Welcome to the CoGrammar Lecture: Exception Handling

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

Exception Handling

March 2024

Lecture Overview

- Try
- Catch
- Finally
- Resource Management
- Custom Exceptions

CoGrammar

What are Exceptions in JavaScript?

- **Exceptions** are unexpected or exceptional **events** that occur during the **execution** of a program.
- In JavaScript, exceptions can be thrown **explicitly** using the **throw statement** or can occur **implicitly** due to runtime **errors**.
- Exception handling allows programmers to gracefully **manage** errors and **prevent** program crashes.

CoGrammar

The throw Statement

- The **throw** statement in JavaScript is used to explicitly **raise** an exception.
- It **interrupts** the normal flow of the program and sends control to the nearest enclosing **try** block's **catch** clause.

```
throw new Error("We can thow errors on our own too yeeeeh.");
```

CoGrammar

The try-catch Statement

- The **try-catch** statement is used to **handle** exceptions in JavaScript.
- Code that **may** potentially throw an exception is placed inside the **try block**.
- If an exception occurs within the **try block**, it's caught by the **catch block**, allowing for error **handling**.

```
try {
    // Code that may throw an exception
    console.log(`sdfdsf${dfds}`);
} catch (error) {
    // Handle the error
    console.log("Error caught:", error.message);
}
```

CoGrammar

The finally Block

- The **finally** block is **optional** and is used to **execute** code regardless of **whether** an exception is **thrown** or **caught**.
- It's typically **used** for cleanup tasks, such as releasing resources or closing connections.

```
try {
    // Code that may throw an exception
    console.log(`Name: ${personName}`);
} catch (error) {
    // Handle the error
    console.log("Error caught:", error.message);
} finally {
    console.log("Byeeeee!");
}
```

CoGrammar

Resource Management

- **Exception handling** is crucial for managing resources efficiently in JavaScript applications.
- Resources such as **file** handles, **network** connections, or **database** connections should be properly released to **prevent** resource leaks.
- Using the **finally** block ensures that resources are **released** even if an exception occurs.

Propagating Exceptions

- Exceptions can be **propagated** up the call stack if they're not caught locally.
- This allows for **centralized** error handling at higher levels of the application.
- Uncaught exceptions in JavaScript can lead to program **termination**, so it's essential to handle or propagate them appropriately.

Propagating Exceptions

```
function foo() {
    throw new Error("An error occurred in foo.");
}

function bar() {
    foo();
}

try {
    bar();
} catch (error) {
    console.log("Error caught:", error);
}
```

CoGrammar

Let’s Breathe!

Let’s take a small break before moving on to the next topic.

CoGrammar

Handling Different Types of Errors

- JavaScript supports different types of errors, such as **SyntaxError**, **ReferenceError**, **TypeError**, and custom errors.
- Each **type** of error provides specific information about the nature of the problem, aiding in **debugging** and **resolution**.

```javascript
try {
    // Code that may throw an error
} catch (error) {
    if (error instanceof SyntaxError) {
        console.log("Syntax error:", error.message);
    } else if (error instanceof ReferenceError) {
        console.log("Reference error:", error.message);
    } else {
        console.log("Other error:", error.message);
    }
}
```

CoGrammar

Custom Exceptions

- JavaScript allows developers to define **custom** exception types for specific use cases.
- **Custom exceptions** can provide additional **context** and **semantics** to error handling.
- They're created by **extending** the built-in Error object.

CoGrammar

Custom Exceptions

```
class MyCustomError extends Error {
    constructor(message) {
        super(message);
        this.name = "MyCustomError";
    }
}

try {
    throw new MyCustomError("An error occurred.");
} catch (error) {
    console.log("Error caught:", error.name, "-", error.message);
}
```

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar