CoGrammar

Welcome to this session:
Functions, Scope and Closure

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

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

Skills Bootcamp Full Stack 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. We will 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.***
- Find all the lecture **content** in your **Lecture Backpack** on GitHub.
- If you are hearing impaired, kindly use your computer's function through Google chrome to enable captions.

Skills Bootcamp Progression Overview

✅ Criterion 1 - Initial Requirements

Specific achievements within the first two weeks of the program.

To meet this criterion, students need to, by no later than **01 December 2024 (C11)** or **22 December 2024 (C12):**

- **Guided Learning Hours** (GLH)**:** Attend a minimum of 7-8 GLH per week (lectures, workshops, or mentor calls) for a total minimum of **15 GLH**.
- **Task Completion:** Successfully complete the **first 4 of the assigned tasks**.

✅ Criterion 2 - Mid-Course Progress

Progress through the successful completion of tasks within the first half of the program.

To meet this criterion, students should, by no later than **12 January 2025 (C11)** or **02 February 2025 (C12):**

- **Guided Learning Hours** (GLH)**:** Complete at least **60 GLH**.
- **Task Completion :** Successfully complete the **first 13 of the assigned tasks**.

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3 – End-Course Progress

Showcasing students' progress nearing the completion of the course.

To meet this criterion, students should:

- **Guided Learning Hours** (GLH)**:** Complete the **total minimum required GLH,** by the **support end date.**
- **Task Completion : Complete all mandatory tasks**, including any necessary resubmissions, by the end of the bootcamp, **09 March 2025 (C11)** or **30 March 2025 (C12).**

✅ Criterion 4 - Employability

Demonstrating progress to find employment.

To meet this criterion, students should:

- **Record an Interview Invite:** Students are required to record proof of invitation to an interview by **30 March 2025 (C11)** or **04 May 2025 (C12)**.
  - **South Holland Students** are required to proof and interview by **17 March 2025.**
- **Record a Final Job Outcome :** Within 12 weeks post-graduation, students are required to record a job outcome.

CoGrammar

Learning Outcomes

- Apply JavaScript functions to perform tasks such as summing array elements.
- Explore the effects of different scopes on variables within functions.
- Use closures to encapsulate functionality and maintain state across function calls.

Stay Safe Series:

Mastering Online Safety One week at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Pause Before You Post:

Managing Your Digital Presence

- Impact on Reputation.
- Permanent Record.
- Privacy Concerns.
- Miscommunication.
- Influence on Others.
- Professional Implications.
- Mental Well-being.

Lecture Overview

- Functions
- Scope
- Nested Functions

What is the purpose of a function in JavaScript?

A. To store multiple values.
B. To group code that can be reused and executed when called.
C. To create a loop.
D. To declare a variable.

How do you pass data into a function in JavaScript?

A. Using loops.
B. Using parameters.
C. Using arrays.
D. Using if statements.

Functions

A block of organised, reusable code that accomplishes a specific task.

- A function can be **called repeatedly** throughout your code.
- Functions can either be **user-defined** or **built-in**.
- This helps us **minimise repeating lines of code** unnecessarily.
- The main benefits of using functions are:
  - It improves code **modularity, management** and **maintenance**.
  - It makes our code more **readable**.
  - It **reduces potential errors**.

CoGrammar

BASIC JAVASCRIPT FUNCTIONS

- Declaring a function in JavaScript involves using the keyword **function**, providing a **function name**, followed by a list of **parameters** enclosed in **parentheses ()**, and the **function body** enclosed within curly braces **{}**.
- Basic syntax of a function:

```
function functionName(parameter1, parameter2, ...parameterN) {
  // function body
  // statements defining what the function does
}
```

BASIC JAVASCRIPT FUNCTIONS

- A JavaScript function has three key components:
  - **Parameters** - These are variables listed as a part of the function definition. They act as placeholders for the values on which the function operates, known as arguments.
  - **Function body** - Enclosed between curly braces {}, the function body consists of statements that define what the function does.

BASIC JAVASCRIPT FUNCTIONS

- **Return statement** - How a function sends the result of its operations back to the caller. Not all functions have to return a value; those that don't are often used for their side effects, such as modifying the global state or producing an output.

BASIC JAVASCRIPT FUNCTIONS

- Example of a function that doesn't return anything:

```
function sayHi() {
  console.log("Hi");
}
```

- Example of a function that returns something:

```
function sayHi() {
  return "Hi";
}
```

CALLING A FUNCTION

- After a function has been declared, it can be invoked or called anywhere in your code by using its name followed by parentheses ().
- If the function requires parameters, you'll include **arguments** within the parentheses.
- Each argument corresponds to the position of the parameter in the function declaration.

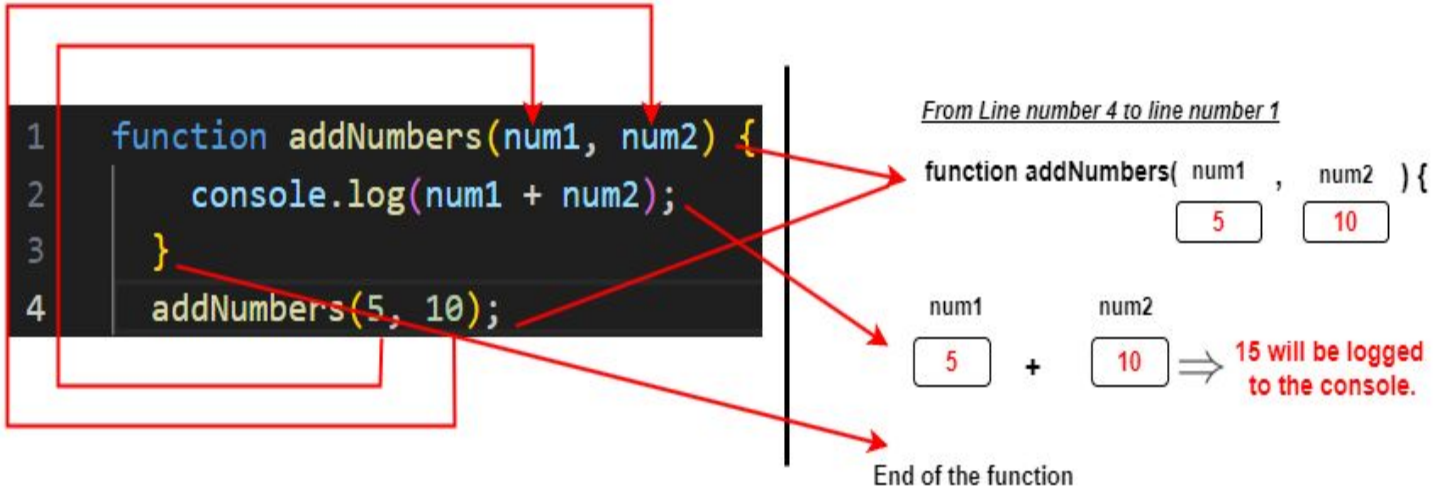
CALLING A FUNCTION

- Example of calling a function:

```
function addNumbers(num1, num2) {

   console.log(num1 + num2); // Log the sum of num1 and num2 to the
console.

}

addNumbers(5, 10); // Calling the addNumbers function with five and ten
as arguments
```

CALLING A FUNCTION

- Let's trace through this function:



CALLING A FUNCTION

- The primary difference between parameters and arguments:
  - **Parameters** - Parameters are used when defining a function. They represent the **'input'** the function needs to do its job, and they act as placeholders for actual data.
  - **Arguments** - Arguments are used when calling a function. They represent the actual **'input'** that will be operated on by the function's code.

Let's take a break

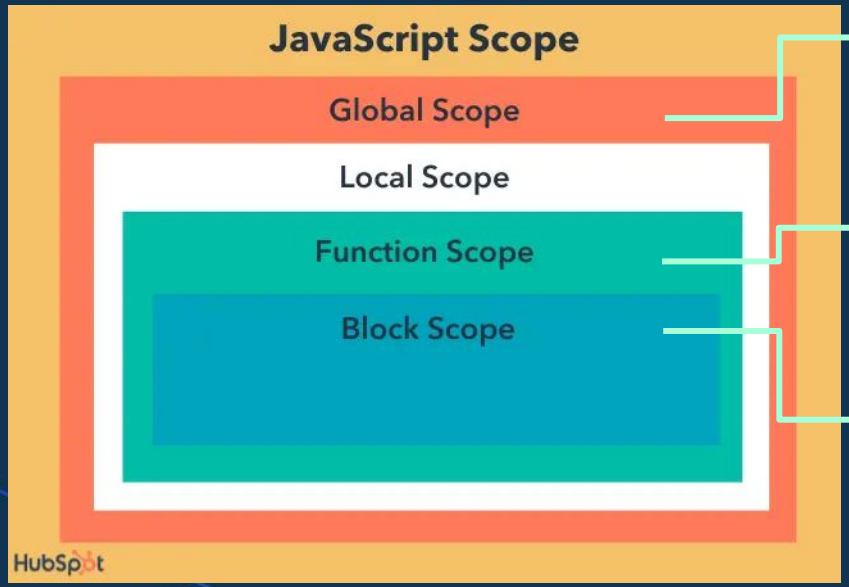
CoGrammar

Scope

The area of visibility and accessibility of a variable in a program.

- The **scope** of a variable determines **where in the code it can be seen**.
- JavaScript has **function scope**, meaning variables declared **inside a function** are only **accessible within** that function.
- Variables declared outside of a function, known as **global variables**, can be accessed anywhere (**hoisting** allows for variables to be accessed before their definition).
- JavaScript has **three types of scope:**
  - Global Scope
  - Function Scope
  - Block Scope

Scope



Source: HubSpot

Global Scope: variables declared outside all functions or blocks. They can be accessed from any part of the code.

Function Scope: variables declared within a function. They are only accessed within their function body.

Block Scope: variables declared with the **let** or **const** keyword inside a block. They can only be accessed in their block (does not apply to **var** keyword).

Nested Functions

A function that is defined inside another function.

- The **nested function** is referred to as the **inner function** and the **containing function** is known as the **outer function**.
- Nested functions can only be called **within the containing function.**
- A nested function forms a **closure**, the function has its **own local variables and parameters** and is able to reference and use its containing **function's function variables and parameters**.

```
function outerFunction(outerParam) {
    let outerFunctionVar;
    function innerFunction(innerParam) {
        console.log(outerParam);
        outerFunctionVar = "initialise";
        return innerParam;
    }
    return innerFunction;
}
```

CoGrammar

What is a closure in JavaScript?

A. A function inside another function that retains access to the outer function's variables.
B. A loop that runs indefinitely.
C. A way to store multiple strings.
D. A way to declare variables globally.

What is the purpose of scope in JavaScript?

A. To declare multiple variables at once.
B. To control where variables are accessible in the code.
C. To create functions.
D. To repeat a block of code.

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