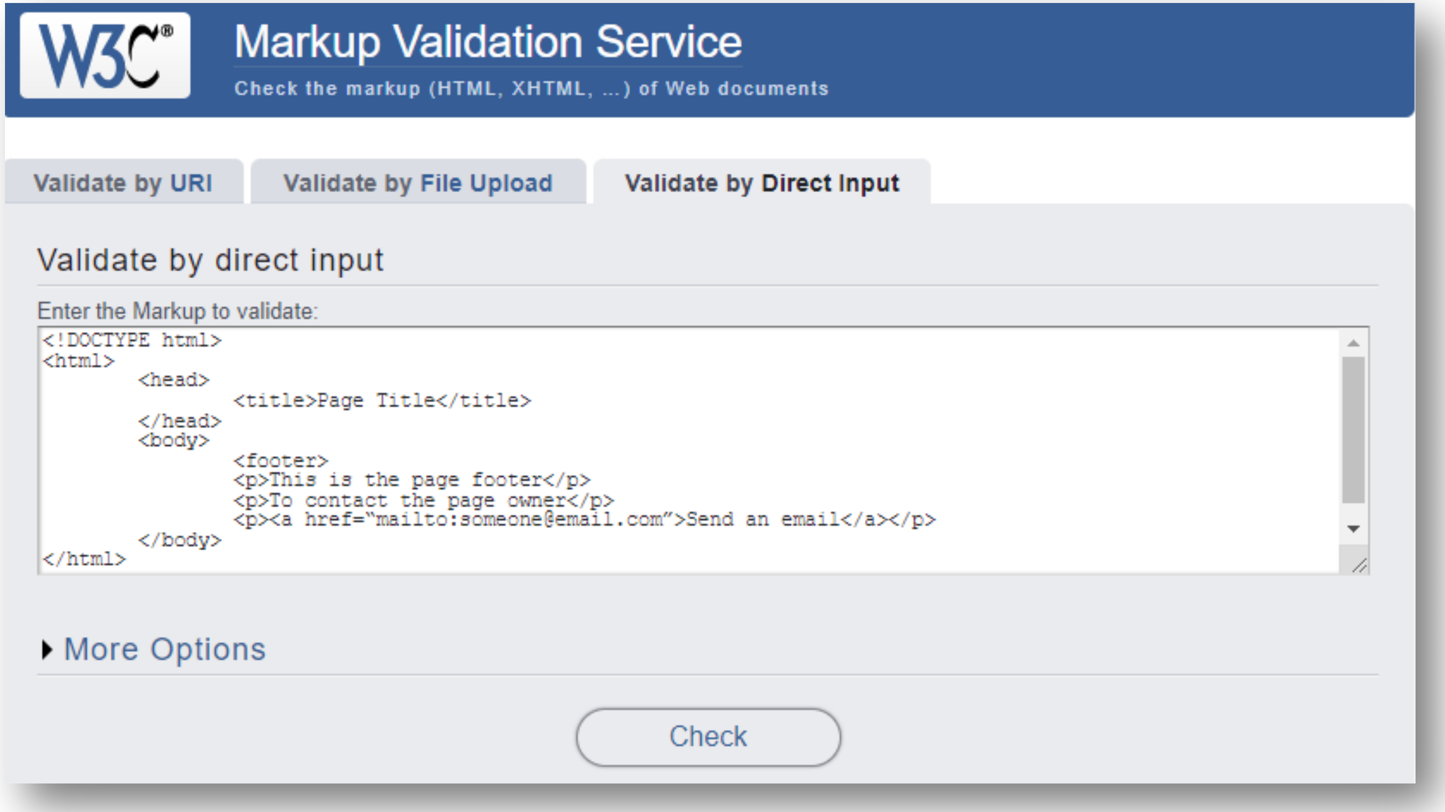


```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

- Item 1
- Item 2
- Item 3

Validation tools

w3.org



```
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
                <footer>
                <p>This is the page footer</p>
                <p>To contact the page owner</p>
                <p><a href=“mailto:someone@email.com”>Send an email</a></p>
        </body>
</html>
```

Workbooks out!

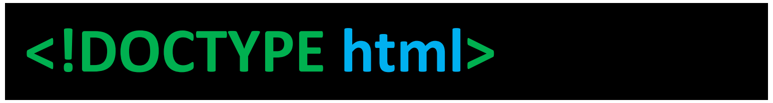
I would like you install one of the HTML Editors and pick a validation tool.

The links are in the workbook

Time for this activity: **15 mins**

Document Structure

1. <!DOCTYPE> Declaration:
   - The <!DOCTYPE> declaration is not an HTML tag but an instruction to the browser that specifies the type and version of HTML being used.
   - It should be placed at the very beginning of an HTML document, before the <html> element.
   - The <!DOCTYPE> declaration ensures that the browser interprets the document's content correctly.



```
<!DOCTYPE html>
```

Document Structure

2. <html> Element:
   - The <html> element is the root element of an HTML document. It encloses al the other HTML elements
   - It has two main parts: the <head> and the <body>.

```
<!DOCTYPE html>
<html>
  <!-- Content goes here -->
</html>
```

Document Structure

3. <head> Element:
   - The <html> element contains metadata and information about the document, which is not displayed on the page itself.
   - Typically includes elements like <meta> <title> <link>.

Document Structure

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>My Web Page</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <!-- Content goes here -->
</html>
```

Document Structure

4. <body> Element:
   - The <body> element contains the visible content of the webpage, including text, images, links, forms and other elements
   - Everything that you want users to see and interact with on the webpage should be on inside the <body>

Document Structure

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>My Web Page</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>Welcome to My Web Page</h1>
    <p>This is some content.</p>
    <!-- More content goes here -->
  </body>
</html>
```

Headings <h1> to <h6> Tags

- Size levels of heading with different levels of importance
- Used to define hierarchy of content on a webpage with <h1> being the most important

```
<h1>Main Heading</h1>
<h2>Subheading 1</h2>
<h3>Subheading 2</h3>
```

Paragraphs and Line Breaks

- Use <p> to create paragraphs of text.
- </p> isn’t a requirement, but it is good practice
- Use <br> within a paragraph to create a line break

```
<p>This is a paragraph of text. It can contain <br>
line breaks.</p>
```

Ordered and Unordered Lists

- There are two types of lists, Ordered and Unordered
- Ordered Lists <ol> are for lists that have a specific sequence or order – typically numbered.
- Unordered Lists <ul> have no order and are usually bulleted

Ordered and Unordered Lists

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

```
<ul>
  <li>Red</li>
  <li>Green</li>
  <li>Blue</li>
</ul>
```

Using List element <li>

- The List element, <li> is used within the <ol> or <ul> containers
- Each <li> represents an item on the list
- You can nest other HTML Elements inside <li> to create complex lists

Using List element <li>

```
<ul>
  <li>Item 1</li>
  <li>Item 2
    <ul>
      <li>Subitem 2.1</li>
      <li>Subitem 2.2</li>
    </ul>
  </li>
  <li>Item 3</li>
</ul>
```

Semantic Tags

- Semantic tags provide meaning to the content.
- These tags help both developers and search engines understand the structure of the webpage.

- <em>
  - The <em> element is used to *emphasize* text. It typically renders as italicized text.

Semantic Tags

- <strong>
  - The <strong> element represents **strong** importance or emphasis and is often displayed as bold text.
- <blockquote>
  - Use <blockquote> to indicate a block of quoted text. It can be used within articles or blog posts.
- <abbr>
  - The <abbr> element is used to mark up abbreviations or acronyms. You can specify the full expansion using the title attribute.

Semantic Tags

```
<p>This is <em>emphasized text</em> and <strong>strong text</strong>.</p>
<blockquote>
  <p>This is a quoted text from a famous author.</p>
</blockquote>
<p><abbr title="World Wide Web">WWW</abbr> is a common abbreviation.</p>
```

Inserting Hyperlinks <a> tag

- Hyperlinks, also known as anchor links, are created with the <a> element
- The href attribute within the <a> tag specifies the url of the document you are linking to
- You can create both internal and external links

Inserting Hyperlinks <a> tag

```
<a href="https://www.example.com">Visit Example</a>
```

```
<a href=«#section2">Jump to Section 2</a>
```

Workbooks out!

Let’s create a basic webpage. Use some headings, lists, and basic hyperlinks. Save As a .html

There is some code in the workbook to get you started.

Time for this activity: **30 mins**

Embedding Images <img> tag

dms4all
Digital Marketing Skills4All

- Use the <img> tag to embed images
- The src attribute within the <img> tag specifies the source URL or file path to the image

```
<img src="image.jpg" alt="A beautiful image">
```

Embed with <video> and <audio>

- HTML5 introduced native support for multimedia content, such as videos and audio.
- The <video> tag is used to embed videos, while the <audio> tag is used for audio files.
- You can specify multiple sources <source> within the tags to ensure compatibility with different browsers and file formats.

Embed with <video> and <audio>

```
<video controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  Your browser does not support the video tag.
</video>
```

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

Building with <form> tag

- Forms allow us to collect user inputs, such as text, selections and submissions
- The action specifies the URL to which the form data will be sent

```
<form action="submit.php" method="post">
  <!-- Form elements go here -->
</form>
```

Text and Password Input Fields

dms4all
Digital Marketing Skills4All

- Text input fields <input type="text"> are used for single-line text input, such as names and email addresses.
- Password input field <input type="password"> are used for securely entering passwords.

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

Checkboxes and Radio Buttons

- Checkboxes <input type="checkbox"> allow users to select multiple options.
- Radio buttons <input type="radio"> allow users to select a single option from a group.
- Use the name attribute to group related checkboxes or radio buttons.

Checkboxes and Radio Buttons

```html
<input type="checkbox" id="option1" name="options" value="Option 1">
<label for="option1">Option 1</label>
<input type="checkbox" id="option2" name="options" value="Option 2">
<label for="option2">Option 2</label>
<br>
<input type="radio" id="choice1" name="choices" value="Choice 1">
<label for="choice1">Choice 1</label>
<input type="radio" id="choice2" name="choices" value="Choice 2">
<label for="choice2">Choice 2</label>
```

Dropdowns <select> and <option>

- <select> is used to create a dropdown list of options
- <options> are defined within <select>

```
<label for="country">Select your country:</label>
<select id="country" name="country">
  <option value="usa">United States</option>
  <option value="canada">Canada</option>
  <option value="uk">United Kingdom</option>
</select>
```

Form Submission <button>

- The <button> element is used to create buttons within forms, including submit buttons.
- The type attribute specifies the button type. Use type="submit" for form submission.

```
<button type="submit">Submit</button>
```

Additional Form Elements

HTML offers various other form elements for specific purposes, including <textarea>, file upload fields <input type="file">, and hidden fields <input type="hidden">

Workbooks out!

Add a <form> to the page. Ask for first and last name, and add a submit button.

If you have time add a free text box, or try a drop down.

Time for this activity: **15 mins**

HTML5

- HTML5 is the latest version of the HTML standard, designed to provide improved functionality and semantics for web development.
- It includes new elements and attributes that make it easier to create structured and meaningful web content.
- HTML5 offers better support for multimedia, graphics, and interactive features.

New Semantic Elements

- HTML5 introduced new semantic elements to help provide meaning to the structure of the page
  - <header>: Represents introductory content at the beginning of a section or page.
  - <nav>: Represents a section containing navigation links.
  - <article>: Represents a self-contained composition within a document, such as a blog post or news article.
  - <section>: Represents a thematic grouping of content within a document.
  - <footer>: Represents the footer of a section or page.

HTML5 Input Types

- New input types that improve user interaction and data validation in web forms:
  - <input type="email">: Allows users to enter email addresses. It provides built-in email validation.
  - <input type="url">: Enables users to enter URLs (web addresses) and performs URL validation.
  - <input type="date">: Provides a date picker for selecting dates. The date format is typically based on the user's locale.

HTML5 Input Types

- <input type="number">: Allows users to input numeric values, with optional minimum and maximum values.
- There are other input types like tel, search, **color**, and more, each tailored for specific data types or input requirements.

```
<label for="email">Email:</label>
<input type="email" id="email" name="email">
<br>
<label for="birthdate">Birthdate:</label>
<input type="date" id="birthdate" name="birthdate">
```

Responsive Mode

- A webpages ability to adapt and provide optimal viewing on different screen sizes
- A lot of this is done in CSS
- Properly structured HTML is vital

Responsive Mode

1. Viewport Meta Tag
   - The <meta> tag with the viewport attribute is crucial for responsiveness. It informs the browser how to scale and size the webpage to fit different screen sizes.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

2. Fluid Layouts
   - Encourage the use of relative units like percentages (%) for specifying widths and heights of elements rather than fixed pixels.

```
<div style="width: 50%;">This div will take up 50% of its parent's width.</div>
```

Responsive Mode

3. Images and Media

   - Use the max-width: 100%; CSS rule for images and media elements like <img> to ensure they don't exceed the width of their container.

```
<img src="responsive-image.jpg" style="max-width: 100%;" alt="Responsive Image">
```

Responsive Mode

4. Responsive Tables
   - For tables, consider using the <table>, <thead>, <tbody>, and <tfoot> elements to structure your tables semantically.
   - You can also use the CSS overflow-x: auto; property to allow horizontal scrolling for wide tables on small screens.
5. Mobile-First Approach
   - Encourage a mobile-first approach by designing your HTML content with mobile devices in mind and then progressively enhancing it for larger screens using CSS media queries in a later course.

Responsive Mode

6. Media Elements

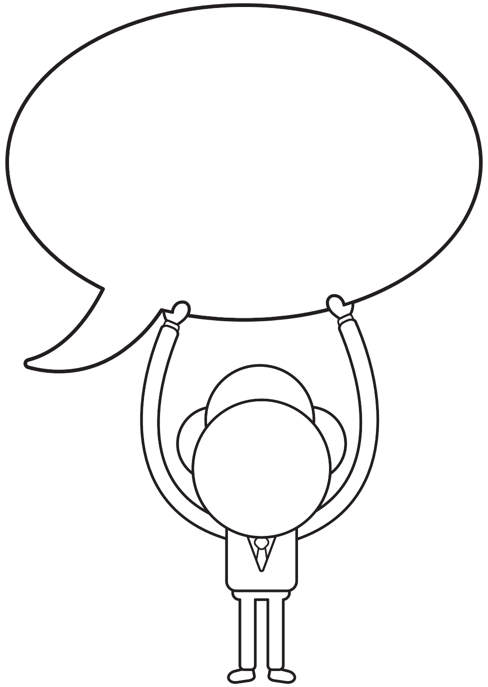
- For media elements like <video> and <audio>, you can specify the width attribute as a percentage to ensure they adapt to the screen width.

```html
<video src="responsive-video.mp4" width="100%" controls>
  Your browser does not support the video tag.
</video>
```

Responsive Mode

7. HTML5 Semantic Elements
   - Continue to emphasize the use of HTML5 semantic elements like <header>, <nav>, <main>, <article>, <section>, and <footer> as they provide a clear content structure that can be beneficial in responsive design.
8. Relative Font Sizes
   - Encourage the use of relative font sizes like em or rem instead of fixed px values for text elements. This ensures that text scales with the viewport.

Show and tell



Does anyone want to share their ideas with us?

Best Practice

Clean structured code

Consider responsive mode

Test, test, test

Any Questions?

Coming up soon to extend these skills

Tomorrow we will look at CSS which will enhance the design of your website. Wednesday we will look at *JavaScript*.

Quiz

Quiz books out

10 quick questions to test your knowledge

Final thoughts

Any comments on today?
What one thing do you feel you have taken from or improved upon from today?

Don’t forget

Make sure you submit your workbooks, once you do you will get a copy to your email.

That email will have the links to download todays slides and notes.

Extra help

dms4all
Digital Marketing Skills4All

If you need any help with anything from today’s session feel free to message me in the WhatsApp group.

If you have any other issues or problems related to the course or the tools speak to Andrew and Irfana,
they are here to help you.

Up next…

See you tomorrow for **CSS**