Welcome to the CoGrammar CSS and GitHub Lecture

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

CoGrammar

Skills Bootcamp 8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

CSS and GitHub

March 2024

Lecture Overview

- → Introduction to CSS
- → Styles
- → Selectors
- → The Box Model
- → GitHub

CoGrammar

CSS

Cascading Style Sheets (CSS) is a language used to change the presentation and styling of a document written in a markup language e.g. HTML

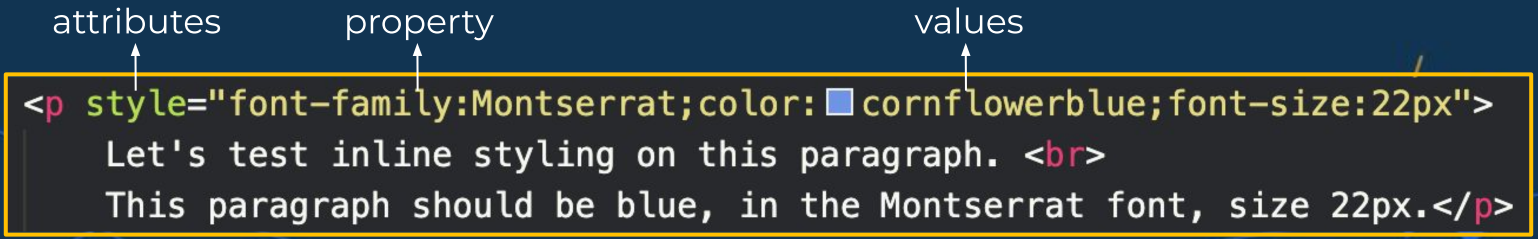
- ❖ Helps us create **visually appealing** and **user-friendly** websites.
- ❖ HTML structures the content, CSS controls how the content looks.
- ❖ CSS uses a **set of rules** written in a **certain syntax** to style HTML.
- ❖ We use CSS to create **style sheets**, which define the appearance and layouts of the elements on a webpage.
- ❖ The various properties which we can control with CSS can be found here.

CoGrammar

Styles: Inline Style

- HTML elements are described using **attributes** and **properties**.
- One of the attributes of an element is **style**, which we can change by **adjusting its properties** using CSS rules.
- Attributes are adjusted **inside the element's beginning tag**.

<u>For example: Text Elements:</u>



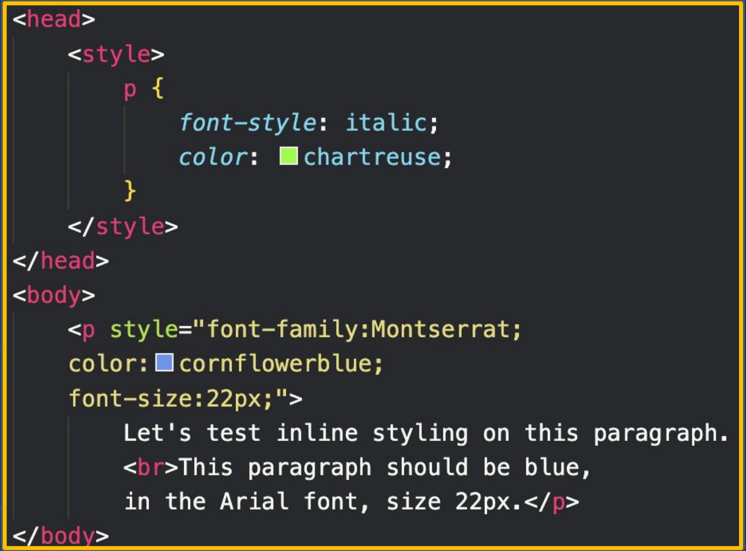
```html
<p style="font-family:Montserrat;color:cornflowerblue;font-size:22px">
    Let's test inline styling on this paragraph. <br>
    This paragraph should be blue, in the Montserrat font, size 22px.</p>
```

CoGrammar

Styles: Internal CSS

❖ CSS rules can be defined in the **head** part of the HTML template, inside the **style element**. This is known as **internal CSS**.

❖ Rules can be defined for every type of element in the HTML document.



```
<head>
    <style>
        p {
            font-style: italic;
            color: chartreuse;
        }
    </style>
</head>
<body>
    <p style="font-family:Montserrat;
    color: cornflowerblue;
    font-size:22px;">
        Let's test inline styling on this paragraph.
        <br>This paragraph should be blue,
        in the Arial font, size 22px.</p>
</body>
```

➔ The style sheet consists of **selectors** and **declarations**

- **Selectors:** indicates which element you want to style
- **Declaration block:** contains one or more declarations, separated by semicolons and enclosed in curly brackets.
- **Declaration:** includes a property and a value separated by a colon

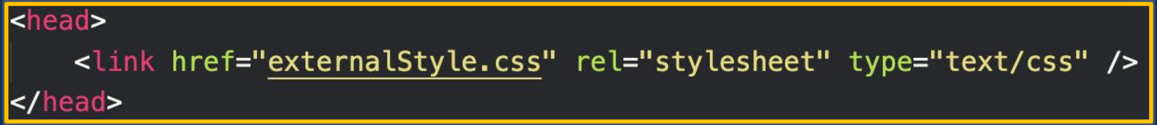
CoGrammar

Styles: External CSS

❖ Another way to define the style for an HTML file is by writing all the style rules in a **separate .css file**. This is called **external CSS**.

❖ The external file can be **linked** to any HTML file to apply the style rules.

❖ This method is useful when **applying the same style rule to multiple HTML files**.



```
<head>
    <link href="externalStyle.css" rel="stylesheet" type="text/css" />
</head>
```

➔ In the **head** part of the HTML file, in a **link element** define

- **href:** define the name and path of your file (relative to the current working directory)
- **rel:** describes the type of relation the external file is to the HTML (i.e. stylesheet)
- **type:** tells the browser what sort of file it is (only necessary for old browsers)

CoGrammar

Best Approaches to Styling

- ❖ Styling is applied depending on which rules are **closest to the element**.
- ❖ Inline styling will be applied to individual elements **overwriting the internal or external CSS** defined for the whole web page.
- ❖ Internal styling will overwrite any external styling defined.
- ❖ **External CSS** should be chosen over internal CSS where possible
  - ➢ **Readability:** separating CSS code and HTML makes code easier to read and follow.
  - ➢ **Maintainability:** updating and debugging styling rules is easier since only external CSS files need to change or be replaced.

CoGrammar

CSS Selectors

CSS selectors attach to the HTML elements on web pages which allows for customized styling

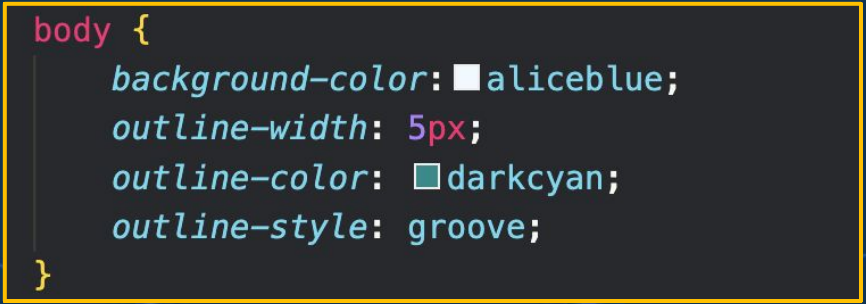
- There are three common CSS selectors that we will look at:
  - **Element selector**
    - The same style is applied to elements with the same tag.
  - **ID selector**
    - Styles are applied to specific elements using a unique ID.
  - **Class selector**
    - The same style is applied to elements in the same class.

CoGrammar

Element Selectors

- ❖ The most basic type of CSS selector.
- ❖ Style rules are defined for all elements of the same type of tag.
- ❖ The selector pinpoints an **element tag** and applies **the same style** to **all elements with that specific tag name**.

<u>For example: Styling the body element</u>

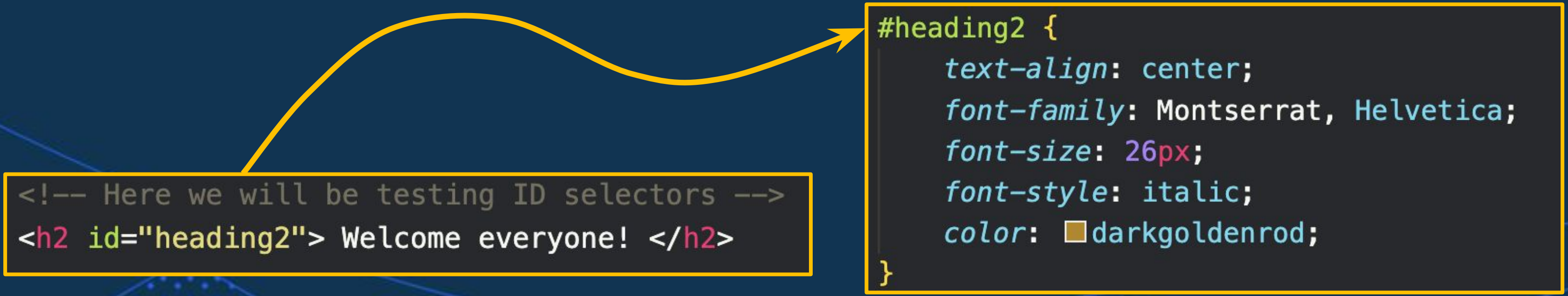


```css
body {
    background-color: aliceblue;
    outline-width: 5px;
    outline-color: darkcyan;
    outline-style: groove;
}
```

CoGrammar

ID Selectors

- ID selectors apply styles to HTML elements which are identified by its **unique ID name**.
- The ID of an element is an **attribute** defined at the beginning of the HTML tag. The value assigned to this attribute must be **unique**.
- The ID selector is called using a **hash (#)**, followed by the **ID name**.



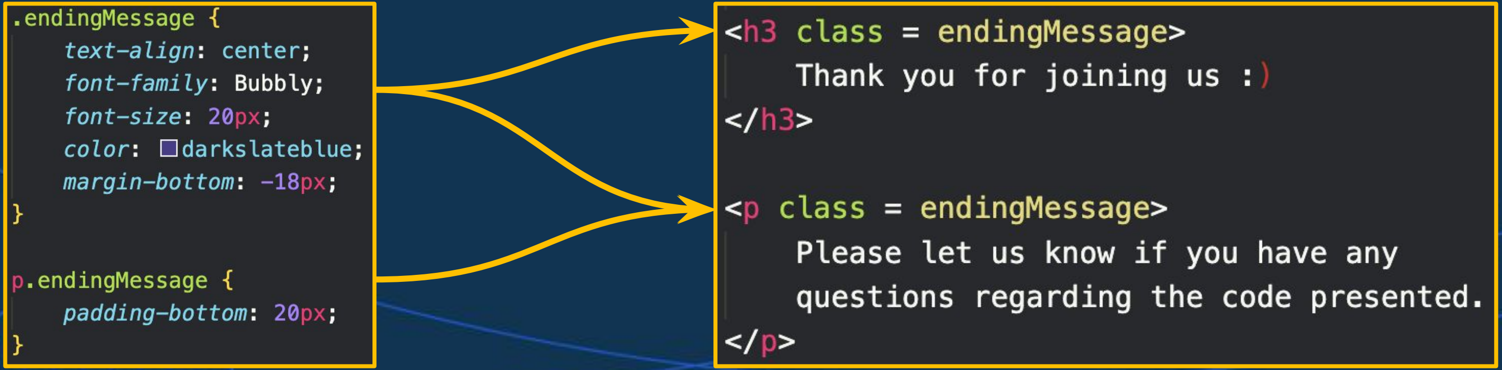
```html
<!-- Here we will be testing ID selectors -->
<h2 id="heading2"> Welcome everyone! </h2>
```

```css
#heading2 {
    text-align: center;
    font-family: Montserrat, Helvetica;
    font-size: 26px;
    font-style: italic;
    color: darkgoldenrod;
}
```

CoGrammar

Class Selectors

- Class selector aims to change **all HTML elements associated with a specific class**.
- **Class** is also an **attribute**, defined like an ID, but it is **not unique**.
- It is called using a **dot (.) followed by the class name**.
- The **element tag** belonging to that class can be referenced as well.



```css
.endingMessage {
    text-align: center;
    font-family: Bubbly;
    font-size: 20px;
    color: ■darkslateblue;
    margin-bottom: -18px;
}

p.endingMessage {
    padding-bottom: 20px;
}
```

```html
<h3 class = endingMessage>
    Thank you for joining us :)
</h3>

<p class = endingMessage>
    Please let us know if you have any
    questions regarding the code presented.
</p>
```

CoGrammar

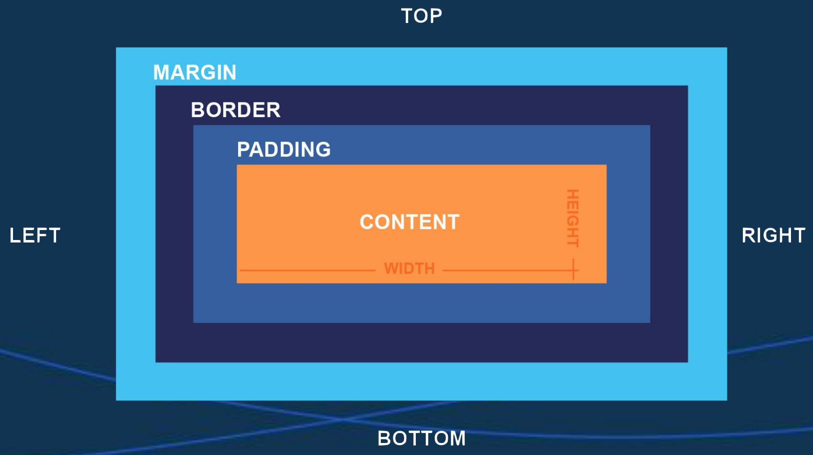
Let's take a break

CoGrammar

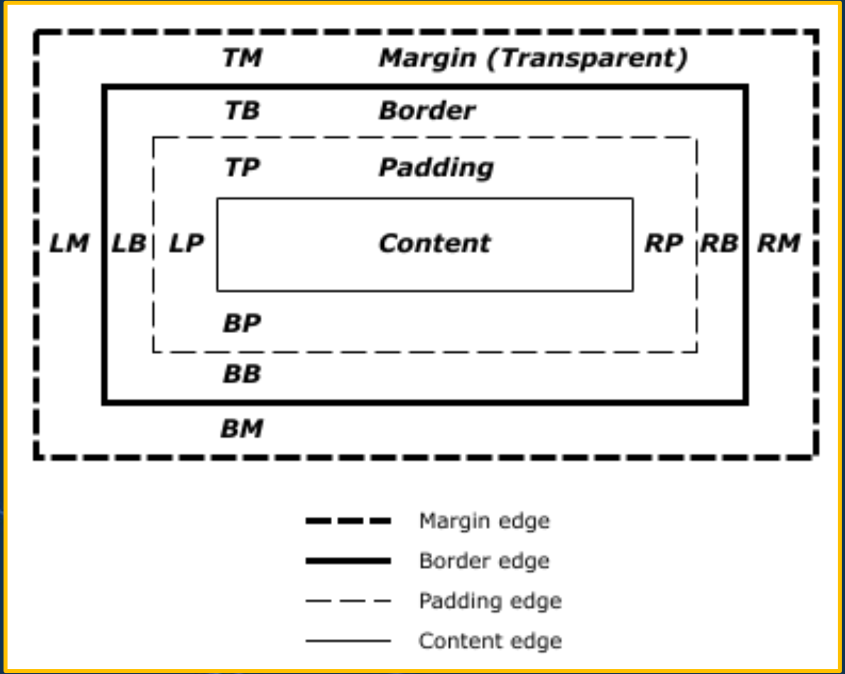
The Box Model

- A **rectangle** is created for each element in the HTML document.
- The **box model** describes how the **padding, border, and margin** are added to the content to create the rectangle.
- Each area is surrounded by a perimeter called an **edge**.

Source: GCFGlobal



CoGrammar



- **Content Edge or Inner Edge**
  - Surrounds the rectangle given by the width and height of the box, depending on the content.
- **Padding Edge**
  - Surrounds the box padding.
  - *padding, padding-top, padding-bottom, padding-left, padding-right*
- **Border Edge**
  - Surrounds the box's border.
  - *border, border-top, border-bottom, border-left, border-right*
- **Margin Edge or Outer Edge**
  - Surrounds the box margin.
  - *margin, margin-top, margin-bottom, margin-left, margin-right*

CoGrammar

CSS Validator

- ❖ An important step in your development journey is **testing** and **debugging** your code.
- ❖ Using tools like VSCode allows us to identify errors in our **syntax** and **formatting**, but some errors may go unnoticed.
- ❖ We can use other tools like this CSS Validation Service, to check our CSS code as well.
- ❖ When our code doesn't behave as expected, or our web pages don't look the way we intended, understanding how to **identify errors** is an important first step before we can **debug**.

CoGrammar

Version Control

- Version control systems **record modifications** to a file or set of files so that you can **recall** specific versions of it later on.
- There are many **benefits** to using version control:
  - **Collaboration:** When working with a team, multiple people can work on the project simultaneously and changes will be merged to a common version and stored in a central place.
  - **Storing versions:** A current version of the project is worked on and is stored locally and all previous versions are stored and managed by the version control system.
  - **Restoring previous versions:** If any changes are made that result in errors or have unintended results, a previous version of the project can easily be restored.

CoGrammar

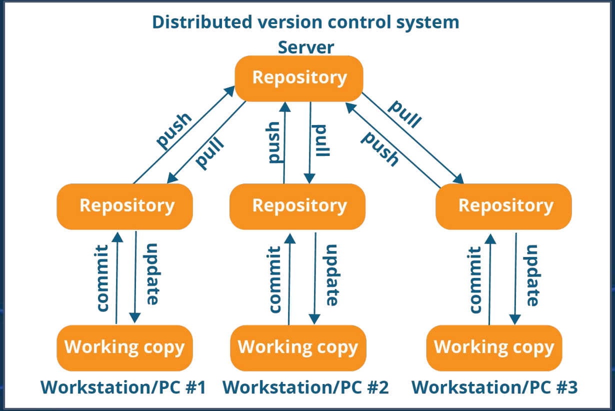
Version Control

- **Benefits** continued:
  - **Understanding what happened:** Whenever changes are made which result in a new version of the project, a short description needs to be provided which describes the changes. This means any changes, why they were made and who made them can be tracked.
  - **Backup:** Every member of the team has a complete version of the project on their disk, which includes the project's complete history. If the central server breaks down or your backup fails, a team members local version can be used instead.

Git Version Control

- Git is the most widely used modern version control system. It is **free**, **open-source**, and **scalable** with project size.
- Git has a **distributed architecture** and is an example of a distributed version control system (DVCS).
- Every developer's working copy of the code is also a repository that contains the **full history of all changes**.

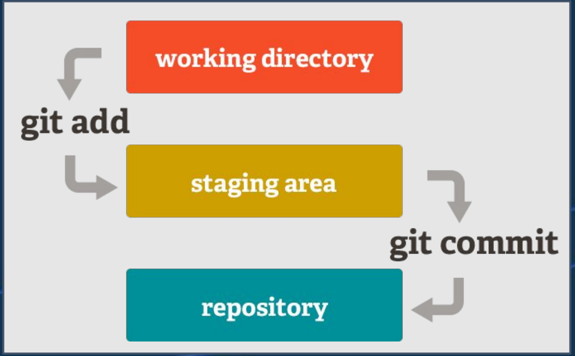
Source: TechJunction



CoGrammar

Git Version Control

- Files can have three main states: **committed**, **modified**, and **staged**.
  - **Committed:** Files are safely stored in your local database.
  - **Modified:** Files have changed but are yet to be committed to your local database.
  - **Staged:** Modified files have been marked to go into the next commit in its current version.
- There are three main sections of a Git project:
  - **Working directory**
  - **Staging area**
  - **Git repository**



Source: Medium

CoGrammar

Installing Git

- Install Git using one of the following links:
  - Windows
  - Mac
  - Linux
- Verify that the installation was successful by typing the following into the terminal:
  - `git --version`
- Configure your Git credentials:
  - `git config --global user.name "Your Name"`
  - `git config --global user.email "youremail@email.com"`

CoGrammar

GitHub

- GitHub is an online Git **repository hosting service**, which provides a **web-based graphical interface**.
- GitHub also offers a service called GitHub Desktop, which grants users access to all Git tools and GitHub services on their **local desktop**.
- GitHub is not just a project-hosting service, it is also a large **social networking site** for developers and programmers.
- This allows users to create a **technical portfolio**, which showcases their work and technical capabilities.

CoGrammar

Repositories

- A repository in Git is a **hidden folder called '.git'**, which is located in the root directory of your project.
- This is where your version control system stores all the files for a particular project.
- A **local repository** is located on your local computer as the '.git' folder inside the project's root folder.
- A **remote repository** is located on a remote server on the internet or in your local network.
- **Local repositories** are created by either **initialising a new repository** or **cloning**

Common Git commands

- Initialise a new repository in the current directory
  - `git init`
- Add a new file to the repository staging area
  - `git add fileName.js`
- Check the status of the files in your working directory
  - `git status`
- Committing staged changes with a meaningful message
  - `git commit -m "Added new file fileName.js"`
- Reviewing the change history
  - `git log`

Let's try adding the files we created in this lecture to a new repo!

CoGrammar

Using remote repositories

- To synchronise a **local repository** to a **remote repository** hosted by GitHub, authenticated access to your GitHub account is needed.
- For this, **GitHub Command Line Interface** needs to be used.
- Once installed, login using the following command and follow the prompts
  - `gh auth login`
  - Choose GitHub.com for the account and HTTPS for the preferred protocol.
- **Sync your local repository to a remote GitHub repository:**
  - `git remote add origin https://github.com/[REPO-OWNER]/[REPO-NAME]`
  - `git branch -M main`
  - `git push -u origin main`

CoGrammar

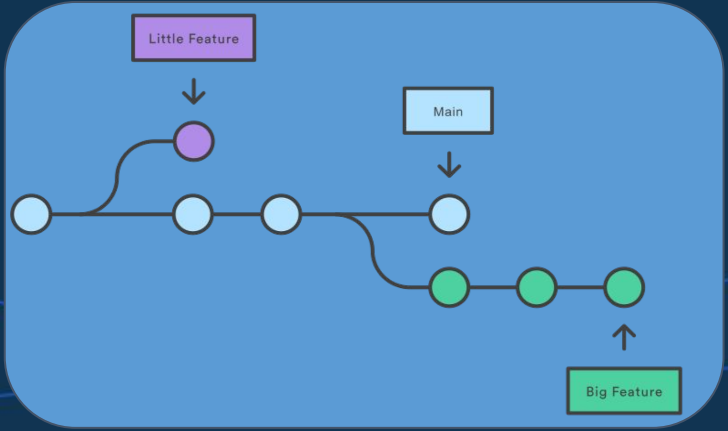
Using remote repositories

- To clone a repository hosted on GitHub:
  - `git clone https://github.com/[REPO-OWNER]/[REPO-NAME]`
- To fetch the newest version of the repository:
  - `git fetch origin`

CoGrammar

Branches

- A branch represents an **independent** line of development.
- It allows each developer to **branch out** from the original codebase and **isolate** their work from others.
- Developers can continue to work without messing up or disrupting the main line.
- This ensures that **unstable code** is not committed to the main codebase.



CoGrammar

Branches

- **Creating a branch:**
  - `git branch branch-name`
- **Switching branches:**
  - `git checkout branch-name`
  - `git checkout -b new-branch-name`
- **Merging branches:**
  - `git merge branch-name`

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

CoGrammar