HTML Elements

Unit 1: The Web and HTML

1

Lesson Objectives

01

Introduction to HTML

What is the WWW?
What is HTML?

02

Creating HTML

HTML5 boilerplate

03

HTML tags

Exploring and understanding further HTML tags

2

Introduction to HTML

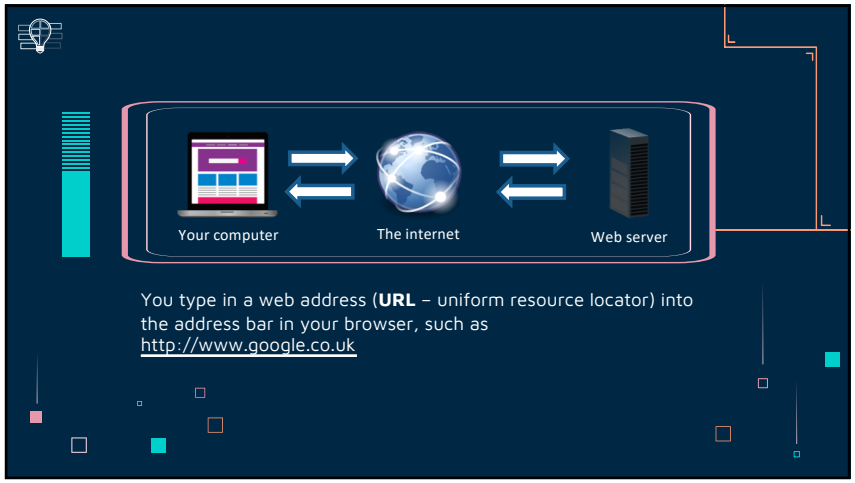
01

What is the WWW?
What is HTML?

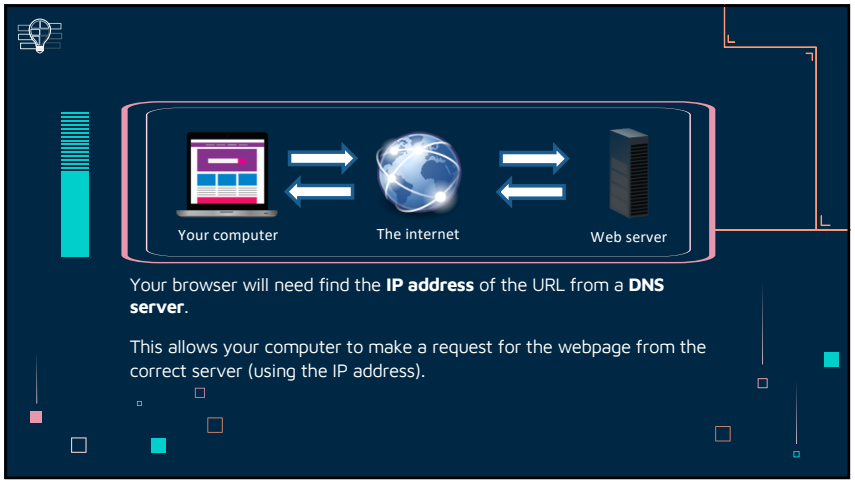
6

How do you think websites get delivered to our computers?

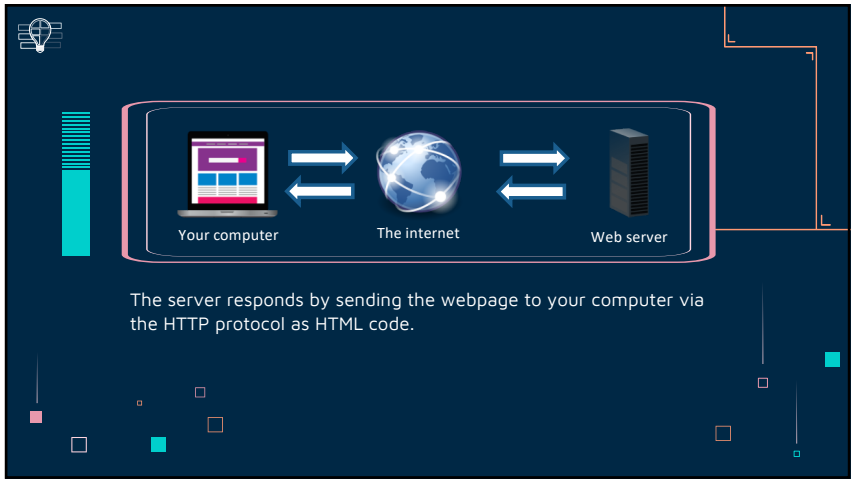
7


Your computer
The internet
Web server
You type in a web address (**URL** – uniform resource locator) into the address bar in your browser, such as http://www.google.co.uk

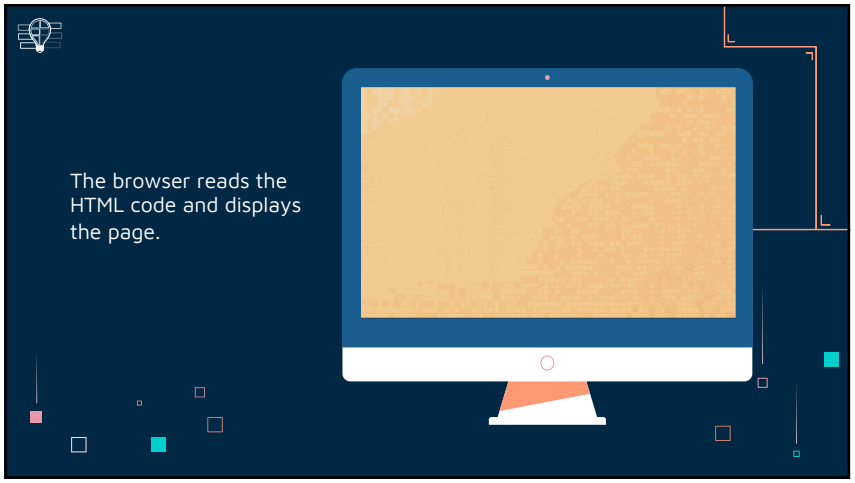
8


Your computer
The internet
Web server
Your browser will need find the **IP address** of the URL from a **DNS server**.
This allows your computer to make a request for the webpage from the correct server (using the IP address).

9


Your computer
The internet
Web server
The server responds by sending the webpage to your computer via the HTTP protocol as HTML code.

10


The browser reads the HTML code and displays the page.

11

What is HTML?

HTML stands for **HyperText Markup Language** and it is used for creating webpages.

A **web browser** is designed to read HTML and then translate it into the things you can see on the screen.

12

What is HTML?

HTML is a **markup language** as it defines structure.

HTML is not a programming language as it does not have features such as iteration (loops/repetition such as for and while) or branching (conditionals/decisions such as if, elif, and else).

13

Creating HTML

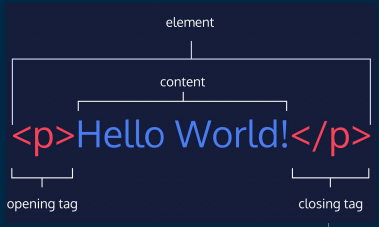
02

HTML5 Boilerplate

14

HTML Anatomy

HTML is composed of **elements**; they **structure** the webpage and **define** its content



15

HTML essentials

1. There are chevrons < > around tags.
2. The four essential HTML features are DOCTYPE, html, head, and body. This is what we call **boilerplate** code as every page has it.
3. Most tags have a closing tag with a forward slash.

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>

    </body>
</html>
```

16

HTML essentials

<!DOCTYPE html> is always the first line of an HTML file.

<html> tells the browser that we have used HTML to code the page.

<head> is meant for things that aren't displayed in the main body of your page, like the title.

<body> contains all the content you want to see displayed.

```
<!DOCTYPE html>
<html>
    <head>

    </head>
    <body>

    </body>
</html>
```

17

Browser

We can prove that all webpages have boilerplate by looking at the underlying code for one.

In a browser we can:

- press CTRL+U
- right-click on the page we are on and choose 'View Source' (IE) or 'View Page Source' (Chrome)

You should be able to find features from the boilerplate such as:

<!DOCTYPE>

<html>

<head>

<body>

18

HTML Headings

Headings in HTML are similar to headings in other types of media: newspapers – large headings to grab attention etc.

In HTML there are six different heading elements, ranging from <h1> (largest) to <h6> (smallest).

```
<body>
    <h1>Dolphins</h1>
    <h2>About Dolphins</h2>
    <h3>Species</h3>
    <h3>Features</h3>
```

Dolphins

About Dolphins

Species

Features

19

Common HTML elements

Main heading
In HTML this is given by
<h1>The Daily News</h1>

Subheading
In HTML this is given by
<h3>The Daily News</h3>
There are six different sizes of the h tag (h1 to h6)

Paragraphs (normal text)
In HTML this is given by
Some text

20

Task: Your first webpage

```
<body>
    <h1>My Favourite things</h1>
    <p>This web page is about my favourite things.</p>
    <h3>My favourites foods</h3>
    <p>Pizza</p>
    <p>Chocolate</p>
</body>
```

What do you think this code will display, and what will it look like?

Use the code editor to see if your prediction was correct.

21

Task: Your first webpage

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5      </head>
6      <body>
7
8      </body>
9  </html>
```

Create a folder in VS Code titled 'Week 2'. This is where you are going to save all of your work for this project.

Be careful to only work within this folder during the project otherwise your website may not work as intended.

22

Task: Your first webpage

```
<body>
    <h1>My Favourite things</h1>
    <p>This web page is about my favourite things.</p>
    <h3>My favourites foods</h3>
    <p>Pizza</p>
    <p>Chocolate</p>
</body>
```

After you have finished writing the code, open the file in your web browser.

Your code will be displayed.

23

HTML Tags

03

Exploring and understanding further HTML tags

24

HTML Text

Paragraphs <p> contain plain text.
Span is used to separate small sections of text on the same line.

Emphasis tag (Italics).
Strong tag – (Bold).

```
<h1>About Me</h1>
<p>My name is <strong>Max</strong> and I'm a Trainer at TTA</p>
```

About Me

My name is **Max** and I'm a Trainer at TTA

25

List tags

```
<ul>
    <li>Pizza</li>
    <li>Burgers</li>
</ul>
```

```
<ol>
    <li>Shrek</li>
    <li>Ghostbusters</li>
</ol>
```

Using HTML list tags we can create two different types of lists:

- **<ul>** stands for unordered list
- **<ol>** stands for an ordered list

What do you think is the difference?

26

List tags

```
<ul>
    <li>Pizza</li>
    <li>Burgers</li>
</ul>
```

```
<ol>
    <li>Shrek</li>
    <li>Ghostbusters</li>
</ol>
```

My favourite foods

- Pizza
- Burgers

My top 10 movies

1. Shrek
2. Ghostbusters

27

List tags

Can you create some lists for your page (top 10 musical artists, games, etc.)?

Challenge: Can you create a list with different levels like this?

- Item 1
  - Item a
  - Item b
- Item 2
- Item 3

```
<ul>
    <li>Pizza</li>
    <li>Burgers</li>
</ul>
```

My favourite foods

- Pizza
- Burgers

```
<ol>
    <li>Shrek</li>
    <li>Ghostbusters</li>
</ol>
```

My top 10 movies

1. Shrek
2. Ghostbusters

28

HTML Images

The **<img>** tag allows you to add an image to a web page. It is a self-closing tag.

The **<img>** tag has a required *attribute* called src. The src attribute must be set to the image's location.

```
<img src="image-location.jpg" />
```

29

HTML Videos

HTML also supports displaying videos. It also requires a src attribute – but is NOT self-closing.

The source can be a video file that is hosted on your web server, or a URL of a video hosted elsewhere.

Width and height attributes are used to set the size of the video. You can include the controls attribute to include your browser's basic video controls.

```
<video src="myVideo.mp4" width="320"
height="240" controls>
  Video not supported
</video>
```

30

HTML Divs

Paragraphs <p> contain plain text.
Span is used to separate small sections of text on the same line.

Emphasis tag (Italics).
Strong tag – (Bold).

```
<body>
  <div>
    <h1>Why use divs?</h1>
    <p>Great for grouping elements!</p>
  </div>
</body>
```

```
<p><strong>The Nile River</strong> is the
<em>longest</em> river in the world, measuring
over 6,850 kilometers long (approximately
4,260 miles).</p>
```

31

HTML Structure

HTML is organized as a collection of family tree relationships.

When an **element** is contained inside another element, it is considered the *child* of that element. The *child* element is said to be *nested* inside of the *parent* element.

```
<body>
  <p>This paragraph is a child of the body</p>
</body>
```

32

HTML Structure

In this example, the **<body>** element is the parent of the ** ** element. Both the **<h1>** and **<p>** elements are children of the element.

The **<h1>** and **<p>** elements are at the same level, so they are considered siblings and are both grandchildren of the **<body>** element.

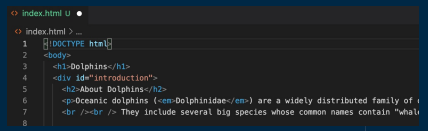
```
<body>
  <div>
    <h1>Sibling to p, but also grandchild of body</h1>
    <p>Sibling to h1, but also grandchild of body</p>
  </div>
</body>
```

33

Preparing for HTML

HTML files require certain elements to set up the document properly. We can let web browsers know that we are using HTML by starting our document with a *document type declaration*.

The <!DOCTYPE html> declaration provides the browser with two pieces of information (the type of document and the HTML version to expect), but it doesn't actually add any HTML structure or content.



34

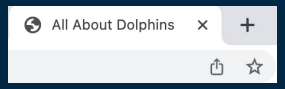
The Head

The <head> element contains the *metadata* for a web page. Metadata is information about the page that isn't displayed directly on the web page. Unlike the information inside of the <body> tag, the metadata in the head is information about the page itself.

35

Page Titles

A browser's tab displays the title specified in the <title> tag. The <title> tag is always inside of the <head>.



36

Linking to Other Web Pages

You can add links to a web page by adding an *anchor* element <a> and including the text of the link in between the opening and closing tags. The anchor element in the example above is incomplete without the href attribute. This attribute stands for *hyperlink reference* and is used to link to a *path*, or the address to where a file is located (whether it is on your computer or another location). The paths provided to the href attribute are often URLs.

```
<li>Bottlenose Dolphin</li>
<a href="https://en.wikipedia.org/wiki/Bottlenose_dolphin">More Information</a>
<li>Amazon River Dolphin</li>
<li>Ganges River Dolphin</li>
```

37

Opening Links in a New Window

The target **attribute** specifies how a link should open.
It's possible that one or more links on your web page link to an entirely different website. In that case, you may want users to read the linked website, but hope that they return to your web page. This is exactly when the target attribute is useful!
For a link to open in a new window, the target attribute requires a value of **_blank**.
The target attribute can be added directly to the opening tag of the anchor element, just like the **href** attribute.

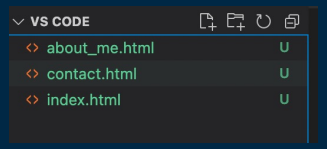
```
<li>Bottlenose Dolphin</li>
<a target=_blank href="https://en.wikipedia.org/wiki/Bottlenose_dol
<li>Amazon River Dolphin</li>
<li>Ganges River Dolphin</li>
```

38

Linking to Relative Page

Many sites also link to internal web pages like Home, About, and Contact.
When making multi-page static websites, web developers often store HTML files in the *root directory*, or a main folder where all the files for the project are stored. As the size of the projects you create grows, you may use additional folders within the main project folder to organize your code.

The example above shows three different files — **about_me.html**, **contact.html**, and **index.html** in one folder.
HTML files are often stored in the same folder, as shown in the example. If the browser is currently displaying **index.html**, it also knows that **about.html** and **contact.html** are in the same folder. Because the files are stored in the same folder, we can link web pages together using a *relative path*.
Contact



39

Linking to Relative Page

In this example, the <a> tag is used with a relative path to link from the current HTML file to the contact.html file in the same folder. On the web page, *'Contact me about Dolphins!'* will appear as a link. A **relative path** is a filename that shows the path to a *local file* (a file on the same website, such as ./index.html) versus an absolute path (a full URL, like https://www.bbc.co.uk/news/home-html which is stored in a different folder). The ./ in ./index.html tells the browser to look for the file in the current folder.

```
<head>
  <title>All About Dolphins</title></head>
<body>
<h1>Dolphins</h1>
<a href="./contact.html">Contact me about Dolphins!</a>
<div id="introduction">
  <h2>About Dolphins</h2>
```

40

Linking at Will

Thankfully, HTML allows you to turn nearly any element into a link by wrapping that element with an anchor element. With this technique, it's possible to turn images into links by simply wrapping the <img> element with an <a> element.

```
<a href="https://en.wikipedia.org/wiki/Bottlenose_dolphin" target="_blank">
<img src="https://pyxis.nymag.com/v1/imgs/260/250/46184df43c05ab8634927f8848667b5ca7-dolphin-b
width="240" height="320" alt="A Bottlenose Dolphin"/></a>
```

41

Linking to Same Page

When users visit our site, we want them to be able to click a link and have the page automatically scroll to a specific section. An id should be descriptive to make it easier to remember the purpose of a link. The target link is a string containing the # character and the target element's id.

```
<ul>
  <li><a href="#introduction">Introduction</a></li>
  <li><a href="#habitat">Habitat</a></li>
  <li><a href="#media">Media</a></li>
</ul>
```

42

Whitespace and Indentation

As the code in an HTML file grows, it becomes increasingly difficult to keep track of how elements are related. Programmers use two tools to visualize the relationship between elements: *whitespace* and *indentation*.

Both tools take advantage of the fact that the position of elements in a browser is independent of the amount of whitespace or indentation in the **index.html** file.

```
<body><p>Paragraph 1</p><p>Paragraph 2</p></body>
```

```
<body>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</body>
```

43

Whitespace and Indentation

The second tool web developers use to make the structure of code easier to read is *indentation*. The spaces are inserted using the space and tab bars on your keyboard.

The World Wide Web Consortium, or W3C, is responsible for maintaining the style standards of HTML. At the time of writing, the W3C recommends 2 spaces of indentation when writing HTML code. Although your code will work without exactly two spaces, this standard is followed by the majority of professional web developers. Indentation is used to easily visualize which elements are nested within other elements.

```
<div id="habitat">
  <h2>Habitat</h2>
  <h3>Countries with Large Dolphin
  <ol>
    <li>Australia</li>
    <li>Oman</li>
    <li>Sri Lanka</li>
  </ol>
</div>
```

44

Comments

HTML files also allow you to add comments to your code.

Comments begin with <!-- and end with -->. Any characters in between will be ignored by your browser.

Including comments in your code is helpful for many reasons:
They help you (and others) understand your code if you decide to come back and review it at a much later date.
They allow you to experiment with new code, without having to delete old code.

```
<li>Bottlenose Dolphin</li>
<a target=_blank href="https://en.wikipedia.org/wiki/Bottlenos
<li>Amazon River Dolphin</li>
<!-- Include another species?-->
<li>Ganges River Dolphin</li>
```

45