Welcome to CoGrammar

WD Week 5 Tutorial

The session will start shortly...

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

CoGrammar

Full Stack Web Development 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** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- 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**

Full Stack Web Development 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**

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

DOM Manipulation and Event Handling

April 2024

Lecture Overview

- Recap of DOM Manipulation
- Recap of Event Handling

CoGrammar

Moving through the madness

- DOM **traversal** involves navigating through the DOM tree to access or manipulate elements.

```
// Retrieve the element with the ID "myDiv"
var element = document.getElementById("myDiv");

// Retrieve all elements with the class "container"
var containers = document.getElementsByClassName("container");

// Retrieve all list item elements
var listItems = document.getElementsByTagName("li");

// Retrieve the first paragraph element within a container
var paragraph = document.querySelector(".container p");

// Retrieve all paragraph elements within a container
var paragraphs = document.querySelectorAll(".container p");
```

CoGrammar

It’s Morphin’ Time

- Adding elements:

```
// Create a new paragraph element
let paragraph = document.createElement("p");
let heading = document.createElement("h1");

// Add text content to the paragraph
paragraph.textContent = "This is a new paragraph.";
heading.textContent = "I love pie";

// Append the paragraph to the body element
document.body.appendChild(paragraph);
document.body.insertBefore(heading, paragraph);
```

CoGrammar

It’s Morphin’ Time

- Modifying Elements:

```
// Change inner HTML content of an element
document.getElementById("myElement").innerHTML = "<strong>New content</st

// Set text content of an element
document.getElementById("myElement").textContent = "Updated text content"

// Set attribute value of an element
document.getElementById("myElement").setAttribute("class", "new-class");
```

CoGrammar

It’s Morphin’ Time

- Removing Elements:

```
// Get the element to remove
let elementToRemove = document.getElementById("toRemove");

// Remove the element from the DOM
elementToRemove.remove();
```

CoGrammar

The event Object

- In JavaScript, when an event occurs, an event object is automatically created by the browser.
- The event object contains information about the event that occurred, such as its type, target element, and additional event-specific data.

```
document.addEventListener("click", function (event) {
  console.log("Event Type:", event.type);
  console.log("Target Element:", event.target);
  console.log("Mouse Coordinates:", event.clientX, event.clientY);
});
```

CoGrammar

Handling UI Events

- One common approach to handle UI events is by using the **addEventListener** method. This method allows developers to attach event listeners to DOM elements and specify callback functions to be executed when the events occur.

```
document.getElementById("myButton").addEventListener("click", function ()
  alert("Button clicked!");
});
```

CoGrammar

Handling UI Events

- Click Event: Initiates an action when a user clicks on a button, link, or any interactive element.

```
document.getElementById("myButton").addEventListener("click", function ()
  alert("Button clicked!");
});
```

Handling UI Events

- Keydown Event: Captures keystrokes, allowing for keyboard-driven interactions within the application.

```
document.addEventListener("keydown", function (event) {
  console.log("Key pressed:", event.key);
});
```

CoGrammar

Handling UI Events

- Mouseover Event: Provides feedback when the mouse cursor enters a specific area or element.

```
let element = document.getElementById("myElement");

element.addEventListener("mouseover", function () {
  console.log("Mouse over element!");
});
```

CoGrammar

Handling UI Events

- Mouseout Event: Triggers actions when the mouse cursor leaves a designated area or element.

```
let fetchedElement = document.getElementById("myElement");
fetchedElement.addEventListener("mouseout", function () {
  console.log("Mouse out of element!");
});
```

CoGrammar

Handling System Events

- System events are events triggered by changes in the system or browser environment, rather than direct user interactions. These events provide valuable information about the application's state or the browser's behavior, allowing developers to respond accordingly.

```
window.addEventListener("resize", function () {
  console.log("Window resized");
});
```

CoGrammar

Handling System Events

- Resize Event: Adjusts layout or UI elements dynamically in response to changes in the browser window size.

```
window.addEventListener("resize", function () {
  console.log("Window resized");
});
```

CoGrammar

Handling System Events

- Load Event: Executes scripts or initializes components once the entire page and its dependencies have been loaded.

```
window.addEventListener("load", function () {
  console.log("Page loaded!");
});
```

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar