

- Learn how to write HTML, CSS and JavaScript
- Skills required for the exam and can be used in your programming project.



What do OCR students need to know for the A Level qualification?

According to the OCR specifications for the AS and A level qualifications, OCR students are expected to have an awareness of the following HTML tags, and any other tags used in an examination question will be introduced in the question itself.

<html>

<link> to link to a CSS file

<head>

<title>

<body>

<h1> <h2> <h3>

 including the src, alt, height and width attributes.

<a> including the href attribute.

<div>

<form>

<input> where the input is a textbox (i.e. has the attribute type="text" and another attribute name to identify it) or a submit button (i.e. has the attribute type="submit")

<p>

<script>

Any other elements used will be introduced in the question.



HTML (HyperText Markup Language):

Purpose:

HTML is the foundation of any webpage. It provides the structure and content of the site.

Function:

Think of HTML as the skeleton of a webpage. It defines the elements like headings, paragraphs, images, and links. Each element is enclosed in tags, like `<h1>`, `<p>`, or ``.

Example:

```
<h1>Welcome to My Website</h1>
```

```
<p>This is a paragraph of text on my webpage.</p>
```



CSS (Cascading Style Sheets):

Purpose:

CSS is used to style and design the HTML content, making it look appealing.

Function:

CSS controls the layout, colors, fonts, spacing, and overall visual presentation of a webpage. It's like adding clothes to the skeleton that HTML provides.

Example:

```
h1 {  
    color: blue;  
    font-size: 24px;  
}  
p {  
    font-family: Arial, sans-serif;  
    line-height: 1.5;  
}
```



JavaScript:

Purpose:

JavaScript adds interactivity and dynamic features to a webpage.

Function:

It enables things like buttons that react when clicked, forms that validate input, and content that updates without needing to reload the page.

JavaScript is like the muscles and brain, making the webpage interactive and responsive.

Example:

```
<body>
  <h1 id="myHeading">Hello, World!</h1>
  <button id="myButton">Click Me</button>

  <script>
    // Get the button element by its ID
    const button = document.getElementById('myButton');

    // Add a click event listener to the button
    button.addEventListener('click', function() {
      // Change the text inside the heading element
      document.getElementById('myHeading').textContent = 'You clicked the button!';
    });
  </script>
</body>
```



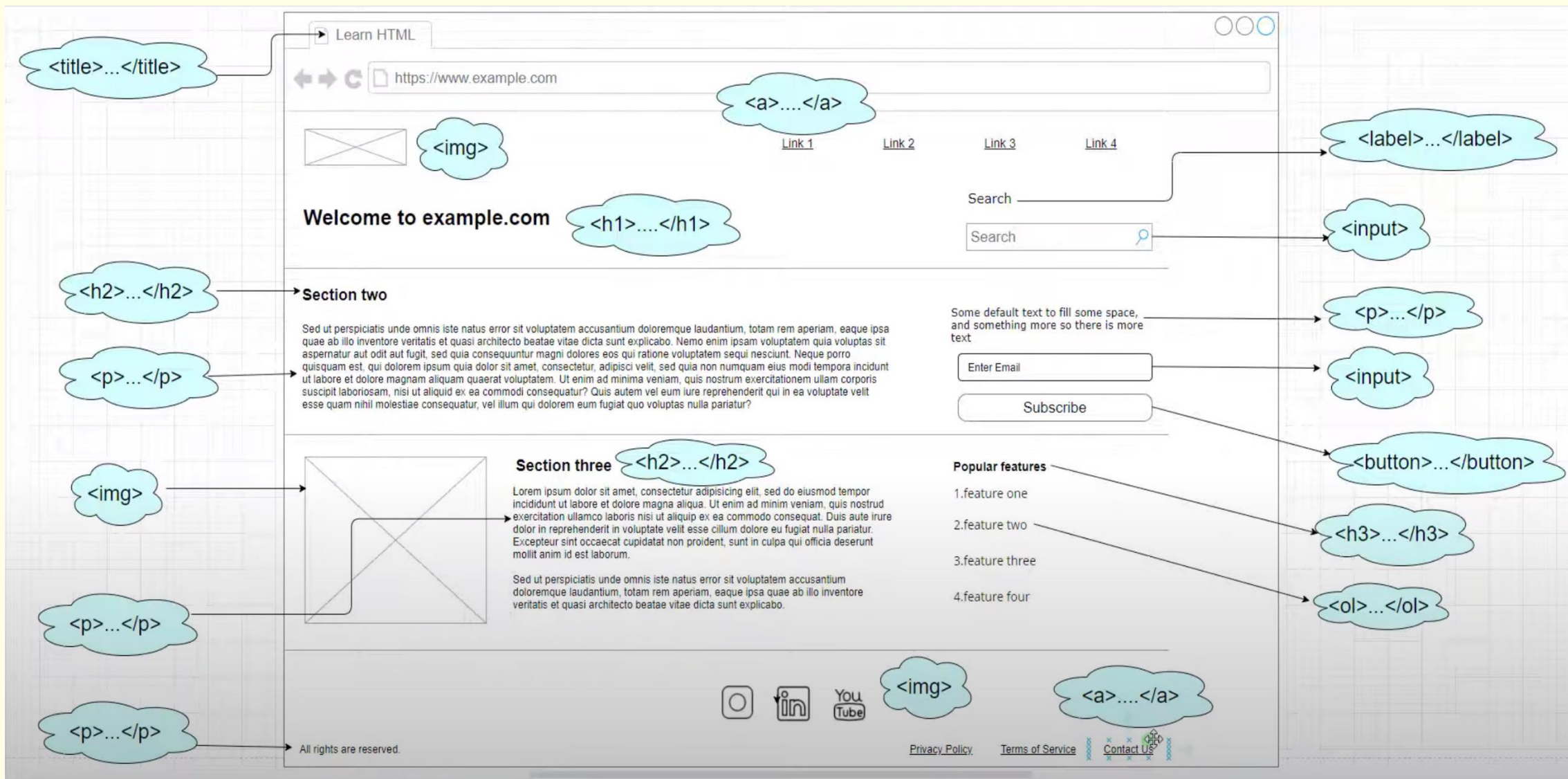
What is HTML?

- Web pages are coded using HTML
- Hypertext Mark Up Language (HTML)
- Using **HTML tags** to structure the layout and content of a web page
- A **Web Browser** is able to process the HTML code and display the web page

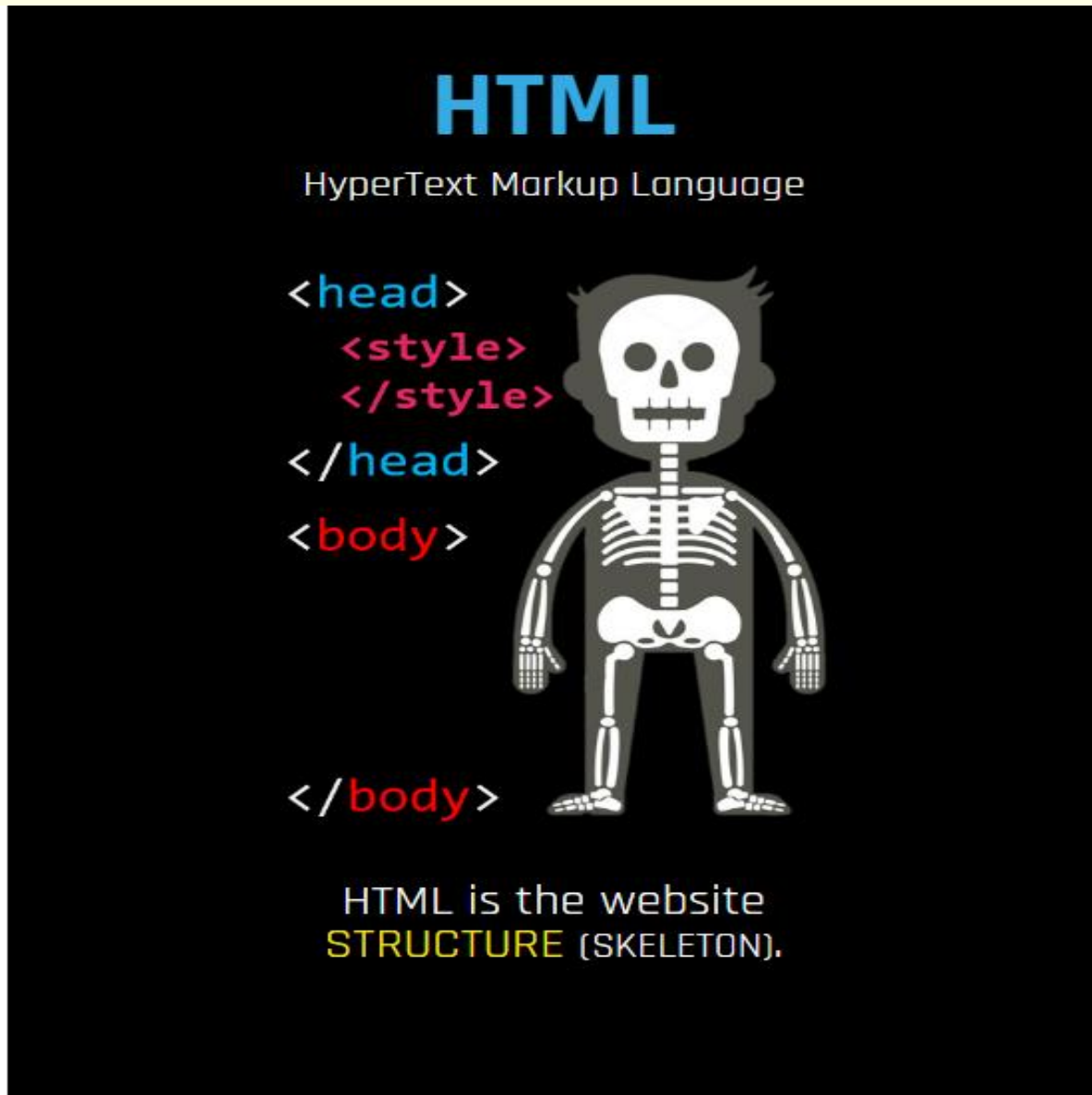
```
trinket  Autorun  Share
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Pugs</title>
5 </head>
6 <body>
7
8 <h1>This is a Heading</h1>
9 <p>This is a paragraph.</p>
10 
11 <p></p>
12 <a href="https://www.akc.org/dog-breeds/pug/">Link to a website about pugs</a>
13
14 </body>
15 </html>
```







Go to <https://me.codetribes.com/new-account>



```
<html>
```

```
<head>
```

```
<style>
```

```
h1{ font-size: 75px; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Name and Surname</h1>
```

```
<h3>MOTIVATION:
```

```
<br><br>
```

```
<i>A short sentence on why you are  
learning to code</i>
```

```
</h3>
```

```
<h3>Launching Soon...</h3>
```

```
<p>10 October, 2009</p>
```

```
</body>
```

```
</html>
```



HTML Tags Explained - <!DOCTYPE html>

The **first tag** in a web page should be:

<!DOCTYPE html>

This tells the web browser that we are using HTML5 code.

The very last HTML tag on a web page should be the <html> closing tag:

</html>



HTML Tags Explained - <head>

After the <HTML> Tag is the <HEAD> section:

```
<head>
```

```
</head>
```

Tags have an opening and closing tag – if you forget to use the closing tag your page will not display properly!

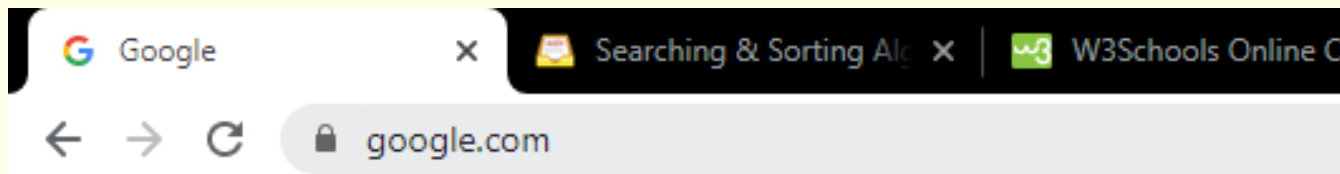
In this example in between the <head> tags we have a <title> tag:

```
<head>
```

```
<title>Page Title</title>
```

```
<head>
```

We can use this to display text in Browser tab



Using the <head> section

- The <head> section is used for:

Document Metadata:

- <title>: Specifies the title of the webpage, which appears in the browser tab and is used by search engines.

Styles:

- <link>: Connects to external resources like stylesheets (CSS) or icons. For example, <link rel="stylesheet" href="styles.css">

Scripts:

- •<script>: Can be used to include JavaScript files or scripts that should be loaded before the body content. Example: <script src="script.js" defer></script>.

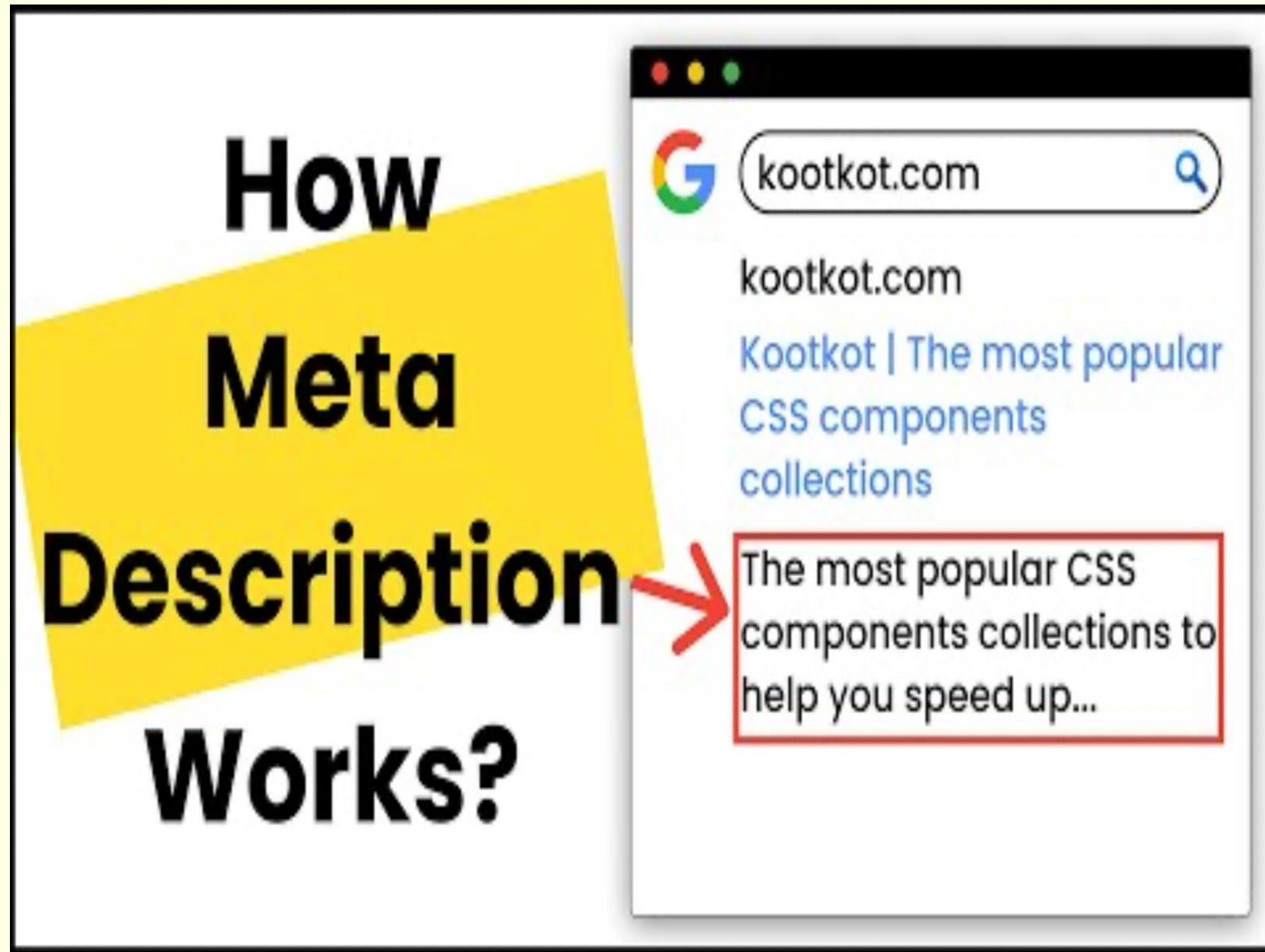
Other Meta Information:

- <meta name="description" content="A brief description of the page">: Provides a description of the webpage content, used by search engines in search results.

Favicon:

- <link rel="icon" href="favicon.ico">: Specifies the icon that appears in the browser tab.





**Why
Page
Keywords?
How?**



<https://youtu.be/GvZBx8zokmo?feature=shared>



HTML Tags Explained - <body>

Everything that we put in between the body tags is what actually gets displayed on the screen (in the browser window)

```
<html>
```

```
<head>
```

```
  <title> </title>
```

```
  <style>
```

```
    ...
```

```
  </style>
```

```
</head>
```

```
  <body>
```

```
    ...
```

```
    ...
```

```
  </body>
```

```
</html>
```



HTML Tags Explained – formatting tags

You will see that inside the body tags we are using:

```
<h1> ....</h1>
```

```
<p>...</p>
```

These are formatting tags.

So if you put some text in between:

`<h1> Some Text </h1>` it will format this to be heading1 style

`<p> some text </p>` it will format this to paragraph style

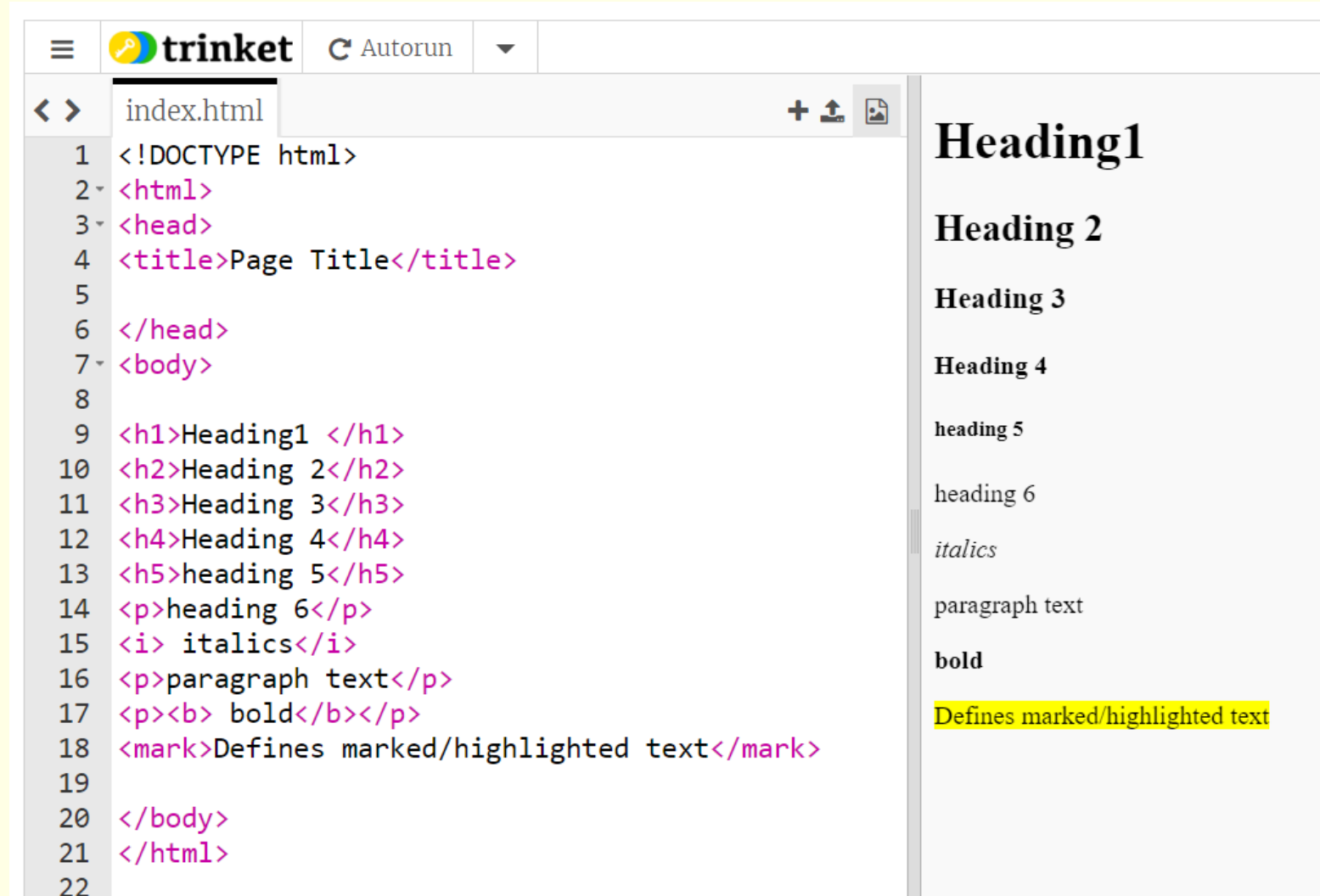
`<p>some text </p>` if you put this `` tag around the `<p>` tag it will make the text bold too.

```
trinket  Autorun  Share
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Pugs</title>
5 </head>
6 <body>
7
8 <h1>This is a Heading</h1>
9 <p>This is a paragraph.</p>
10 
11 <p></p>
12 <a href="https://www.akc.org/dog-breeds/pug/">Link to a website about pugs</a>
13
14 </body>
15 </html>
```



HTML Tags Explained

Other formatting tags:



The screenshot shows the Trinket online code editor interface. The top bar includes the Trinket logo, an 'Aautorun' button, and a dropdown menu. The editor window displays a file named 'index.html' with the following HTML code:

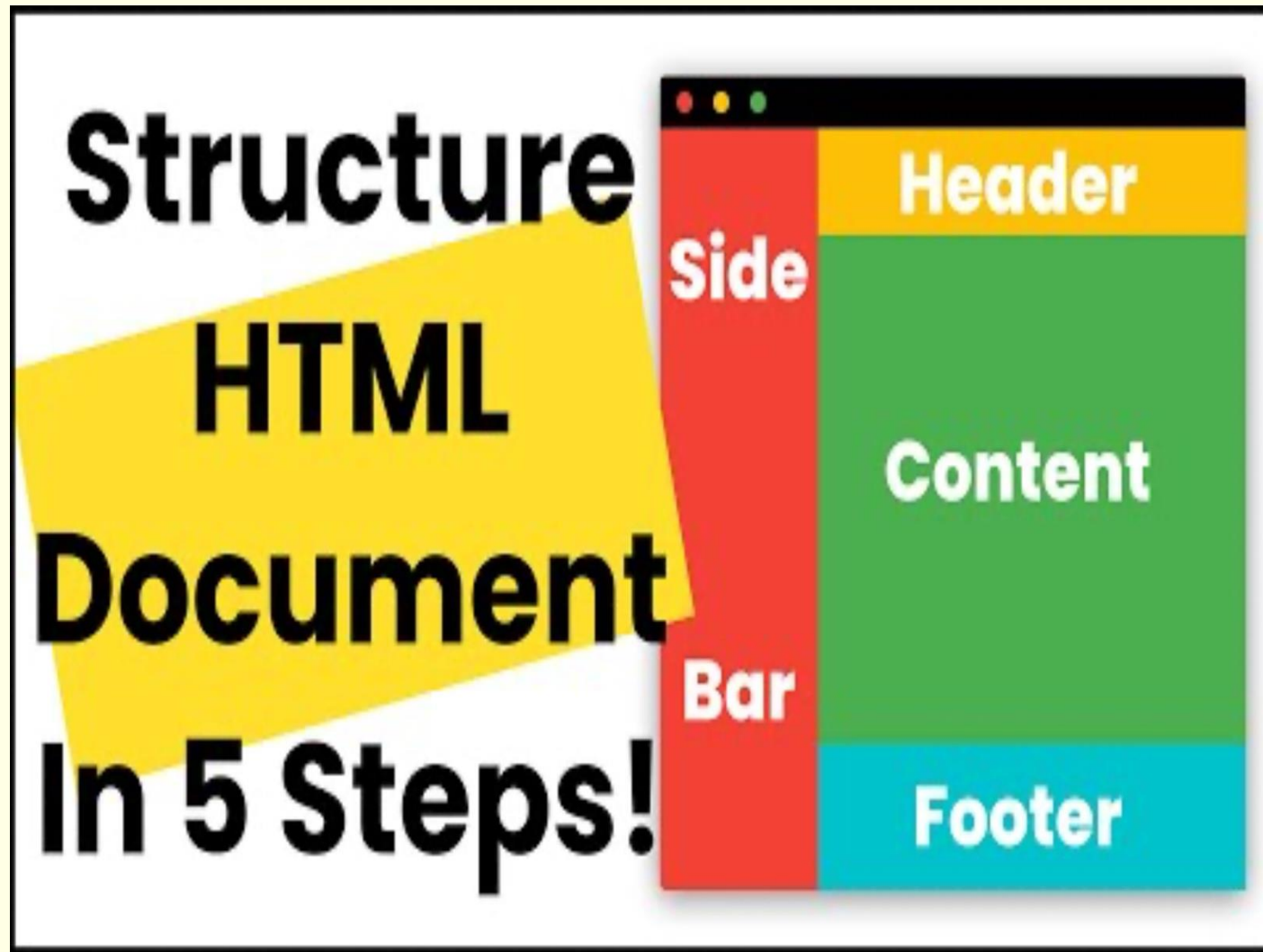
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Page Title</title>
5
6 </head>
7 <body>
8
9 <h1>Heading1 </h1>
10 <h2>Heading 2</h2>
11 <h3>Heading 3</h3>
12 <h4>Heading 4</h4>
13 <h5>heading 5</h5>
14 <p>heading 6</p>
15 <i> italics</i>
16 <p>paragraph text</p>
17 <p><b> bold</b></p>
18 <mark>Defines marked/highlighted text</mark>
19
20 </body>
21 </html>
22
```

The right-hand pane shows the rendered output of the code:

- Heading1
- Heading 2
- Heading 3
- Heading 4
- heading 5
- heading 6
- italics*
- paragraph text
- bold**
- Defines marked/highlighted text



How to write a standard HTML document?



<https://youtu.be/ZEZOrZLLoW0?feature=shared>



HTML Tags Explained

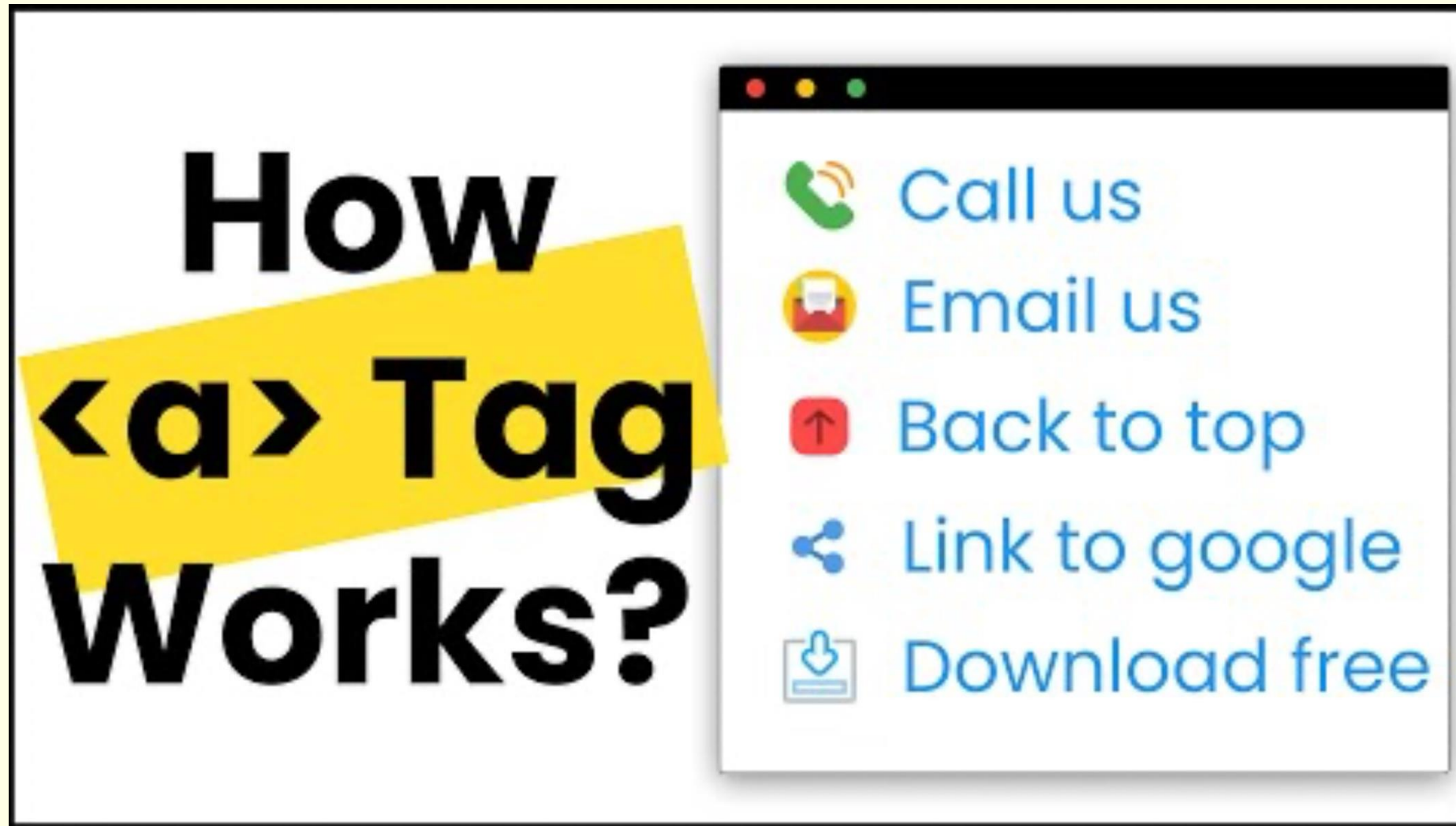
Image tags

We can add images too using this tag:

```

```





HTML Tags Explained – Anchor tags

Hyperlinks

A hyperlink in a web page can be used to navigate to another web page.

We can add hyperlinks by using:

`Go to this webpage`

```

  trinket  Autorun  Share
  index.html
  1 <!DOCTYPE html>
  2 <html>
  3 <head>
  4 <title>Pugs</title>
  5 </head>
  6 <body>
  7
  8 <h1>This is a Heading</h1>
  9 <p>This is a paragraph.</p>
 10 
 11 <p></p>
 12 <a href="https://www.akc.org/dog-breeds/pug/">Link to a website about pugs</a>
 13
 14 </body>
 15 </html>
```



What Are Anchor Tags?



Anchor tags define hyperlinks.



This is the page, website or page section you want to jump to.



Anchor tags start with **<a>**.



Anchor tags can define **INTERNAL** or **EXTERNAL** hyperlinks.



Internal Hyperlinks



An internal hyperlink will link to another page in your website.



It might also link to another section in the same web page.



This is used for web pages with very large amounts of text.



External Hyperlinks.



External Hyperlinks will link to a separate website.



These are used when you want to reference another site.



This could be used for creating SEO backlinks.




Anchor Tag Attributes

attribute	meaning	HTML	description
href	The address of the page you are linking to	<code></code>	This will link to the BBC.
target	Specifies whether a link will open in a new tab or the same tab.	<code></code>	Opens a link to the BBC in a new browser window



How to create different types of lists?

**How
, ,
 Tags
Work?**



- 1 Questions
- Articles
- References
- ★ Sign
- ▲ Station
- # New
- Blind
- * Resident

<https://youtu.be/53zJ63OiNpo?feature=shared>



List Type	Tag	Description	Example	Render Result
Ordered List		Displays items in a numbered sequence.	<pre> First item Second item Third item </pre>	<ol style="list-style-type: none"> 1. First item 2. Second item 3. Third item
Unordered List		Displays items with bullet points.	<pre> Item one Item two Item three </pre>	<ul style="list-style-type: none"> • Item one • Item two • Item three
Definition List	<dl>	Used for terms and their definitions.	<pre><dl> <dt>HTML</dt> <dd>Hypertext Markup Language</dd> <dt>CSS</dt> <dd>Cascading Style Sheets</dd> </dl></pre>	HTML: Hypertext Markup Language CSS: Cascading Style Sheets
Nested Lists	Nested	Lists within lists to create complex structures.	<pre> Item one Subitem one Subitem two Item two </pre>	<ul style="list-style-type: none"> • Item one <ul style="list-style-type: none"> ◦ Subitem one ◦ Subitem two • Item two



Common HTML Tags

HTML Tag	Purpose	Example
<html>	Root element of an HTML document	<html>...</html>
<head>	Contains meta-information about the document	<head><title>My Webpage</title></head>
<title>	Sets the title of the webpage	<title>My Webpage</title>
<body>	Contains the visible content of the webpage	<body><h1>Welcome</h1><p>Content</p></body>
<h1> to <h6>	Define headings with <h1> as the largest	<h1>Main Heading</h1><h2>Subheading</h2>
<p>	Defines a paragraph of text	<p>This is a paragraph.</p>
<a>	Creates a hyperlink	Visit Example
	Embeds an image	
	Creates an unordered (bulleted) list	Item 1Item 2
	Creates an ordered (numbered) list	First ItemSecond Item
	Defines a list item	Item
<div>	Block-level container for grouping content	<div class="container">...</div>
	Inline container for styling text	red
<form>	Defines a form for user input	<form action="/submit" method="post">...</form>
<input>	Defines an input field	<input type="text" placeholder="Enter your name">
<button>	Creates a clickable button	<button type="button">Click Me</button>
<table>	Creates a table	<table>...</table>
<tr>	Defines a row in a table	<tr><td>Data</td></tr>
<th>	Defines a header cell in a table	<th>Header</th>
<td>	Defines a data cell in a table	<td>Data</td>



Forms – used to collect user data

- ▶ Text field – This defines a one-line input field for text input.

```
<form>
  Please enter your first name:<br>
  <input type="text" name="firstname">
  <br>
  Please enter your age:<br>
  <input type="text" name="age">
</form>
```

- ▶ Text area – Allows you to have more user input.

```
<form>
  <textarea name="textarea"> Please enter your text here</textarea>
</form>
```

- ▶ Submit button – Defines a button for submitting a form to a form handler.

```
<form action="test_page.php">
  Where were you born?:<br>
  <input type="text" name="birthplace">
  <br>
  Where do you currently live?:<br>
  <input type="text" name="livingplace">
  <input type="submit" value="submit">
</form>
```

- ▶ Radio buttons – Radio buttons let a user select one of a limited number of choices.

```
<form>
  Are you sure?
  <input type="radio" name="validation" value="yes" checked>YES
  <br>
  <input type="radio" name="validation" value="no">NO
</form>
```

- ▶ Check boxes – As opposed to radio buttons, which only let you select one of a limited number of choices, check boxes allow you to select more than one choice.

```
<form>
  I own the following:<br>
  <input type="checkbox" name="games">Playstation<br>
  <input type="checkbox" name="games">XBox<br>
</form>
```





CSS stands for **Cascading Style Sheet**

CSS is used to apply the same styles and layout across all pages in a website. For example: same background colour, text size, text colour, how text is aligned on the web pages.

To add styles you can use the <style> tag:

```
<head>
```

```
<link href="styles.css" rel="stylesheet">
```

```
</head>
```



What do OCR students need to know for the A Level qualifications?

According to the OCR specifications for the AS and A Level qualifications, OCR students are expected to have an awareness of the following CSS instructions, and any other properties used in an examination question will be introduced in the question itself.

Learners are expected to be able to use CSS directly inside elements using the style attribute

```
<h1 style="color:blue;">
```

and external style sheets. In the style sheets they should be able to use CSS to define the styling of elements:

```
h1{  
color:blue;  
}  
classes  
.infoBox{  
background-color: green;  
}
```

and Identifiers

```
#menu{  
background-color: #A2441B;  
}
```

They are expected to be familiar with the following properties.

background-color
border-color
border-style
border-width
color with named and hex colours
font-family
font-size
height
width

Any other properties used will be explained in the question.



Example

If you want the same font style and background colour across all pages then you use these styles:

```
body{  
background-color:lightblue;  
font-family: Arial, Helvetica, sans-serif;  
}
```

If we want all the <p> tag text to be the colour blue and font-size 14 we can use a style to do this:

```
p{  
color: lightblue;  
font-size: 14;  
}
```



Colour codes:

Use can use either colour names or HEX colours code

```
h1 {  
  color: #2b6cc4;  
}
```

Go to <https://www.color-hex.com/>

gold #D4A017	goldenrod #EDDA74	yellow #FFFF00	orange #F87A17
red #FF0000	sienna #CCECEC	firebrick #800517	deeppink #F52887
magenta #FF00FF	orangered #F63817	brown #980517	yellowgreen #9ACD32
chocolate #C85A17	coral #F76541	papayawhip #FEECCF	bisque #FDE0BC
khaki #ADA96E	lemonchiffon #FFF8C6	lavender #E3E4FA	peru #C57726
pink #FFC0CB	darkgoldenrod #AF7817	darkkhaki #B7AD59	darkorange #F88017
darksalmon #E18B6B	paleturquoise #DDDDDC	aquamarine #43B7BA	blueviolet #7931DF
darkviolet #842DCE	blue #0000FF	lightslateblue #736AFF	darkturquoise #3B9C9C
slateblue #737CA1	dodgerblue #1589FF	deepskyblue #3BB9FF	cadetblue #578693
darkslateblue #2B3856	cornflowerblue #151B8D	forestgreen #4E9258	darkgreen #254117
darkseagreen #8BB381	lawngreen #87F717	darkolivegreen #4A4117	seagreen #2E8B57
olivedrab #658017	slategray #657383	gray #736F6E	darkslategray #25383C

font-family styles

font-family: Times, "Times New Roman", Georgia, serif;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-family: "Lucida Console", Courier, monospace;
font-family: cursive;
font-family: fantasy;
font-family: math;
font-family: fangsong;

```
5 - body {  
6     background-color: lightblue;  
7     font-family: Verdana;
```



Other Formatting styles:

font-size

text-align – left, right, center

border: 2px solid blue;

```
h1{  
  font-size: 20;  
  text-align: center;  
  border: 2px solid blue;  
}
```

Pioneering women in space

As of March 2021, 65 women have flown in space, including cosmonauts, astronauts, payload specialists, and space station participants. The first woman in space was Russian cosmonaut Valentina Tereshkova, who flew on Vostok 6 on June 16, 1963. The first American woman in space, Sally Ride, flew aboard the Space Shuttle STS-7 in June of 1983.



Formatting images

```
img{  
border: 5px solid darkblue;  
border-radius: 10px;  
}
```

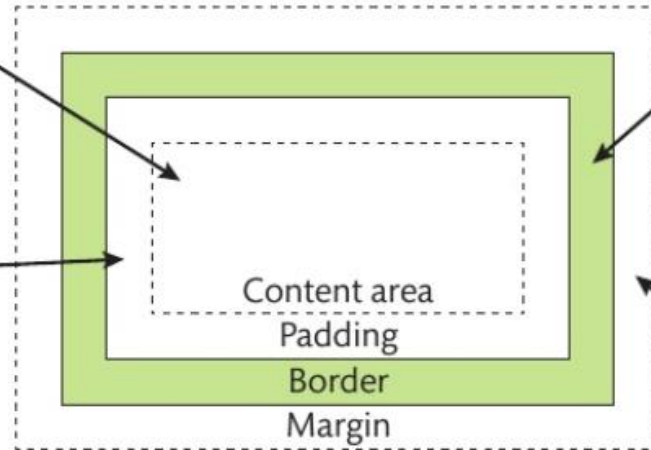


Content area:

This is where any text and images appear.

Padding:

This is the difference between the content and the border.



Border: This is the border that goes around the padding and the content area.

Margin: Margin is the difference from the border.

```
<div class="box">  
  Content Area  
</div>
```

```
.box {  
  width: 200px;      /* Width of the content area */  
  height: 100px;     /* Height of the content area */  
  padding: 20px;      /* Padding inside the box */  
  border: 5px solid blue; /* Border around the box */  
  margin: 30px;       /* Margin outside the box */  
  background-color: lightgray; /* Background color  
of the content area */  
}
```

Content Area



Formatting Hyperlinks

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Example:

```
a {  
  color: hotpink;  
}
```

[Find out more](#)

In addition, links can be styled differently depending on what state they are in.

a:link - a normal, unvisited link

```
a:link {  
  color: red;  
}
```

a:visited - a link the user has visited

```
a:visited {  
  color: green;  
}
```

a:hover - a link when the user mouses over it

```
a:hover {  
  color: hotpink;  
}
```


Example CSS

```
1  /* General styles */
2  ▼ body {
3      font-family: Arial, sans-serif;
4      margin: 0;
5      padding: 0;
6      line-height: 1.6;
7  }
8
9  /* Header styles */
10 ▼ header {
11     background-color: #4CAF50;
12     color: white;
13     padding: 10px 0;
14     text-align: center;
15 }
```

```
17  /* Navigation styles */
18 ▼ nav {
19     background-color: #333;
20     color: white;
21     padding: 10px;
22 }
23
24 ▼ nav ul {
25     list-style: none;
26     margin: 0;
27     padding: 0;
28     text-align: center;
29 }
30
31 ▼ nav ul li {
32     display: inline;
33     margin-right: 20px;
34 }
35
36 ▼ nav ul li a {
37     color: white;
38     text-decoration: none;
39 }
40
41 ▼ nav ul li a:hover {
42     text-decoration: underline;
43 }
```

```
46 ▼ main {
47     padding: 20px;
48 }
49
50 ▼ section {
51     margin-bottom: 20px;
52 }
53
54 ▼ h2 {
55     color: #4CAF50;
56 }
57
58 /* Footer styles */
59 ▼ footer {
60     background-color: #333;
61     color: white;
62     text-align: center;
63     padding: 10px 0;
64     position: fixed;
65     bottom: 0;
66     width: 100%;
67 }
```





```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta name="description" content="A simple webpage using div tags for layout, including navigation.">
5   <title>Simple Web Page with Navigation</title>
6   <link href="styles.css" rel="stylesheet">
7 </head>
8 <body>
9   <header>
10    <h1>Welcome to My Website</h1>
11  </header>
12
13  <nav>
14    <ul>
15      <li><a href="#home">Home</a></li>
16      <li><a href="#about">About</a></li>
17      <li><a href="#contact">Contact</a></li>
18    </ul>
19  </nav>
20
21  <main>
22    <section id="home">
23      <h2>Home</h2>
24      <p>This is the homepage of my website.</p>
25    </section>
26
27    <section id="about">
28      <h2>About</h2>
29      <p>Information about this website.</p>
30    </section>
31  </main>
32
33  <footer>
34    <p>&copy; 2024 My Website</p>
35  </footer>
36 </body>
37 </html>
```

<!DOCTYPE html>: Declares the document type and version of HTML (HTML5 in this case).

<head>: Contains meta-information about the document, like the character set, title, and links to CSS.

<meta > Specify metadata about the document used by search engines

<title>: Defines the title of the page that appears in the browser tab.

<link rel="stylesheet" href="styles.css">: Links an external CSS file to style the webpage.

<body>: Contains the visible content of the webpage.

<header>: Element containing navigation or a logo.

<nav>: Element for navigation links.

<main>: The primary content area

<section> to group related content.

<footer>: Element for the footer, typically containing copyright information or links.



Navigation

- ▶ **Menus** – Menus can appear anywhere within a web page, but most website developers tend to place them towards the top of a web page. Instead of traditional hyperlinks, these menus are more attractive and aesthetically pleasing. Visitors to a website can click on a menu and it will redirect them to the page they are looking for. Alternatively, sometimes when you hover over a menu a sub-menu will appear (see Figure 6.10).



- ▶ **Hyperlinks** – Hyperlinks are links that, when clicked on, take you to a particular part of the website (an internal link) or they will take you to another website (an external link). Hyperlinks can be text, images or buttons.
- ▶ **Anchors** – Anchors are used to redirect a visitor to a certain part of a web page. For example, if you are reading a long document online, at the beginning of the web page there will be links to the beginning of each section within the document. By clicking on one of these section links you will be redirected to the precise point in the web page that you require.



Interactive Components

Example of interactive components include:

- Roll over buttons– A button that changes colour when hovered over and triggers an action when clicked.
- Pop-ups – usually the first time you visit a website a pop up will appear to tell you that cookies is being downloaded to use the website and grant permissions. Pop up's can be annoying when used for advertising!
- Hot spots – an area on an image that is acts as a hyperlink.
- Roll over images
- A simple form that includes text input fields, a checkbox, and a submit button.
- A dropdown menu that allows users to select an option from a list.
- A tabbed interface that allows users to switch between different content sections without leaving the page.
- A simple image carousel/gallery that automatically transitions between images.
- An interactive map using Google Maps or another mapping API.
- An accordion that expands and collapses sections of content.

Go to <https://component.gallery/>



The World Wide Web Consortium

The World Wide Web Consortium (W3C) is a body which promotes the standardisation of web design, especially of HTML. It aims to ensure universal accessibility of the web. For example, they promote the need for all websites to be displayed on a variety of browsers and resolutions so that they are usable by people with special needs. The W3C produces guidelines and tools for standardising websites which contribute towards increased accessibility. HTML compliance plays an important factor in this because if we use outdated HTML coding techniques or conventions this prevents a website from becoming fully accessible.

Link

For more on the W3C and their Web Accessibility Initiative visit their website: <http://www.w3.org/WAI/>



Accessibility Features

Use Semantic HTML:

- Use appropriate HTML tags (e.g., <header>, <nav>, <main>, <article>, <footer>) to structure your content.
- Use heading tags (<h1>, <h2>, etc.) in a logical order to create a clear hierarchy, which helps screen readers and other assistive technologies understand the page's structure.

Ensure Keyboard Accessibility:

- Make sure that all interactive elements (e.g., links, buttons, forms) can be accessed and operated using a keyboard alone, without requiring a mouse.
- Use focus indicators (like outlines or highlights) to show which element is currently selected, helping users navigate using the keyboard.

Alt Text for Images

- **Example:**
- **Description:** Always provide descriptive alt attributes for images. This allows screen readers to describe the image content to visually impaired users. If an image is decorative, use an empty alt attribute (alt="").

Use Sufficient Colour Contrast:

- Ensure that text and background colours have sufficient contrast to be readable by users with visual impairments, including colour blindness.
- Avoid using colour as the sole means of conveying important information. Supplement it with text labels or patterns.



Platform compatibility

When developing websites, it is important that they are tested across a range of different platforms, particularly if it is part of the client's brief that the website should be usable on these platforms. These platforms can be broken into three categories:

- ▶ web browsers
- ▶ operating systems
- ▶ mobile devices.

When testing a website, you must ensure that it has a consistent appearance across these different platforms. The consequence of not doing this is that your website may display incorrectly on one or more web browsers, using a particular operating system or on certain mobile devices. Consider that every user visiting your website may use a different combination of browser and operating system, or be using a mobile device. Therefore, you must

cater for all platforms and test your website to ensure that it appears the same across them all.

Obviously, developing a website to be compatible across multiple platforms is expensive, so this is an important constraint and consideration for clients and website developers. It might be that, initially, a client decides to focus on ensuring compatibility with only the most popular web browsers and the most popular mobile device operating systems, with the intention of ensuring compatibility with the others once the website has been established successfully on the most popular ones.



CSS and Platform compatibility

The @media rule in CSS is used to apply styles based on specific conditions such as screen size, device type, or orientation.

The condition @media screen and (max-width: 767px) specifically targets devices with a screen width of 767 pixels or less, which typically includes mobile phones and small tablets.

When this media query is applied, the styles inside it will only take effect when the screen width is 767 pixels or narrower.

Here's how you can use the @media rule to apply different styles for smaller screens:

- **Explanation:**
- **@media screen and (max-width: 767px):** This media query checks if the screen width is 767 pixels or less.
- **body { font-size: 14px; background-color: lightgray; }:** Changes the font size and background colour for smaller screens.
- **.container { width: 100%; padding: 10px; }:** Sets the container to take up the full width of the screen and adds padding.
- **.nav-menu { display: none; }:** Hides the navigation menu on small screens.

```
1  /* Default styles for larger screens */
2  ▼ body {
3      font-size: 16px;
4      background-color: white;
5  }
6
7  ▼ .container {
8      width: 80%;
9      margin: 0 auto;
10 }
11
12 /* Styles for screens 767px wide or smaller */
13 ▼ @media screen and (max-width: 767px) {
14     ▼ body {
15         font-size: 14px;
16         background-color: lightgray;
17     }
18
19     ▼ .container {
20         width: 100%;
21         padding: 10px;
22     }
23
24     ▼ .nav-menu {
25         display: none; /* Hide navigation menu on small screens */
26     }
27 }
```

Key term

Responsive web design (RWD) – makes your web pages appear correctly (look good) on all types of device, including desktop PCs, mobile and tablet devices.



Uploading of files to a web server

To allow a website to be seen by the public across the internet, it must be uploaded on to a web server (going live). The process of uploading involves a protocol called File Transfer Protocol (FTP). Uploading files via FTP (commonly known as FTPing) can be done directly through a web browser or by using a program such as FileZilla®.

It is not only the web pages that must be uploaded onto the web server, but all the associated files including media assets and CSS files. This is because these files are not embedded into the web pages. Instead, they are linked to them but remain as separate entities.



HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are two of the main tools for designing and displaying websites and web pages. If they are combined with a client-side processing language like JavaScript to add dynamic and interactive content, then websites can be built that are clear, memorable, useful, informative, accessible, organised and maintainable. This section deals with JavaScript.

What is JavaScript?

JavaScript is an object oriented programming language, although it can also be used as an imperative language as well. It is most commonly used to write client-side scripts in web pages. These allow a visitor to a web page to interact with it and to allow dynamic data changes on the web page. Don't confuse JavaScript with Java, by the way! They are two different languages, and used for different things.



What JavaScript do OCR students need to know for the A Level qualifications?

- According to the OCR Specifications, "learners are expected to be able to follow and write basic JavaScript code. It is hoped they will get practical experience of JavaScript in their study of the course.
- They will not be expected to commit exact details of syntax to memory.
- Questions in the exam will not penalise learners for minor inaccuracies in syntax.
- Learners will be expected to be familiar with the JavaScript equivalents of the structures listed in the pseudocode section (with the exception of input and output.
- They will not be expected to use JavaScript for Object Oriented programming or file handling.
- Questions will not be asked in JavaScript where something is passed to a subroutine by value or reference is relevant."



Example

Input

Input will be taken in by reading values from a form. NB learners will not be expected to memorise the method for doing this as focus will be on what they do with that input once it is received.

Output

By changing the contents of an HTML element

```
chosenElement = document.getElementById("example");  
chosenElement.innerHTML = "Hello World";
```

By writing directly to the document

```
document.write("Hello World");
```

By using an alert box

```
alert("Hello World");
```

Any other JavaScript used will be explained in the question.

