CoGrammar

Welcome to this session:

Task Walkthrough - Higher-order Functions and Callbacks

The session will start shortly...

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

Safeguarding & Welfare

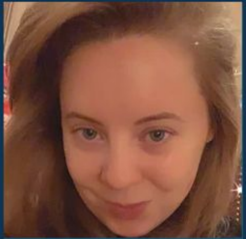
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

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

Rafiq Manan



Charlotte Witcher

Nurhaan Snyman

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

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

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- 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**

Skills Bootcamp Cloud Web Development

- 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

- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- ❖ **Create a custom higher-order function** that filters data based on specific criteria.
- ❖ **Define callbacks** and how to use them within higher-order functions.
- ❖ **Use setInterval() and clearInterval()** to control the timing of function execution in JavaScript.
- ❖ **Combine multiple functions** effectively to create dynamic, interactive applications.

Lecture Overview

- → Presentation of the Task
- → Higher Order Functions
- → Callbacks
- → Task Walkthrough

CoGrammar

Higher-Order Functions Task

Imagine you're creating a personalised recommendation tool for an online bookstore! 📚 Your goal is to build a custom higher-order function called myRecommender that takes in an array of book genres and a callback function to select only the genres your friend prefers.

By writing a custom filter, you'll have full control over how data is selected, giving users a personalised experience. This task is a chance to put your creativity to work while practising higher-order functions.

❖ Write a higher-order function myRecommender() that accepts an array of genres and a callback function

❖ Write a function that checks if a genre is in the list of favorites and returns true if so, false otherwise.

Callbacks Task

To encourage regular study breaks, you'll create a timed reminder that prompts users to take a short break every hour. You'll use setInterval() to schedule reminders, and clearInterval() to stop the reminders when they're no longer needed.

This task lets you put your timed event skills to practical use, perfect for managing regular notifications, alerts, or updates. ⏰

- Create two buttons: a Start Reminders button to begin the hourly alerts and a Stop Reminders button to turn them off.
- Use setInterval() in your JavaScript file to send a reminder message.
- Use clearInterval() to stop reminders when the Stop Reminders button is clicked.

What is a higher-order function in JavaScript?

A. A function that returns a string.
B. A function that accepts another function as an argument or returns a function.
C. A function that loops through an array.
D. A function that outputs to the console.

CoGrammar

What does setInterval() do in JavaScript?

A. Runs a function once after a delay.
B. Executes a function repeatedly at set intervals.
C. Logs data to the console.
D. Stops a function from executing.

CoGrammar

Arrow Functions

Shorthand syntax for writing function expressions.

- We use an arrow (**=>**) to define these function shorthands.
- They are specifically used when the **function block** is **one line of code**.
- Arrow functions can improve the **readability** and **organisation** of code.

Higher Order Functions

Higher order functions are functions that can accept other functions as arguments or return functions as results.

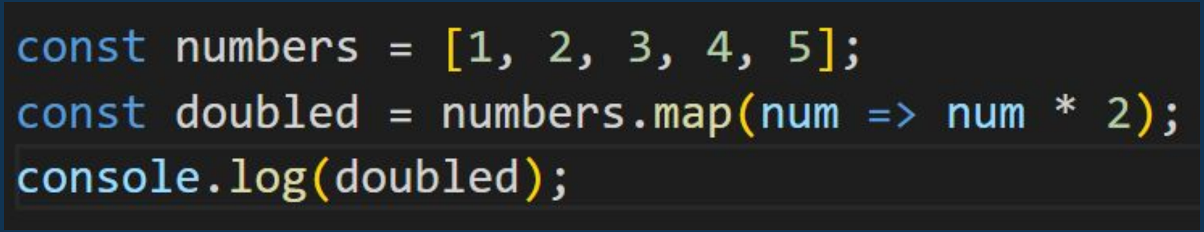
- They enable **abstraction** and code **reusability**, crucial principles in functional programming.
- Some notable examples include **map**(), **filter**(), and **reduce**() in JavaScript.

CoGrammar

Higher Order Functions

Higher order functions are functions that can accept other functions as arguments or return functions as results.

- The **map()** function **applies** a provided function to each element of an array and returns a new array with the results.



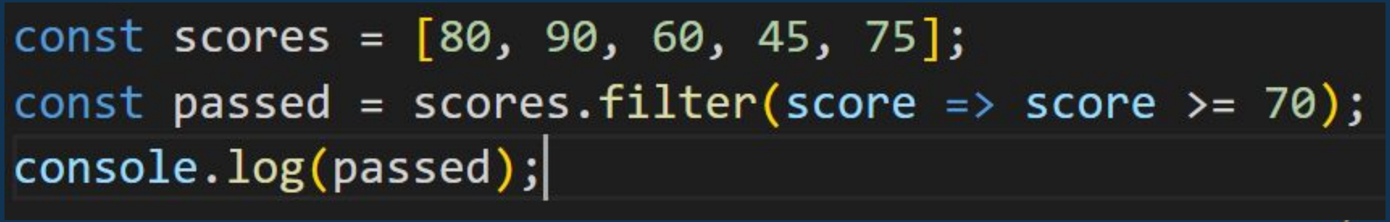
```
const numbers = [1, 2, 3, 4, 5];
const doubled = numbers.map(num => num * 2);
console.log(doubled);
```

CoGrammar

Higher Order Functions

Higher order functions are functions that can accept other functions as arguments or return functions as results.

❖ The **filter()** function creates a new array with all elements that pass the test implemented by the provided function.



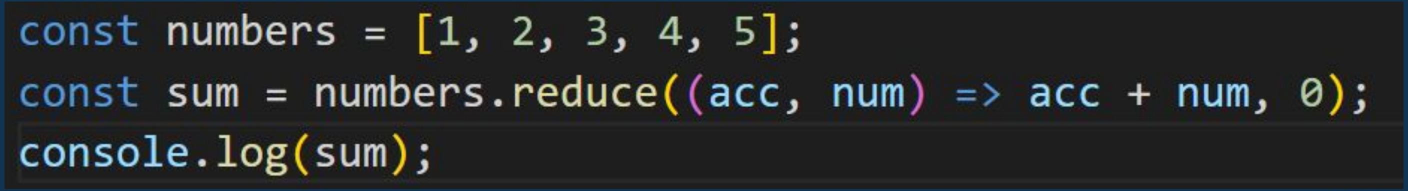
```
const scores = [80, 90, 60, 45, 75];
const passed = scores.filter(score => score >= 70);
console.log(passed);
```

CoGrammar

Higher Order Functions

Higher order functions are functions that can accept other functions as arguments or return functions as results.

- The **reduce()** function executes a **reducer** function on each element of the array, resulting in a single output value.



```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((acc, num) => acc + num, 0);
console.log(sum);
```

CoGrammar

Callback Functions

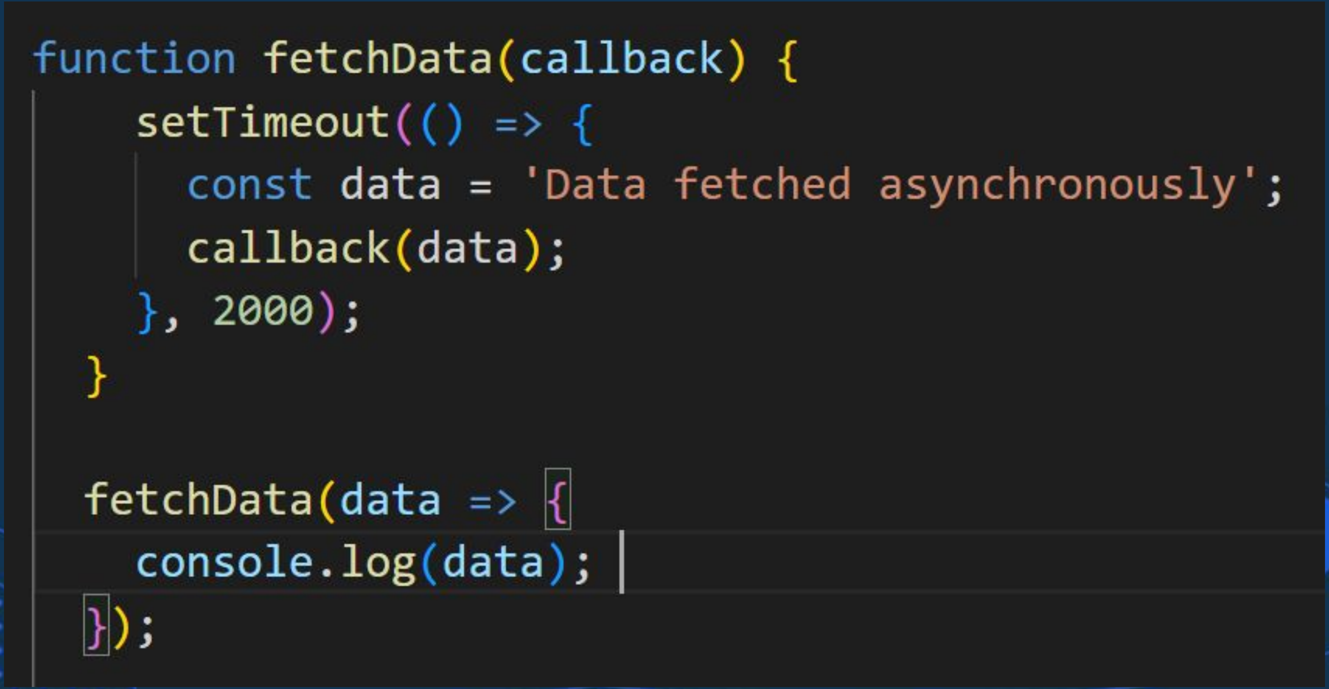
Callback functions are functions passed as arguments to other functions and executed later.

- They are commonly used in **asynchronous programming** and **event handling**.
- Callbacks are vital in handling asynchronous tasks, such as fetching data from an API.
- Callbacks play a crucial role in **event-driven programming**, responding to user interactions.

CoGrammar

Callback Functions

Callback functions are functions passed as arguments to other functions and executed later.



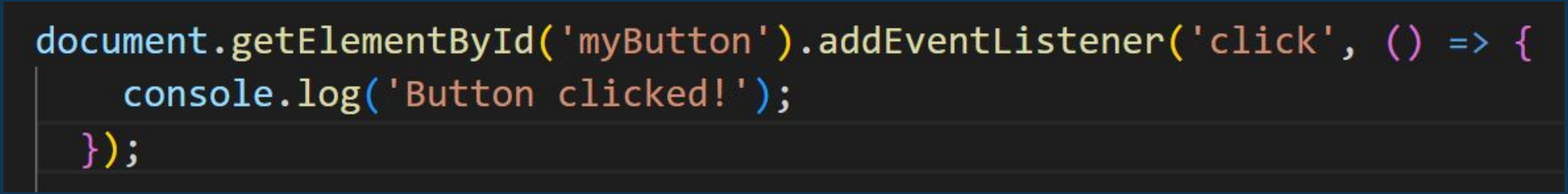
```
function fetchData(callback) {
    setTimeout(() => {
      const data = 'Data fetched asynchronously';
      callback(data);
    }, 2000);
  }

  fetchData(data => {
    console.log(data);
  });
```

CoGrammar

Callback Functions

Callback functions are functions passed as arguments to other functions and executed later.



```
document.getElementById('myButton').addEventListener('click', () => {
    console.log('Button clicked!');
});
```

CoGrammar

Higher-Order Functions Task

Imagine you're creating a personalised recommendation tool for an online bookstore! 📚 Your goal is to build a custom higher-order function called myRecommender that takes in an array of book genres and a callback function to select only the genres your friend prefers.

By writing a custom filter, you'll have full control over how data is selected, giving users a personalised experience. This task is a chance to put your creativity to work while practising higher-order functions.

- ❖ Write a higher-order function myRecommender() that accepts an array of genres and a callback function
- ❖ Write a function that checks if a genre is in the list of favorites and returns true if so, false otherwise.

Callbacks Task

To encourage regular study breaks, you'll create a timed reminder that prompts users to take a short break every hour. You'll use setInterval() to schedule reminders, and clearInterval() to stop the reminders when they're no longer needed.

This task lets you put your timed event skills to practical use, perfect for managing regular notifications, alerts, or updates. ⏰

- ❖ Create two buttons: a Start Reminders button to begin the hourly alerts and a Stop Reminders button to turn them off.
- ❖ Use setInterval() in your JavaScript file to send a reminder message.
- ❖ Use clearInterval() to stop reminders when the Stop Reminders button is clicked.

Which of the following stops an interval created by setInterval()?

A. clearInterval()
B. stopInterval()
C. pauseInterval()
D. endInterval()

In a higher-order function, what is a callback?

A. A function that only runs once.
B. A function passed as an argument to another function.
C. A function that logs output to the console.
D. A variable within a function.

CoGrammar

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education