

HTML



CI435 Introduction to Web Development

Lecture 3

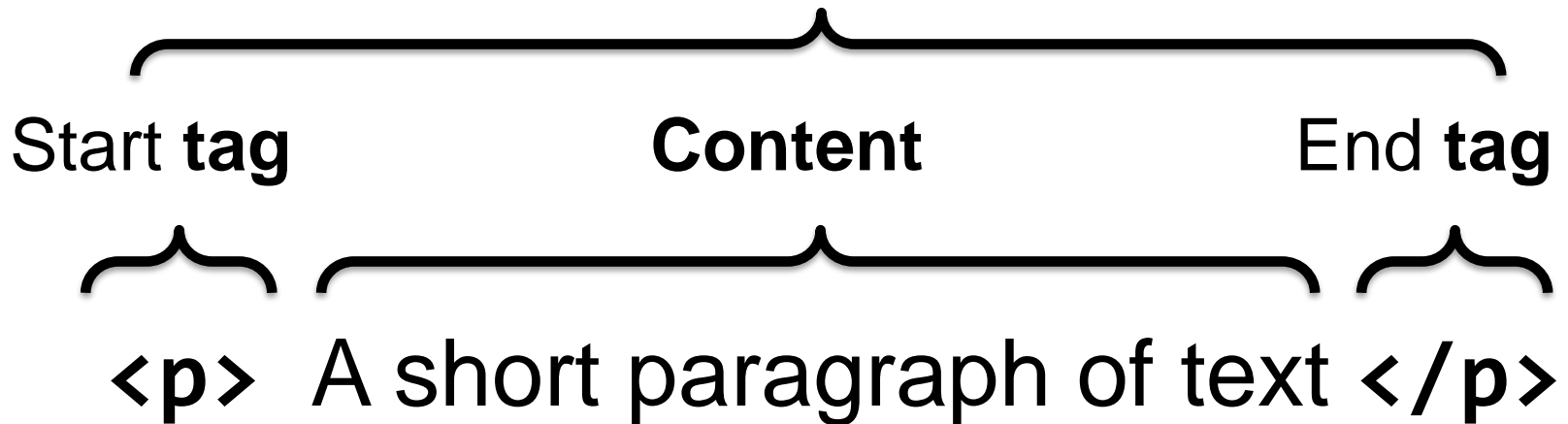
HTML5 document structure and hypertext

This lecture will cover...

- HTML5 document structure -
 - Section elements
 - The `<div>` element
 - *Learning Journal* document structure
- Hypertext: navigating and organising content
 - Anchors `<a>` – absolute and relative links
 - File management
- The `viewport` meta element

Reminder – the terminology

Element



HTML5 - document structure

- We've started you off with an HTML template - **index.html** – for your *Learning Journal*, which is available in lab tutorial 1
- This starts with the HTML5 doctype declaration –
<!doctype HTML>
- Between the <body> tags there is some basic HTML5 markup to structure the *Journal* content, using HTML5 section elements - <header>, <nav>, <main>, <article>, <aside>, <footer>

HTML5 section elements

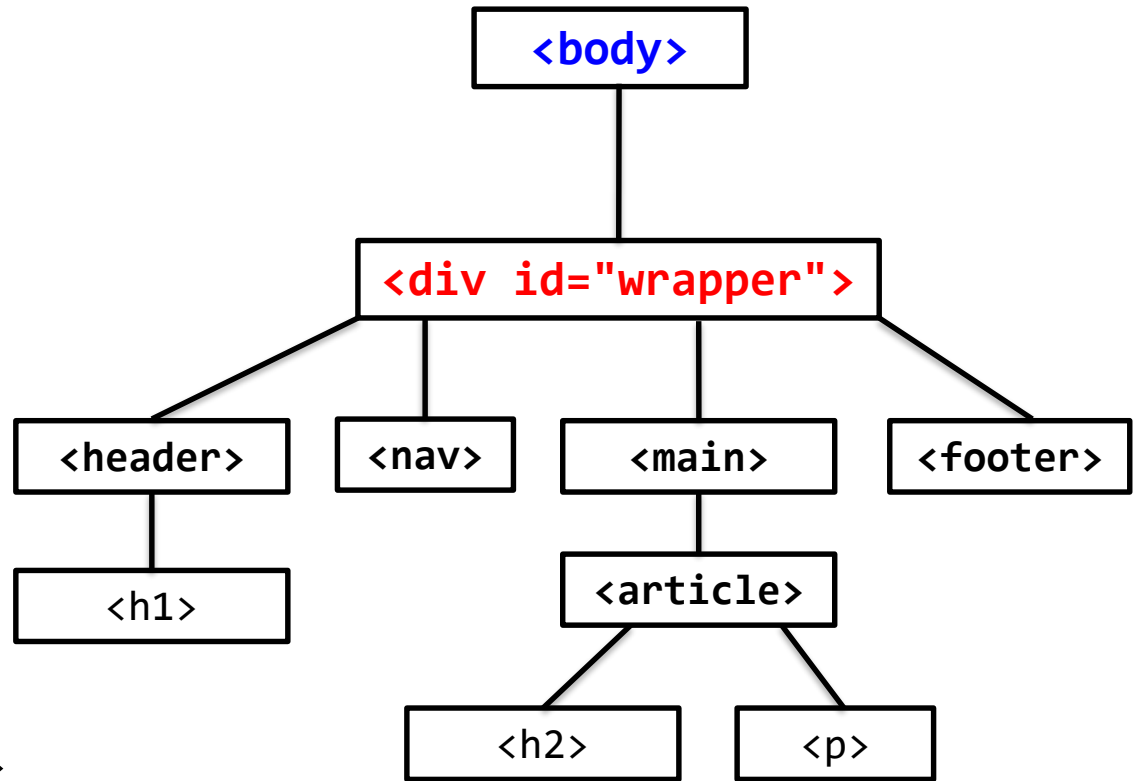
- Pre-HTML5 – document sections were marked up with **<div>** (division) tags
- Developers adopted similar names for web document sections -
 - **<div class="nav"></div>** - navigation
 - **<div id="header"></div>** - heading at top of page
 - **<div id="footer"></div>** - content usually at foot of page
- **class attribute** – used to identify elements of the same class; one or more occurrences in a web document allowed
- **id attribute** – used to uniquely identify an element; only **one** occurrence in the web document allowed
- My view - simpler to just use **class** all the time

HTML5 section elements

- In HTML5 the common attribute names have been used for new section elements - **<nav>**, **<header>**, **<footer>**, **<main>**
- HTML5 reflects the web as it is today: semantically meaningful section tags replace **<div>** tags
- **<div>** can still be used to markup content groups for styling purposes – for example, layout or image galleries
- HTML5 semantic section elements should be used *wherever appropriate* to structure the web page and content
- HTML5 elements have much richer semantics for the structure of a document; allows browsers (particularly **screen readers**) to navigate to content sections of a web page

index.html document object model*

```
<body>
  <div id="wrapper">
    <header>
      <h1></h1>
    </header>
    <nav> </nav>
    <main>
      <article>
        <h2></h2>
        <p></p>
      </article>
    </main>
    <footer> </footer>
  </div>
</body>
```



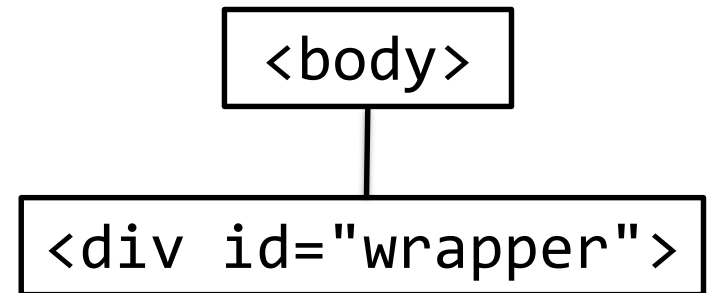
*you'll come back to DOM with JavaScript in Semester Two

div elements – where to use them

- **<div>** tags - *i.e. division* - are still used to mark up grouped content in a document in order to style these elements with CSS properties
- *E.g.* a **<div>** with the **id "wrapper"** is used to nest all the elements – this can be styled with CSS to centre them in the viewport with a margin each side
e.g. <https://www.brighton.ac.uk/index.aspx>

<body>

<div id="wrapper">



HTML5 section element semantics

<header> - header of a web document or content in a document; used to group heading content – *e.g.* **<h1>**, **<h2>**, **<time>** (and a banner or logo ****)

<nav> - navigation for a document – contains links to other sections of the current document, or to other documents

<main> - contains the main content of the page, use only once per page

<aside> - section of a page with content that is separate from the main content (often presented in a sidebar)

HTML5 section element semantics

<article> - a self-contained chunk of content – *e.g.* a weekly post in the learning journal, a comment on a blog post, a news item in a magazine style page

<footer> - content at the foot of a document – *e.g.* author details and date, *OR* at the foot of a section of the document – *e.g.* author and date of an **<article>**

See the **MDN elements reference** for definitions and examples of how to use these elements

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

HTML5 section element semantics

- Elements such as **<header>** and **<footer>** may be used several times on a page
- *E.g.* the main header and footer at the beginning and end of the page...
- ...and the header and footer to an **<article>** - *i.e.* journal post
- To be able to target them with CSS rules and style them differently they can be given **id** or **class attributes** -
<header id="banner"> - unique **id** because it only occurs once on the top of the web page
<header class="post"> - **class** attributes are used for elements that occur more than once on the page, *e.g.* the headers for many articles

HTML5 section element semantics

- Let's look at how the section elements have been used in the example *Learning Journal* document for **Week 2**
- I've embedded CSS in the <head> of the document for demonstration purposes only – to show the section elements
- *Please* don't do this in your own Learning Journal – CSS should only be in a separate stylesheet

http://jh1033.brighton.domains/ci435/tutorials/lj-index_2.html

Navigating content: anchors and links

- **HTML** documents are **hypertext** – navigate from one chunk of content to another through **links**...
- ...the essential difference between the web and everything that had gone before *i.e.* print
- The experience of reading hypertext is different from reading a book – this has changed our learning and information seeking behaviour
- Browsing – skipping from topic to topic following a trail of links – obtaining information in small chunks – discovery - connectivity *e.g.* Wikipedia

Navigating content: anchors / links

- Hypertext navigation is achieved through **anchors** – using the `<a>` tag to 'anchor' together two separate but related pages or pieces of information
- Link anchors point the browser to a **destination** –
 - Another section of the same page – **internal link**
 - Another page on the site – **local link**
 - Another website – **external link**
 - An email address – **email link**
- By **lab tutorial 3** you will have made all these types of link and learned how to mark them up

Navigating content: anchors and links

```
<a href="index.html">Home page</a>
```

```
<a href="https://developer.mozilla.org/en-US/docs/Learn/HTML">MDN</a>
```

- **<a>** tells the browser to link to another resource
- **href attribute** [hypertext reference] tells the browser the destination of the link
- The link label (text) tells the user where they will go when they click the link – *very* important to make this descriptive and clear to help users navigate

```
</a href="">MDN – learn web development</a>
```


Absolute and relative links

- External links **always** have the full URL of the destination = an **absolute link**
- Could also use an **absolute link** to link to pages on the same site

```
<a href="http://jh1033.brighton.domains/ci435/index.html">Home page</a>
```

- But it's more usual to use a **relative link** –

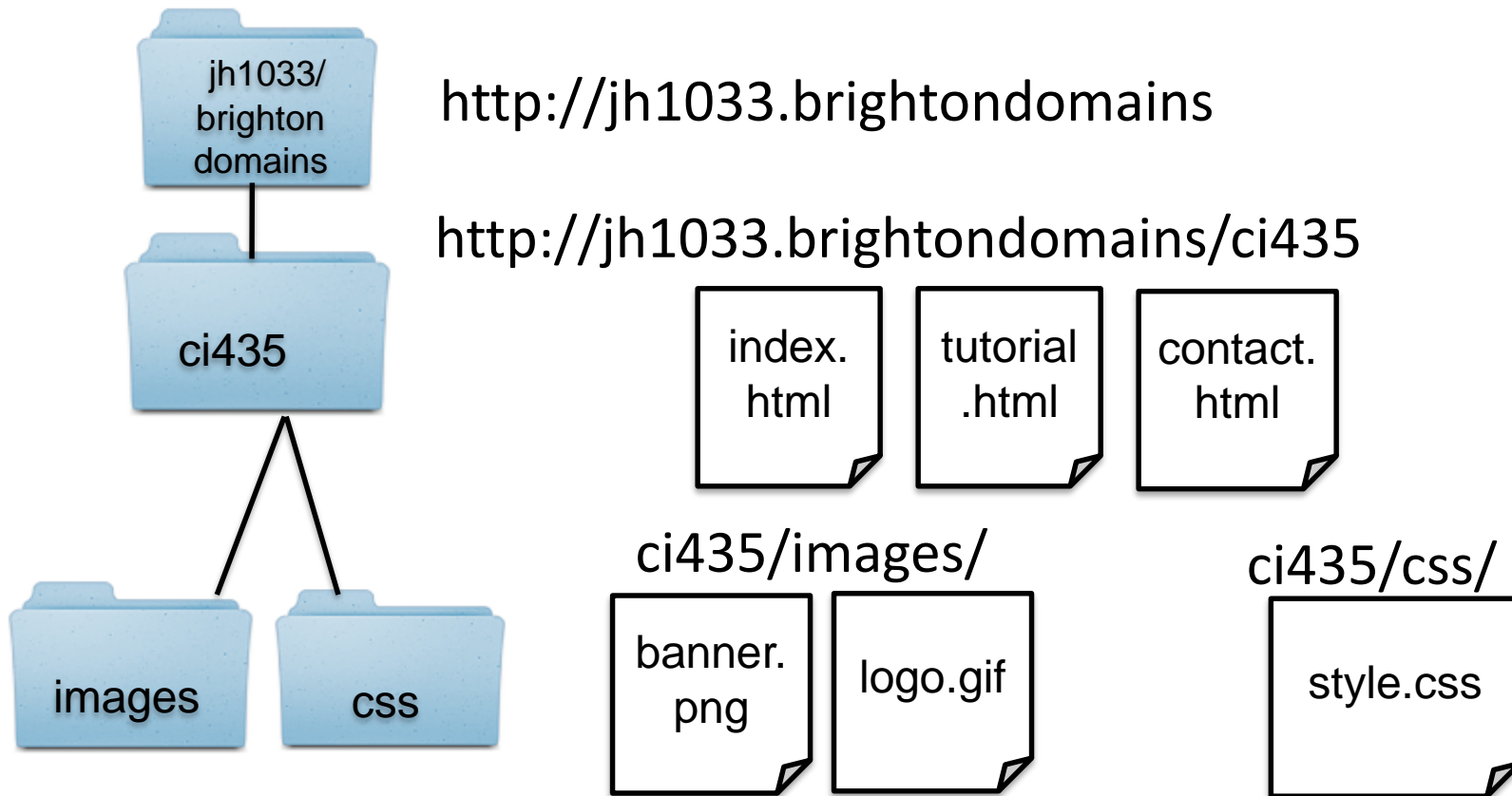
```
<a href="index.html">Home page</a>
```

WHY???

- Relative links reference the destination address relative to the page the anchor link is on

Absolute and relative links

- Well managed site folders and files are essential to maintaining the relative links between pages



Absolute and relative links

- To create a relative link between files in the **same** folder
`Guide to...`



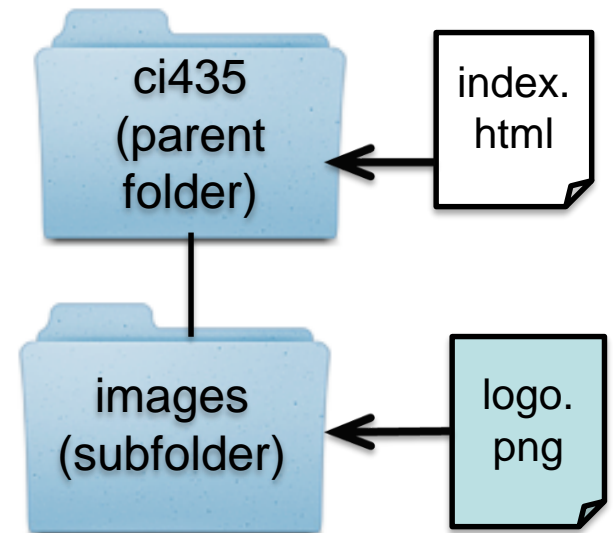
in index.html

- To create a relative link from a file (index.html) in a parent folder **down** to a file in a child folder –

in index.html

```

```



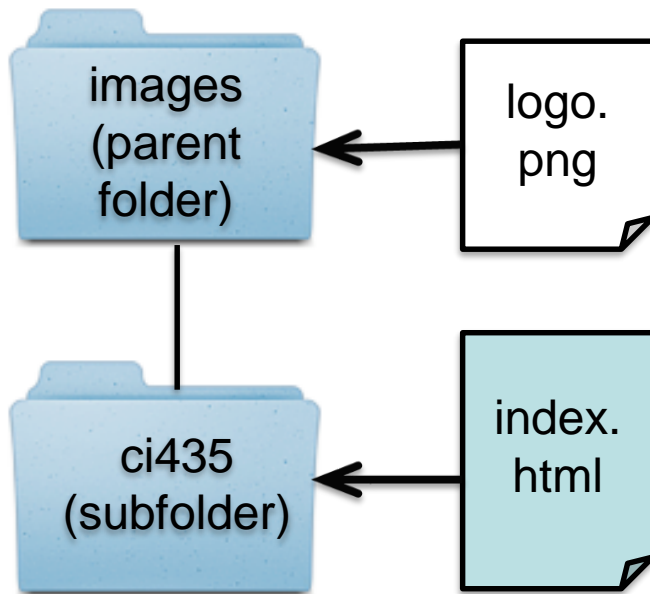
This is a common layout – html files in parent, with subfolders for other assets.

Remember, look at your file extensions too – part of the name. Also case sensitive on a Linux system.

Absolute and relative links

- To create a relative link from a file in a child folder **up** to a file in a parent folder –

`` *in index.html*

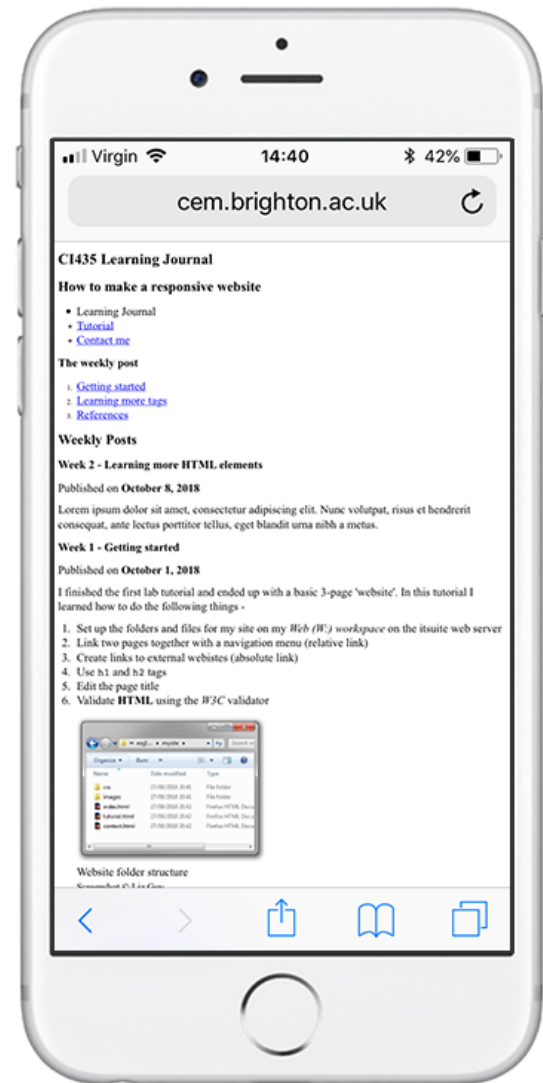


- `../` - means step up one level in the folder structure
- If you rename files or folders, create new folders, or move files you will have to update the relative links between files**

This folder pattern is often used when many sub areas of a website share common assets across a large organization e.g. logos etc.

Viewing on a smartphone browser

- Have you tested your Learning Journal on a smartphone?
- The template you are using has not been modified to make it display well on a smartphone browser
- Displays as a miniaturised page with no scroll bars
- Too small to read or interact with links unless you 'pinch and zoom' the page



The viewport meta element

- On desktop/laptop browsers the **viewport** = width of the browser window (minus chrome, scrollbars, menu *etc.*)
 - If the viewport/browser window is made smaller the web page remains the same size; vertical and horizontal scroll bars are added by the browser
- On mobile devices web pages are scaled down to fit the viewport/screen width and users have to 'pinch and zoom'
 - Very poor user experience
- We have to make a mobile browser behave like a desktop browser



The viewport meta element

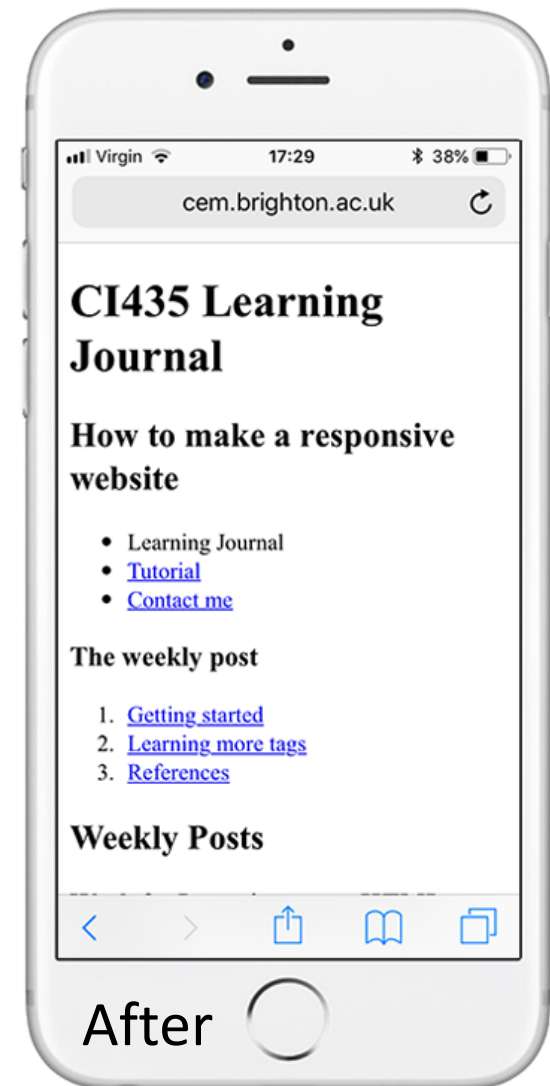
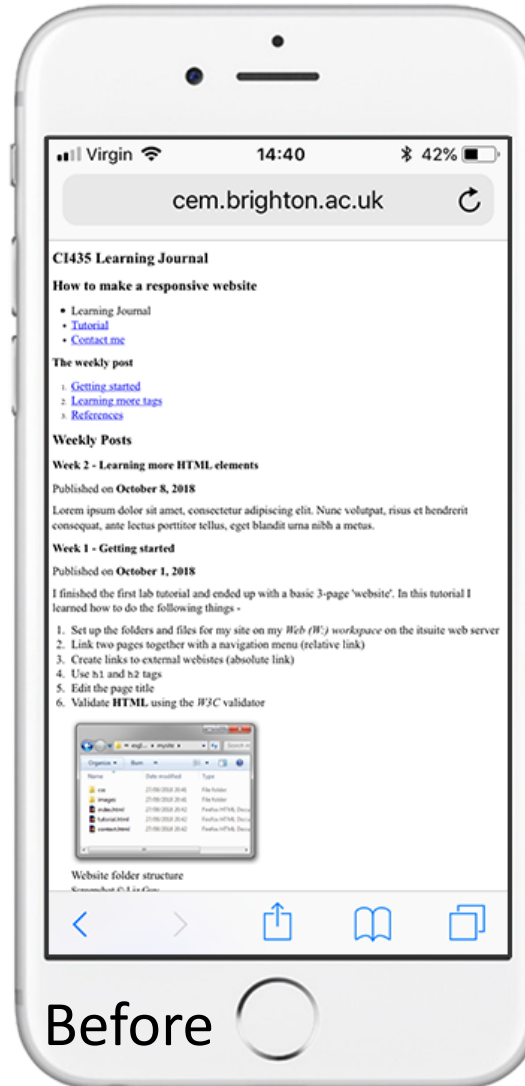
- The `viewport` attribute for the `meta` element allows the mobile device viewport to be set to a specific width
- Write this line in the `<head>` of your HTML documents after the `<title>` element-

```
<meta name="viewport" content="width=device-width, initial-scale=1"/>
```

- `"width=device-width"` - sets the pixel width of the viewport to be equal to the screen width of the device – so that a mobile browser will behave the same as a desktop one
- `"initial-scale=1"` – tells the browser to render the web page at full-size when it's loaded, rather than scaling it down

Viewing on a smartphone browser

- When you have added the viewport meta element test your Learning Journal page again on your smartphone
- **This is the first step to making a responsive web page**



This week's independent learning

- **MDN** - document and website structure
[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction to HTML/Document and website structure](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure)
- **MDN** - creating hyperlinks
[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction to HTML/Creating hyperlinks](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks)
- HTML5 video tutorials on the [LinkedIn Learning](#) Playlist
- Jon Duckett, *HTML & CSS: design and build websites*. Read Chapter 4 (Links)