Welcome to this

CoGrammar

Task Walkthrough: Task 19 and 20

The session will start shortly...

Questions? Drop them in the chat.

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Task Walkthrough: HTML and CSS

Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** throughout this session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [Questions]

Software Engineering Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Learning Outcomes

- Understand the basic structure and purpose of HTML.
- Create simple HTML documents using appropriate tags.
- Use common CSS properties to style elements on your web pages.
- Transfer learnings to complete the HTML and CSS tasks.

Task Walkthrough: HTML

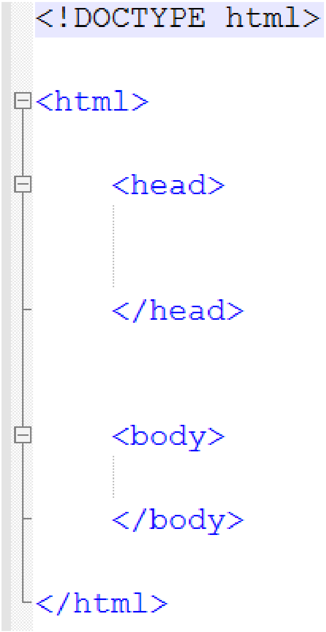
CoGrammar

What is HTML?

- HTML stands for HyperText Markup Language.
- It is a language that we use to create files that tell the browser how to lay out or structure text, images, tables, “content” etc. on a web page.

HTML Document Layout

- A DOCTYPE which indicates which version of HTML to load.
- A head which contains metadata about the page.
- A body which contains the actual content.
- Both the head and body are nested inside the html element.



```
<!DOCTYPE html>

<html>

        <head>


        </head>


        <body>

        </body>

</html>
```

HTML Tags vs Attributes

- **Tags** are the fundamental building blocks of HTML and are used to define the structure and content of a webpage.
- **Attributes** provide additional information about the objects created by HTML elements and modify their behaviour or appearance such as size, colour, alignment, links, and more.

Importance of HTML Structure

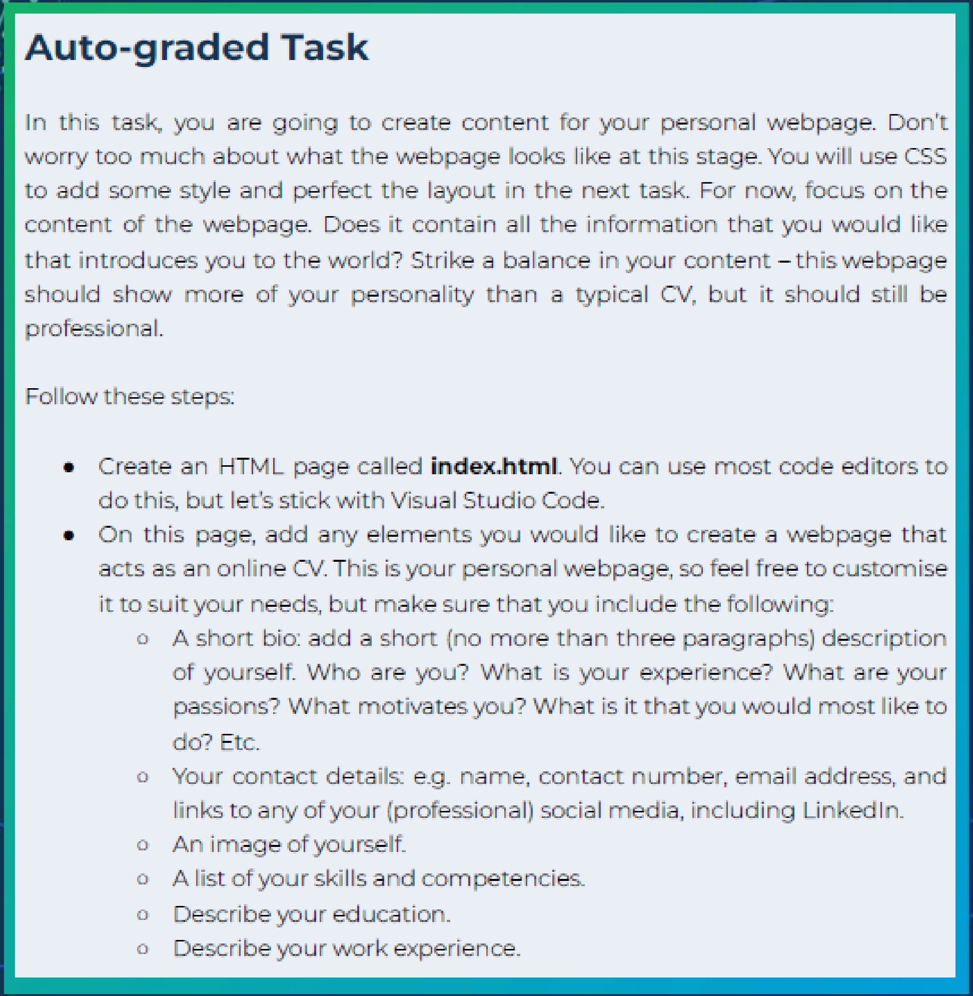
- With understanding the different components of and HTML page, let’s consider the importance of proper structure.
- Proper document structure is fundamental to creating well-organised, accessible, and maintainable web content.
- Accessibility
- Search Engine Optimisation (SEO)

Importance of HTML Structure ...

- Consistency and Maintainability
- Responsive Design
- Performance Optimisation
- Future Scalability

Separation of Concerns (SoC)

- HTML plays a crucial role in separating content from presentation in web development.
- This concept, known as "separation of concerns," is essential for creating maintainable, scalable, and accessible web content.
- **Content Layer:** HTML serves as the foundation or "content layer" of webpages, housing the actual content of the website, including text, images, videos, links, and other media.

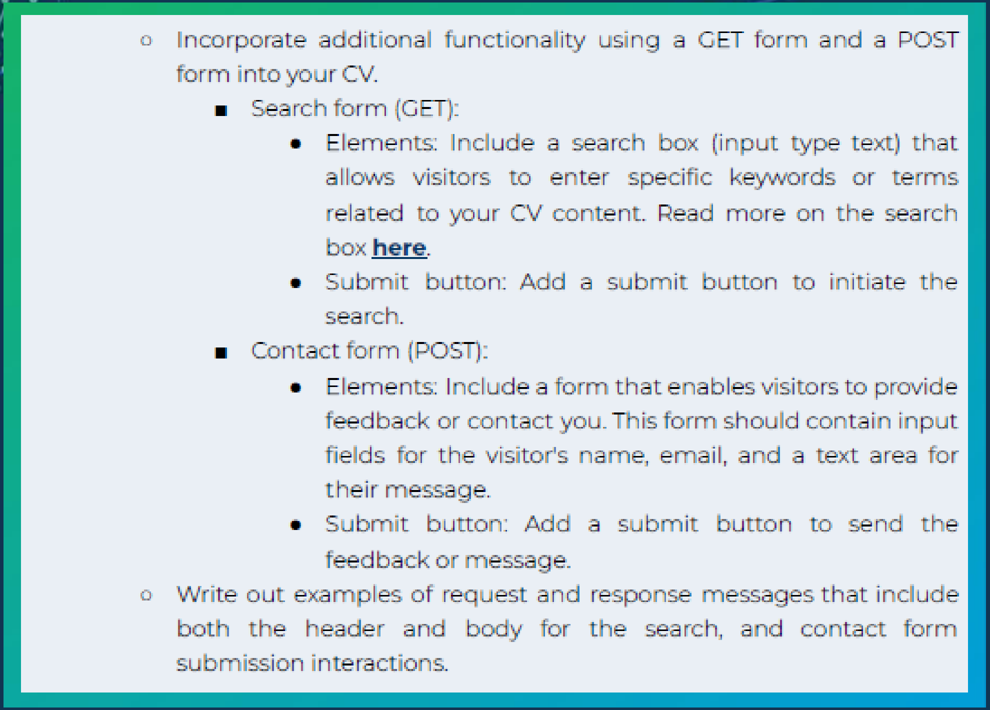


Auto-graded Task

In this task, you are going to create content for your personal webpage. Don't worry too much about what the webpage looks like at this stage. You will use CSS to add some style and perfect the layout in the next task. For now, focus on the content of the webpage. Does it contain all the information that you would like that introduces you to the world? Strike a balance in your content – this webpage should show more of your personality than a typical CV, but it should still be professional.

Follow these steps:

- Create an HTML page called **index.html**. You can use most code editors to do this, but let's stick with Visual Studio Code.
- On this page, add any elements you would like to create a webpage that acts as an online CV. This is your personal webpage, so feel free to customise it to suit your needs, but make sure that you include the following:
  - A short bio: add a short (no more than three paragraphs) description of yourself. Who are you? What is your experience? What are your passions? What motivates you? What is it that you would most like to do? Etc.
  - Your contact details: e.g. name, contact number, email address, and links to any of your (professional) social media, including LinkedIn.
  - An image of yourself.
  - A list of your skills and competencies.
  - Describe your education.
  - Describe your work experience.



- Incorporate additional functionality using a GET form and a POST form into your CV.
    - Search form (GET):
        - Elements: Include a search box (input type text) that allows visitors to enter specific keywords or terms related to your CV content. Read more on the search box **here**.
        - Submit button: Add a submit button to initiate the search.
    - Contact form (POST):
        - Elements: Include a form that enables visitors to provide feedback or contact you. This form should contain input fields for the visitor's name, email, and a text area for their message.
        - Submit button: Add a submit button to send the feedback or message.
- Write out examples of request and response messages that include both the header and body for the search, and contact form submission interactions.

Task Walkthrough: CSS

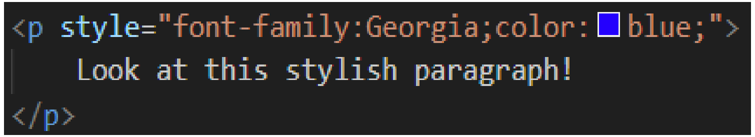
CoGrammar

What is CSS?

- Cascading Style Sheets (CSS) is a language which is used to change the presentation and look of a particular document which has been written in a markup language, such as HTML.
- HTML elements are described using attributes and properties. You can style a web page by changing the properties of the elements that make up that webpage.
- For example: font-family (Arial, Times New Roman, etc.), font-style (normal, italics, etc.) and font-size.

Using Inline Styling

- An example of using the style attribute to change the font of an element is shown below:

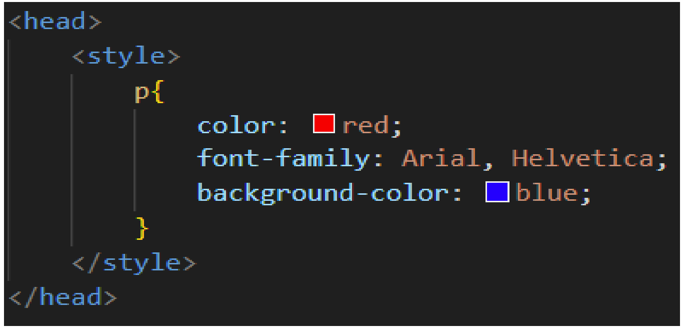


```
<p style="font-family:Georgia;color:blue;">
    Look at this stylish paragraph!
</p>
```

Look at this stylish paragraph!

Using Internal CSS

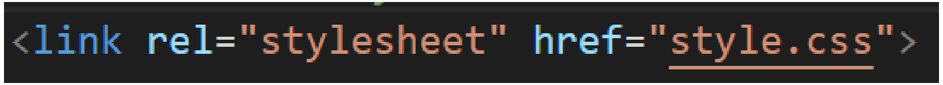
- You can define a CSS rule in the head part of your HTML template -> This is called internal CSS.
- The CSS rule below will format all paragraphs to the colour red with font-family Arial. If the browser can’t find Arial, then it will look for Helvetica. Paragraph backgrounds will be blue.



```
<head>
    <style>
        p{
            color: red;
            font-family: Arial, Helvetica;
            background-color: blue;
        }
    </style>
</head>
```

Using External CSS

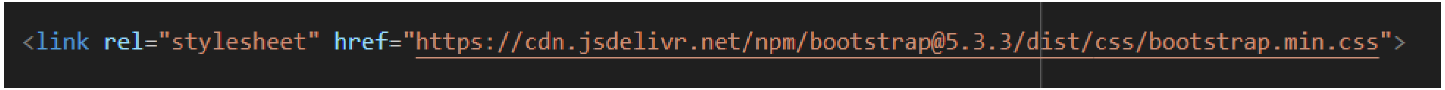
- If your website consists of many HTML files, you likely want to be able to apply the same style rules to all the web pages. To accomplish this, use external CSS instead of internal CSS.
- To link an external CSS file to a specific HTML file, include the below in the <head> section of the HTML file.



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

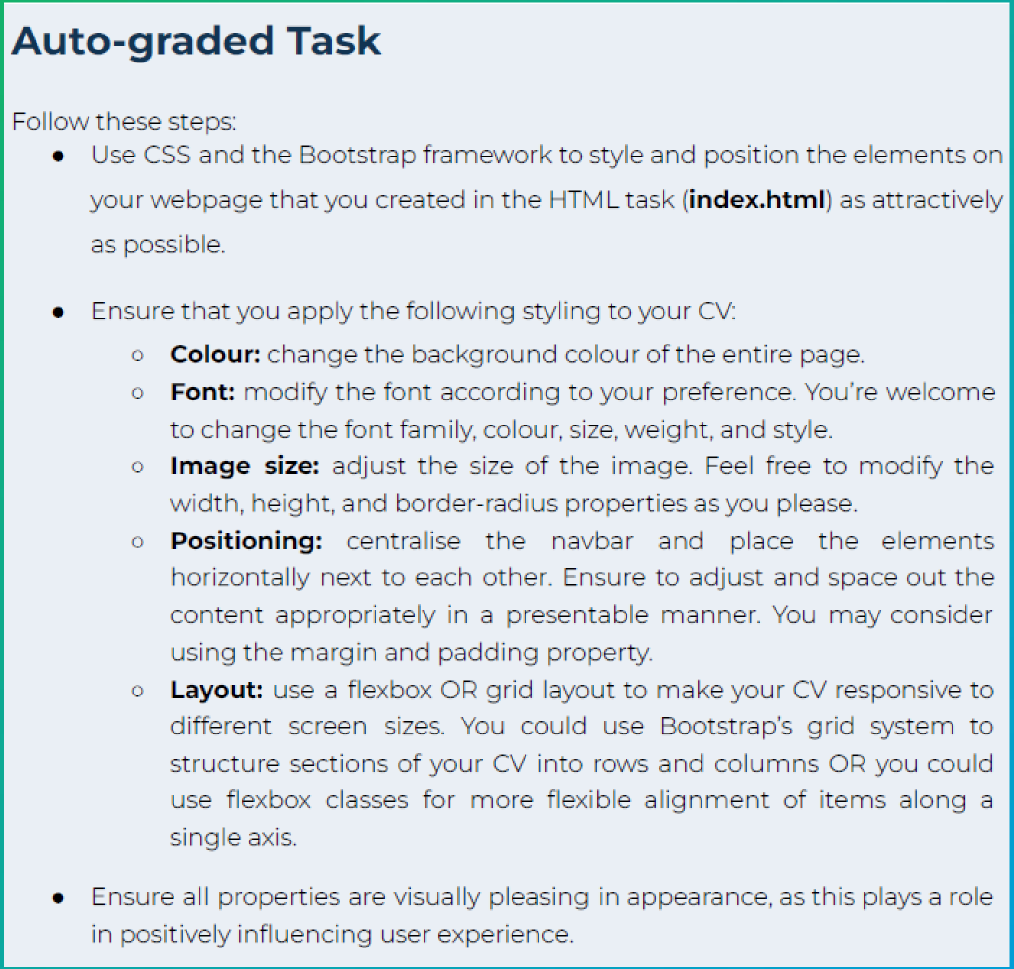
Using Bootstrap

- Bootstrap is an open-source CSS framework.
- It contains predefined templates we can use for styling our web pages.
- We link Bootstrap with our html pages similarly to how we link our own style sheets.



```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
```

- Now that we have access to the style rules we can apply them to our pages like we do with external styling.



Auto-graded Task

Follow these steps:

- Use CSS and the Bootstrap framework to style and position the elements on your webpage that you created in the HTML task (**index.html**) as attractively as possible.

- Ensure that you apply the following styling to your CV:
    - **Colour:** change the background colour of the entire page.
    - **Font:** modify the font according to your preference. You're welcome to change the font family, colour, size, weight, and style.
    - **Image size:** adjust the size of the image. Feel free to modify the width, height, and border-radius properties as you please.
    - **Positioning:** centralise the navbar and place the elements horizontally next to each other. Ensure to adjust and space out the content appropriately in a presentable manner. You may consider using the margin and padding property.
    - **Layout:** use a flexbox OR grid layout to make your CV responsive to different screen sizes. You could use Bootstrap's grid system to structure sections of your CV into rows and columns OR you could use flexbox classes for more flexible alignment of items along a single axis.

- Ensure all properties are visually pleasing in appearance, as this plays a role in positively influencing user experience.

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar