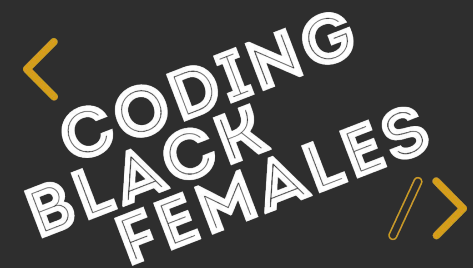


BLACK CODHER

CODING PROGRAMME

Black Codher Bootcamp

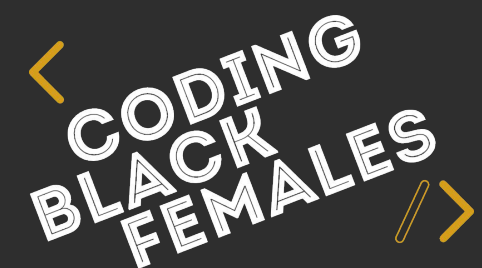


BLACK CODHER

CODING PROGRAMME

UNIT 2

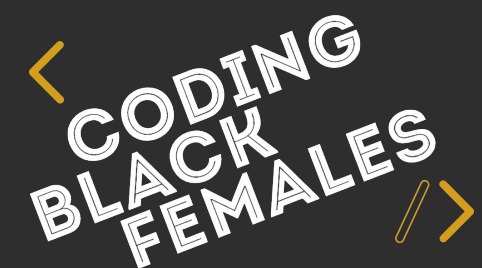
HTML & CSS



BLACK CODHER

CODING PROGRAMME

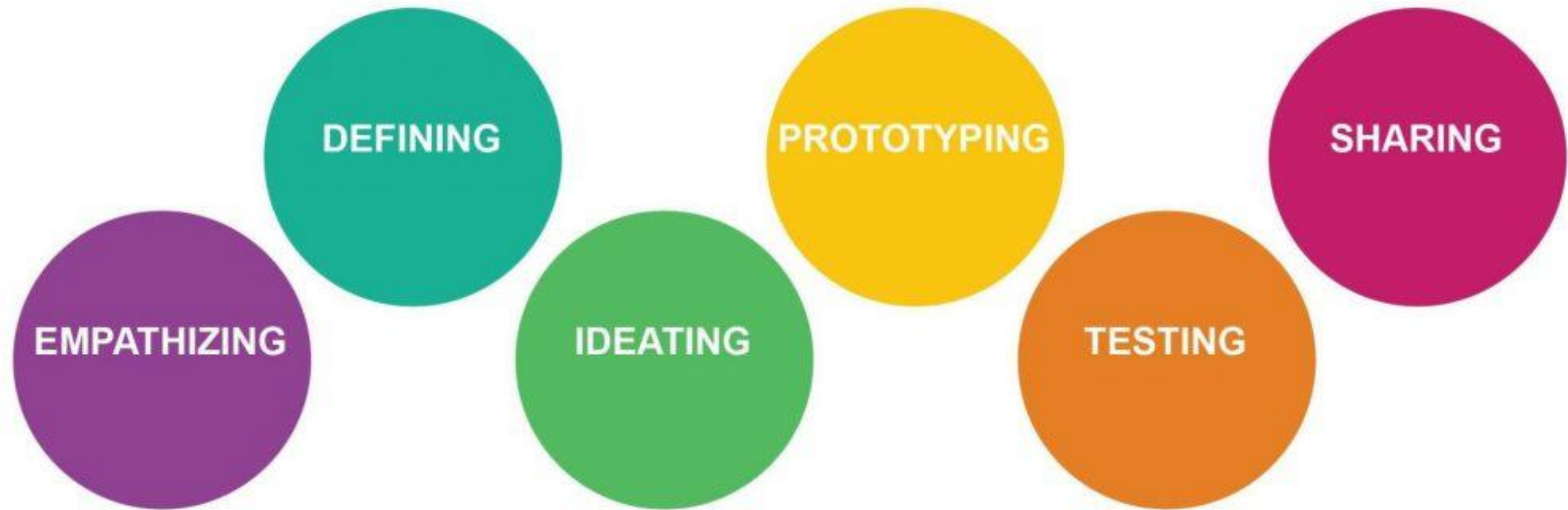
SESSION 2
Preview Time



Learning objectives

1. Understand when to use various formatting tags
2. Understand how to make ordered and unordered lists
3. Understand where different types of quotations might fit in





Design Thinking

Formatting tags - semantic/logical

**** tags indicate **importance** which normally displays as **bold** text.

**** tags indicate *emphasis* which normally displays as in *italics*.

You use these when you want to convey meaning by stressing a specific thought or idea as more significant.

A screen reader usually uses verbal stress, pronouncing the word(s) slightly more slowly or loudly than the surrounding ones.

As before, you're welcome to code as you go if you want to follow along with this like that - checking things in your browser to test.

```
<!-- When you go through hardships  
and decide not to surrender... -->
```

That is **strength**.

```
<!--Janelle Monáe, re: her beautiful  
visual album Dirty Computer...-->
```

This is an *emotion* picture.

Formatting tags - display/physical

``tags make text **bold**. As you know, they help make something stand out.

`<i>` tags make text *italic*. They are often used to indicate a technical term, a word from different language, a name etc.

Here are some other examples:

`<mark>`ed e.g. **highlighted**

``eted i.e. ~~strikethrough~~

`<u>`nderlined* (deprecated)

`<sup>`erscript ^{up top}

`<sub>`script _{down low}

`She is bold.`

`<i>It is italic.</i>`

Lists - (un)ordered

An unordered list is made with **** tags. Each list item uses **** tags. The list items are in bullets points by default.

An ordered list is created via **** tags. Each item is also wrapped with **** tags. Its list items are then shown as numbered, as you might reasonably expect.

- Anne Marie Imafidon
- Betelhem Dessie
- Chi Onwurah

1. Katherine Johnson
2. Mary Jackson
3. Dorothy Vaughan

```
<ul>
  <li>Anne-Marie Imafidon</li>
  <li>Betelhem Dessie</li>
  <li>Chi Onwurah</li>
</ul>

<ol>
  <li>Katherine Johnson</li>
  <li>Mary Jackson</li>
  <li>Dorothy Vaughan</li>
</ol>
```


Blockquotes & quotations

The **<blockquote>** element defines a section that is cited from another source.

Browsers tend to indent them.

The *cite* attribute allows us the option of citing a source in the form of a URL.

It tends to be for longer pieces of text.

<q> tags are for a short quotation.

Browsers then insert quotation marks “around the quotation.”

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<p>Here is a quote from client testimonial:</p>
<blockquote cite="https://facebook.com/mybiz/review1">
<p>I was really happy with the work that she did! I
would highly recommend her services.</p></blockquote>

<p>My goal is to:<q>Build a website that you love which
helps grow your business.</q></p>
</body>
</html>
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said



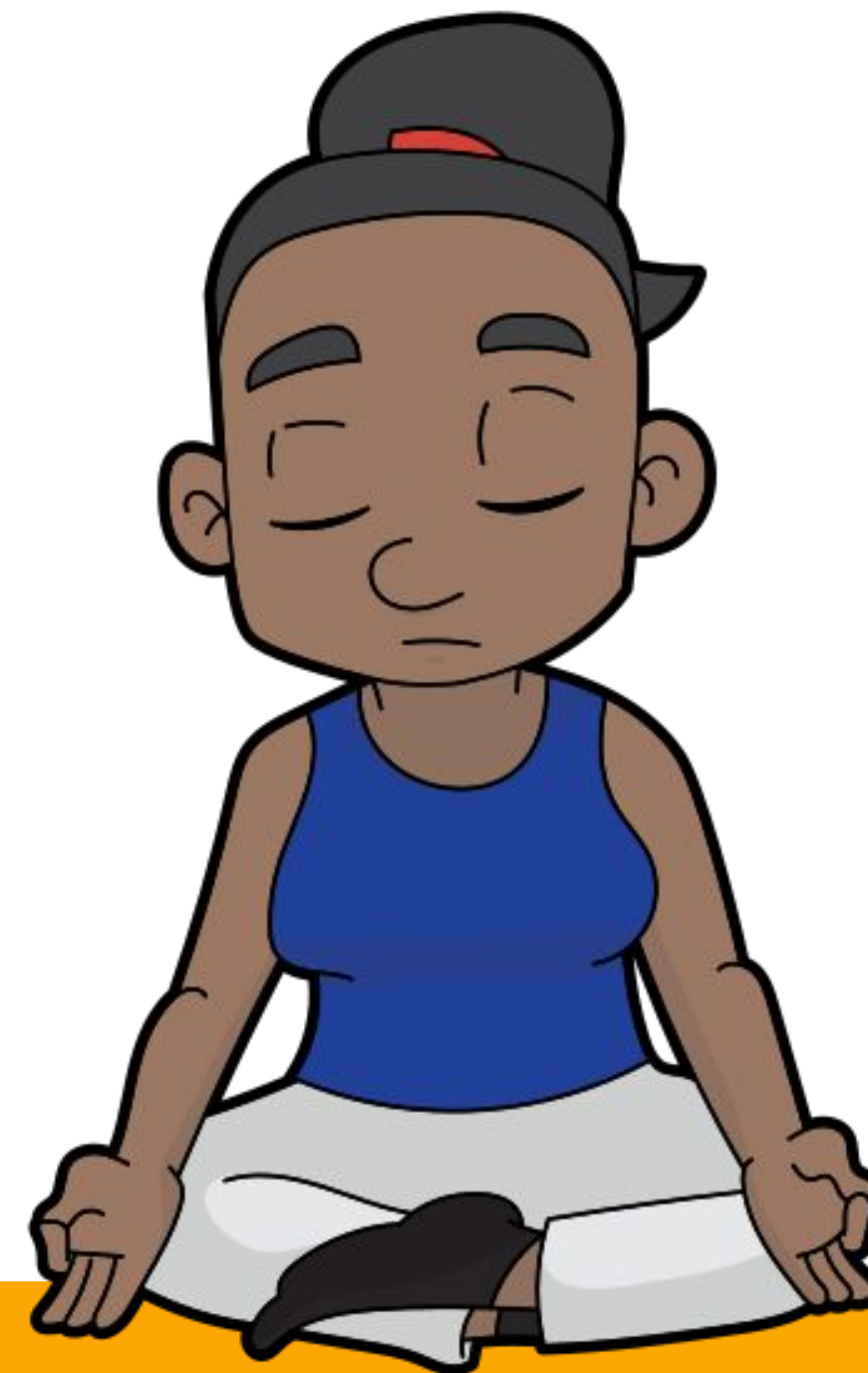
Let's discuss

Which of these are semantic - **** or **** and ** or *<i>* ?

How do we make lists and how do the unordered ones look *verzuz* ordered?

What are the 2 different ways of adding quotes?



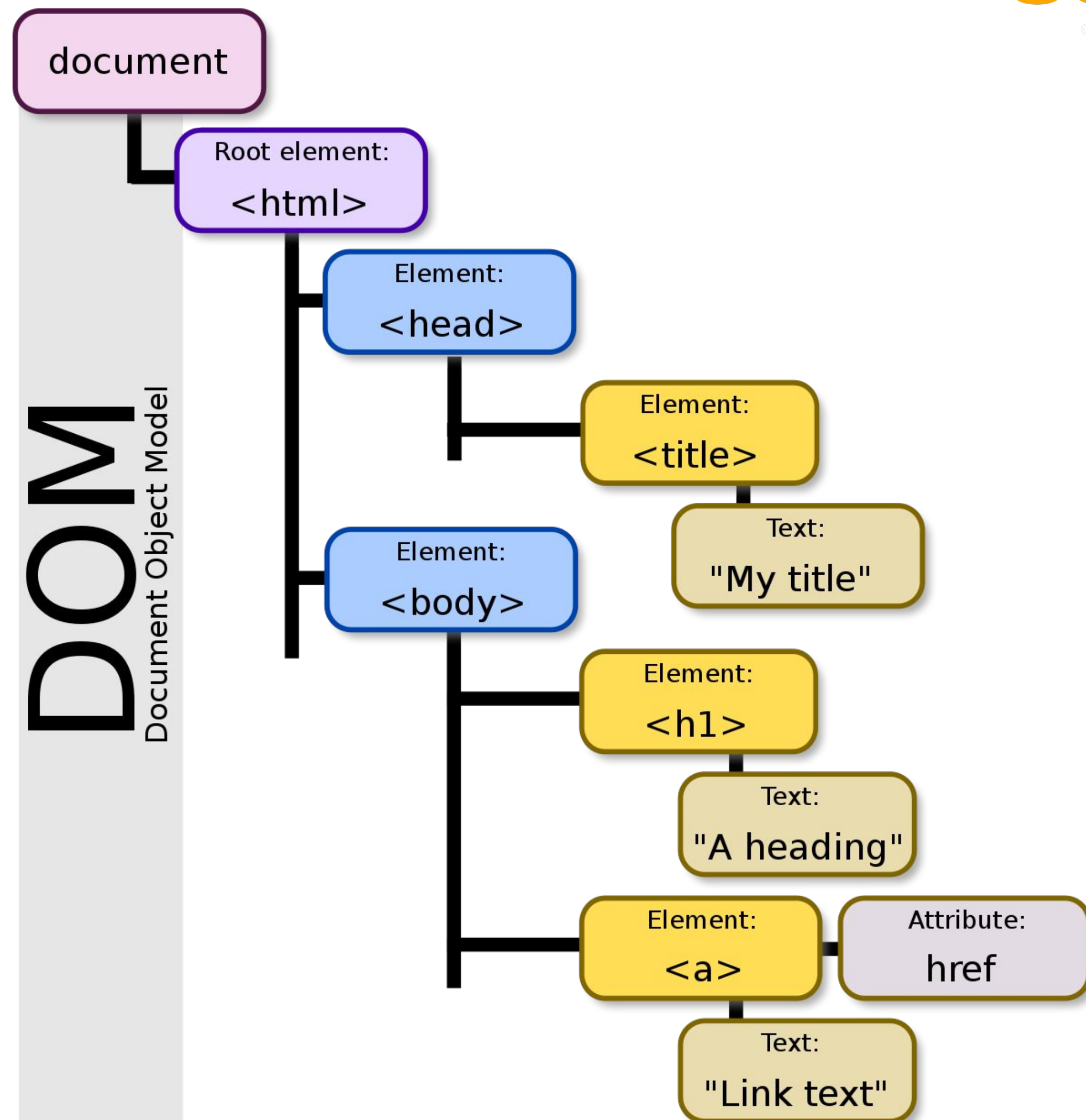


Let's take a quick break

Learning objectives

1. Understand how to add forms
2. Understand how to add buttons
3. Understand how to add tables





BLOCK:



INLINE:



Forms

We use **<form>** to collect user input especially contact details e.g. names, emails, numbers etc.

Forms have special elements which are known as controls. These include submit buttons, radio buttons, checkboxes...

Users enter text, choose options... then submit the form to a web server for further processing.

For instance, you could set things up so their completed form comes to your email.

```
<form action="action_page.php">

  <label for="name">Name</label>
  <input type="text" id="name" name="name" placeholder="Your name">
  <label for="email">Email</label>
  <input type="text" id="email" name="email" placeholder="Enter Email">

  <label for="subject">Subject</label>
  <textarea id="subject" name="subject" placeholder="How can I help?"
  style="height:250px">
</textarea>
  <input type="submit" value="Submit">
```

```
input[type=text], textarea {
  border-radius: 4px; /* Rounded borders */
  box-sizing: border-box; /* Make sure that things stay in place */
  margin: 6px 0px 16px 0px; /* Top right bottom left margin */
  padding: 12px; /* Some surrounding space */
  resize: vertical; /* Allow users to vertically resize the textarea */
  width: 100%; /* Full width */
```

Buttons

The **<button>** tags create a button which users can click on - inside the element we can put text and formatting tags.

It's best practice to tell browsers what type it is by specifying the type attribute for a button element.

There are several ways to style a button. You can make it look like one of those call to action ones prompting visitors to do something.

What sorts of colour schemes do you think would make sense for that?

```
<button type="button">Click Here</button>
```

```
/* This would go at the end of our contact form */  
<input type="submit" value="Submit">
```

```
/* Styling the submit button */  
input[type=submit] {  
  background-color: purple;  
  color: white;  
  padding: 12px 20px;  
  border: none;  
  border-radius: 28px;  
  cursor: pointer;  
}
```

```
/* Styling it for the hover state (mouse moving on top */  
input[type=submit]:hover {  
  background-color: #45a;  
}
```


Tables

<table> tags are used for - yup, you guessed it!

<tr> tags indicate each table row.

<th> tags are for the headers.
They are bold and centered.

<td> tags wrap data points / cells.
They are regular and left-aligned.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Businesses</th>
  </tr>
  <tr>
    <td>Sharmadean</td>
    <td>Reid</td>
    <td>5</td>
  </tr>
  <tr>
    <td>Izzy</td>
    <td>Obeng</td>
    <td>3</td>
  </tr>
</table>
```



Another brief break

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said



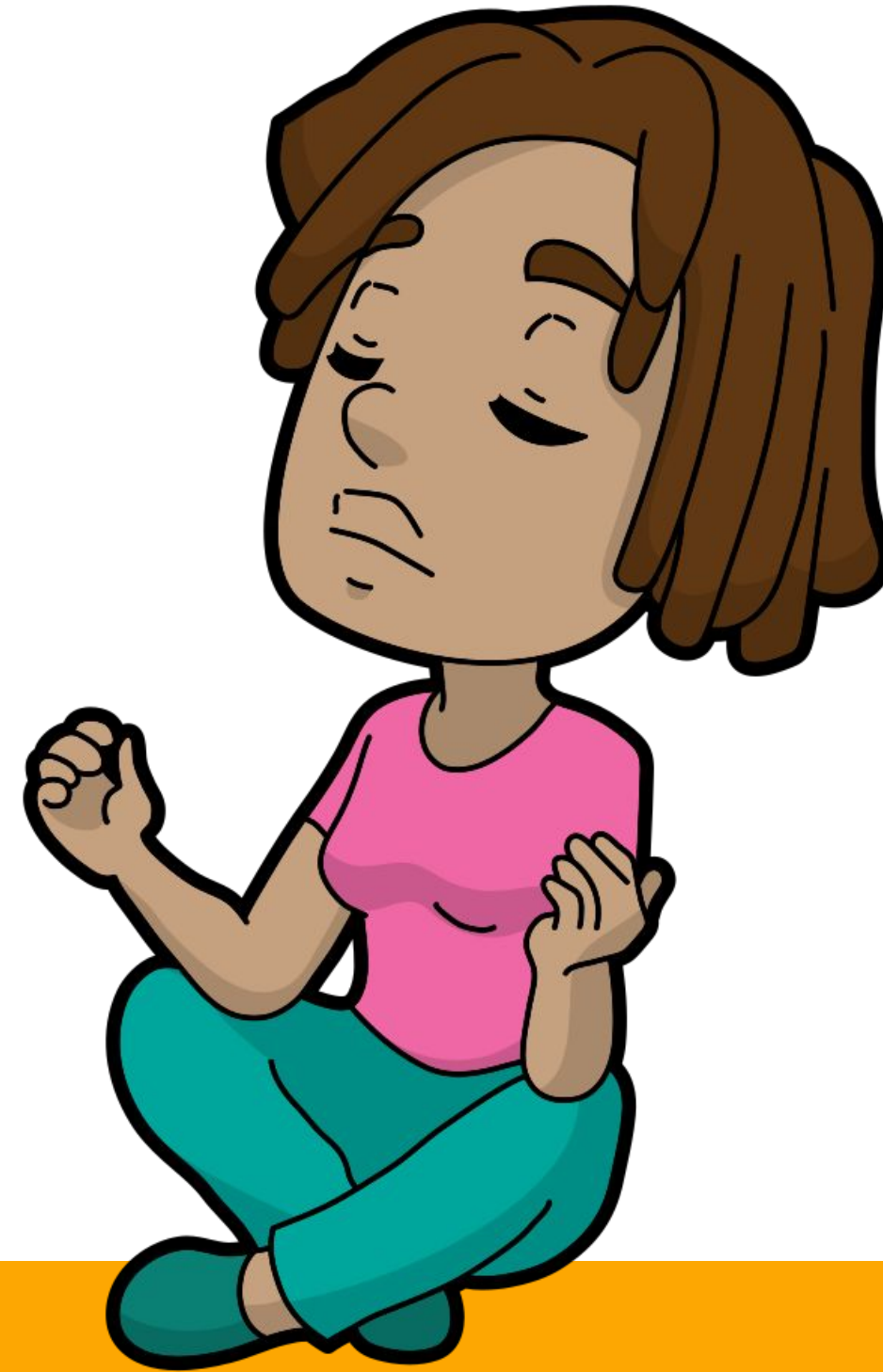
Task 1: Enhance your text

Go back to your draft portfolio, or create another **index.html** file to check what you recall from last time and notes.

Format text to show **importance / significance** so screenreaders convey *emphasis* - always consider **accessibility**.

Now create a **contact form**. Squeeze in an action button too. Lastly, is there anywhere you can use the other **tags, lists, quotes, and or tables?**



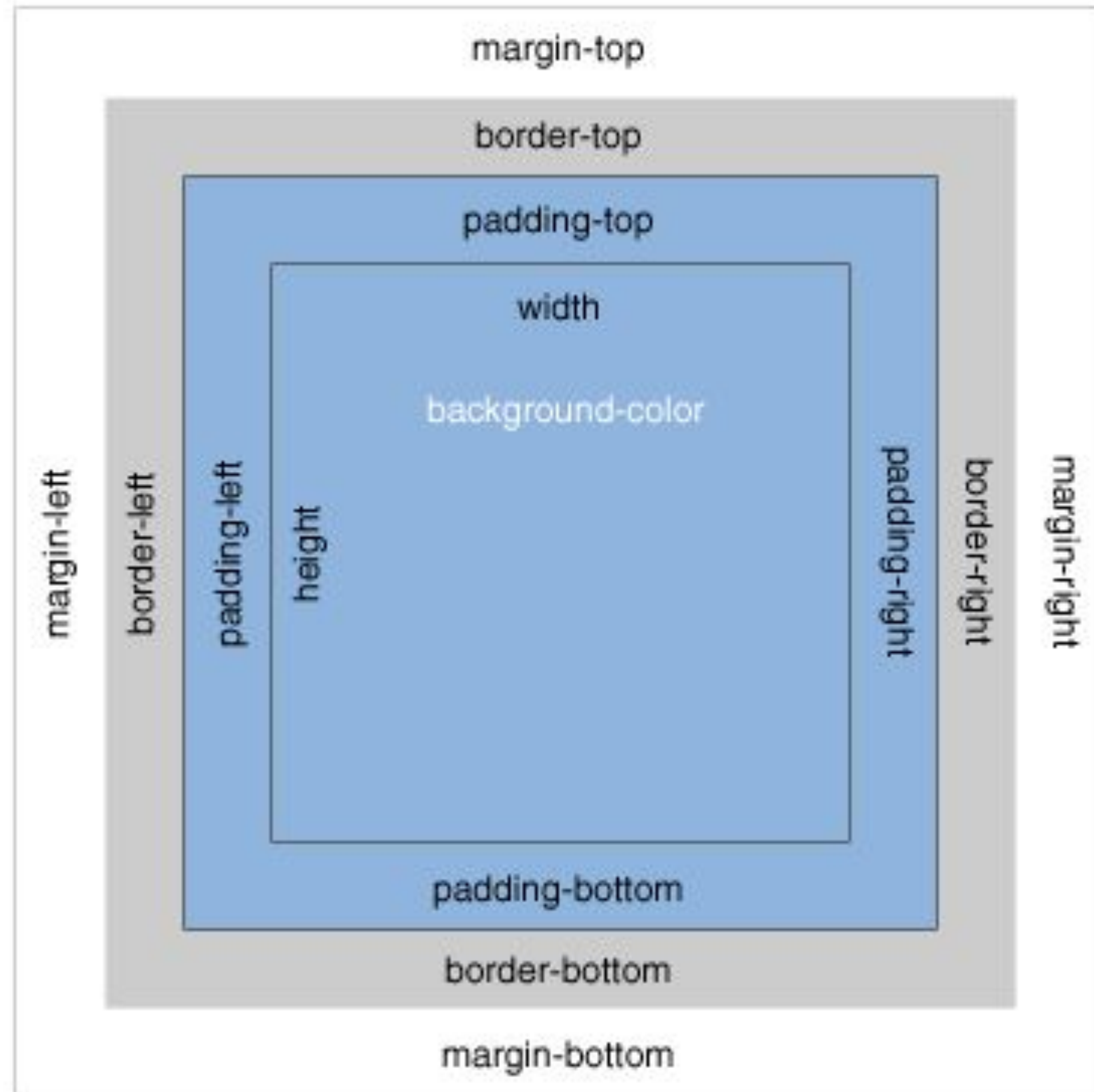


How about a meditative break?

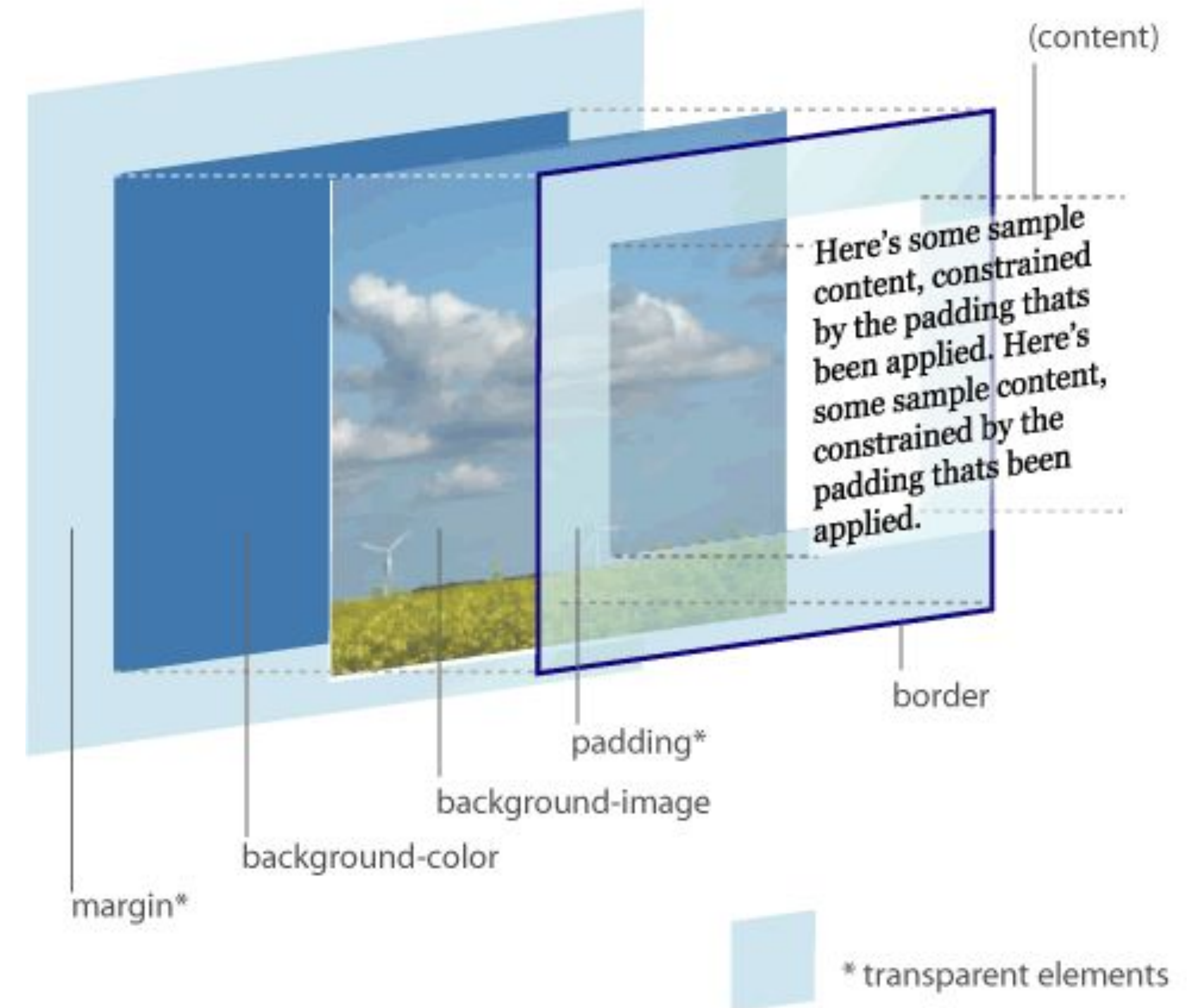
Learning objectives

1. Understand how to embed images
2. Understand how to embed audio
3. Understand how to embed video





THE CSS BOX MODEL HIERARCHY



Images

The **** tag is used to embed an image in a web page.

Images are not technically inserted; they are linked to pages. So the **** tag creates a holding space for the referenced image.

**** is empty: it has no closing tag and only has attributes. There are two attributes required:

src - the file path to the image

alt - an alternate text for the image. which is important for accessibility. (Width and height are optional.)

```
<!DOCTYPE html>
<html>
<body>

</body>
</html>
```

```
/* Choose a couple to play with in style.css */
img1      {opacity: 0.5; }
img2      {filter: grayscale(100%); }
img3:hover {transform: scaleX(-1);}
```


Audio

We use **<audio>** tags for this.

The *controls* attribute adds options like play, pause, and volume. Alternatively, to start a video automatically you can use *autoplay*.

The **<source src>** element allows us to specify alternative files which the browser can choose from, specifying the **type** too.

```
<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="song.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

</body>
</html>
```

Videos

For our third multimedia, **<video>** tags of course!

It's a good idea to always include *width* and *height* attributes. If they are not set, the page might flicker while the video loads.

Again, the error message is only displayed when videos are not supported.

As we did in the previous session, you can of course use a link. Security is vital, so ideally, ensure you use https: vs http. (Hypertext Iransfer Protocol Secure.)

```
<!DOCTYPE html>
<html>
<body>

<video width="320" height="240" autoplay>
  <source src="clip.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said



Task 2: Mix it up with multimedia

Finally a bit more vibrancy on the page!

Make another **index.html** file. Choose a couple of **images (or audio/video)** to add in, and try to look up some other presentation options to use which are different to the ones shown on the slides. Put these into your **style.css** .

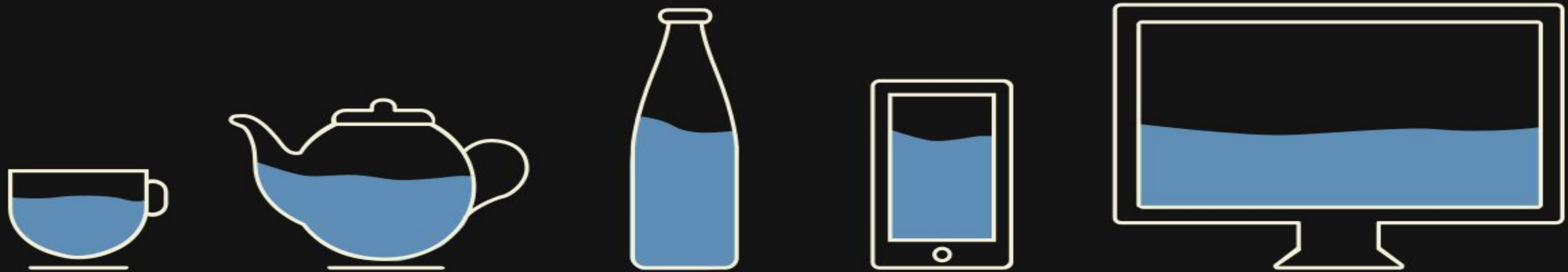
If you're okay to share, present what you've done to the rest of your group. They then have to guess what your site is about (don't include any text!)





Time to take a break

CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Responsive web design

Responsive web design is creating sites that automatically adjust to the screen sizes of different devices, especially mobiles.

According to the Office for National Statistics (ONS), 72% of people in the UK tend to access the internet most often using a phone.

The **<meta name="viewport">** tags gives browsers instructions on how to control page dimensions and scaling.

This resizes, hides, shrinks, or enlarges, a website to make it look good on all computers, tablets, and phones etc.

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<meta name="viewport"
content="width=device-width, initial-scale=1.0">
<body>
</body>
</html>

<!-- Viewport is the browser window size. 1vw =
1% of viewport width. If the viewport is 50cm
wide, 1vw is 0.5cm.-->
```

Fill in the gaps

Responsive web design improves the _____ of our pages.

_____/____ embeds multimedia and
____ enables accessibility.

What are the different attributes we can
use for the above? _____ x _____



Checkpoint!

How are you feeling?

RED - I have no idea what you're talking about

YELLOW - I have some questions but feel like I understand some things

GREEN - I feel comfortable with everything you've said





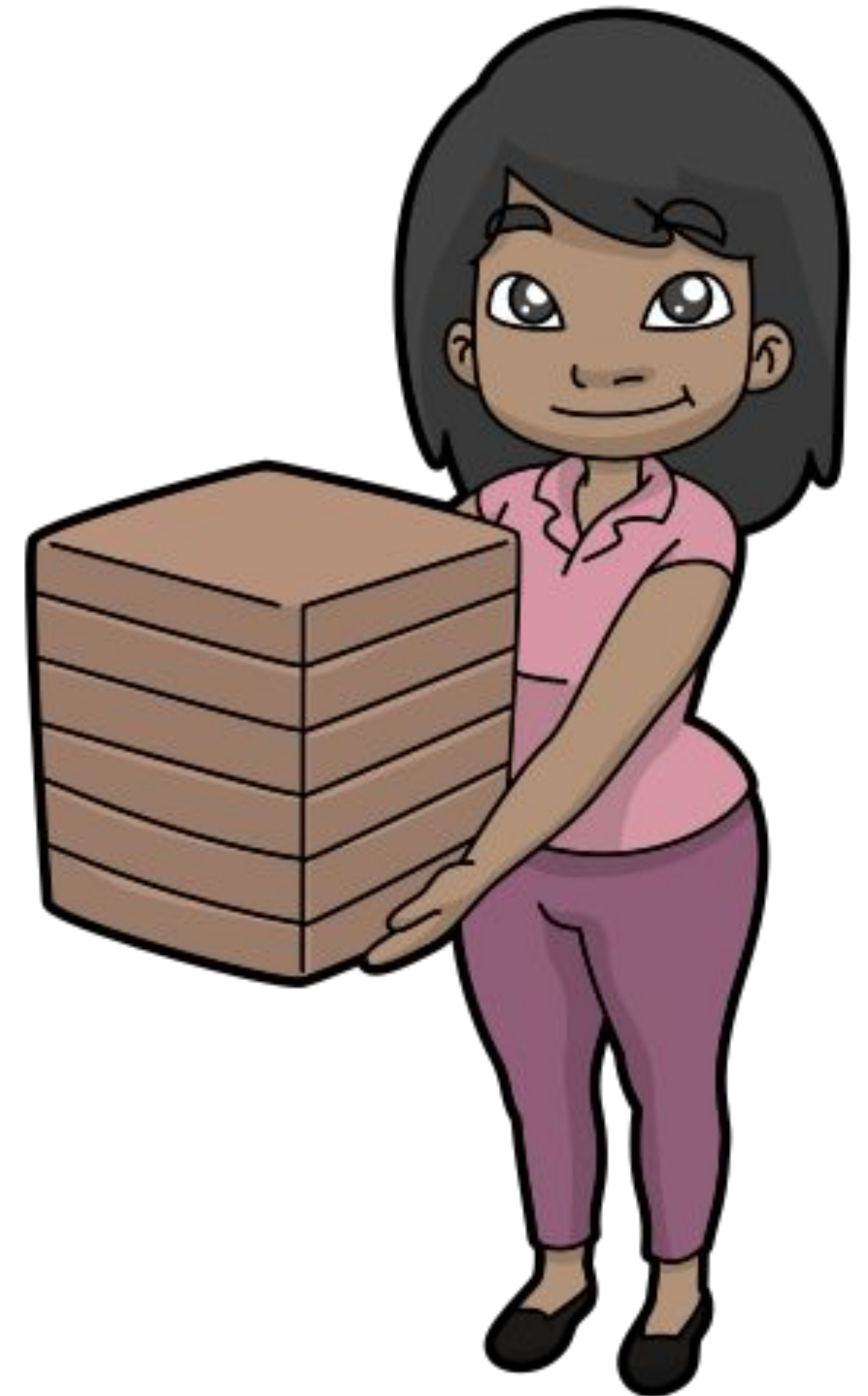
One final break

Task 3: Pushing our work to GitHub

Let's review the homework from last week. After that, we'll move our `index.html` and `style.css` files into our initialised Git repository if we need to (perhaps you were already working from within that directory).

Then we're going to add them so that their status shows as having been staged. Then it's time to commit - with a meaningful message of course.

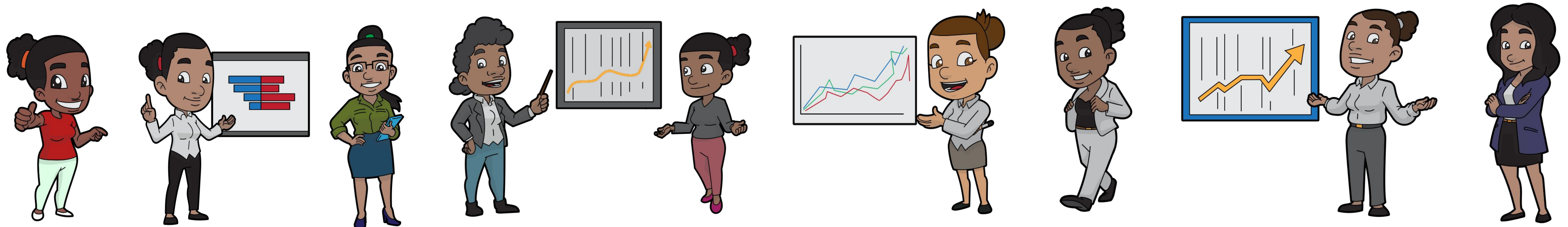
We'd need to have ensured that GitHub is remotely connected to our local folder. Once that's in place, we'll push and pull as needed: following similar steps.



Summary: What you've learnt so far

- Tags, elements, declarations, properties, property values, content, comments, browser compatibility
- Layout: **<!DOCTYPE html>**, **<html lang>**, **<head>** for non visual e.g. metadata, **<meta charset>**, **<title>**
- External CSS **<link rel>**, internal style, inline attribute +3 simple *selectors*: **name**; **ID & class** attributes;
- Responsive web design with **<viewport>**, universal selectors (and many more), {indentation}, nesting;
- Semantic web, **<body>** for visuals: **<header>**, **<footer>**, **<nav>**, **<main>**, **<section>** (and many more)
- Hierarchy **heading** up and down **<h1>** - **<h6>**, **<p>**aragraph, **
, **<a href>nchor link, different states
- Non semantic block level containers e.g. **<div>**, and inline ones like **** (and many more -Google!)
- Formatting **** and for ****phasis vs **** and **<i>** and inserting ****, **<audio>**, **<video>**s
- Security, accessibility, testing, standards, documentation, style guides, SEO, c.theory, design thinking
- GUI + CLI, Git commands e.g. git init, git status, git add, git commit, git remote, push to GitHub

P.S. Don't forget your cheat sheets and our best friend Google!



Homework: Start your portfolio

Begin creating the baseline `index.html` and `style.css` files for your one page portfolio / personal profile. You can start afresh or build upon what you've already done these past 2 sessions - it's up to you. Try to use as many of the things we've covered as possible. Keep saving your work by pushing your commits to GitHub.

When doing the homework, please try to review the slides first of all if there is something which is unclear as this will be the fastest resolution. If you google too, even better! Feel free to use the search function to search for a word and or phrases if it's not clear where it they might have appeared. You can always use the I need help channel in Slack or ask for feedback.



HTML5 Element Index

Head	Sections	Grouping	Tables	Forms	Forms 2	Embedded	Text-level	Text-level 2
doctype	body	p	table	form	keygen	img	a	time
html	article	hr	caption	fieldset	output	iframe	em	code
head	nav	pre	thead	legend	progress	embed	strong	kbd
title	aside	blockquote	tbody	label	meter	object	i, b	samp
base	section	ol	tfoot	input	Interactive	param	u	var
link	header	ul	tr	button		video	s	mark
meta	footer	li	th	select		audio	small	bdi
style	h1-h6	dl, dt, dd	td	datalist	summary	source	abbr	bdo
script	main	figure	col	optgroup	command	canvas	q	ruby, rt, rp
noscript	address	figcaption	colgroup	option	menu	track	cite	span
		div		textarea	Edits	map	dfn	br
						area	sub, sup	wbr
					del, ins			

More Posts from [HTML5 Doctor](#)

SPONSORS

BLACK CODHER

CODING PROGRAMME



We made it!

BLACK CODHER

CODING PROGRAMME



That's a wrap on week 2! See you next week!

< CODING
BLACK
FEMALES >

