CoGrammar

Welcome to this session:

JavaScript Data Types and Conditional Statements

The session will start shortly...

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

Safeguarding & Welfare

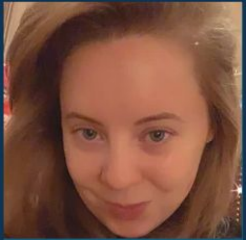
We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

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.

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.

Keep it Secret, Keep it Safe: Why Passwords Should Stay Private

Keeping passwords private is crucial for several reasons:

Protects Personal Information

Prevents Unauthorized Access

Maintains Account Security

Safeguards Reputation

Facilitates Compliance

Select the correct answer about CSS

A. CSS helps us create visually appealing and user-friendly websites.
B. CSS doesn't define the appearance and layouts of the elements on a webpage.
C. CSS helps in creating a static HTML.
D. CSS does not have a function on the HTML.

CoGrammar

What is the primary purpose of responsive design?

A. To improve website loading speed.
B. To enhance SEO rankings.
C. To ensure websites look good on all devices.
D. To reduce website maintenance costs.

CoGrammar

Learning Outcomes

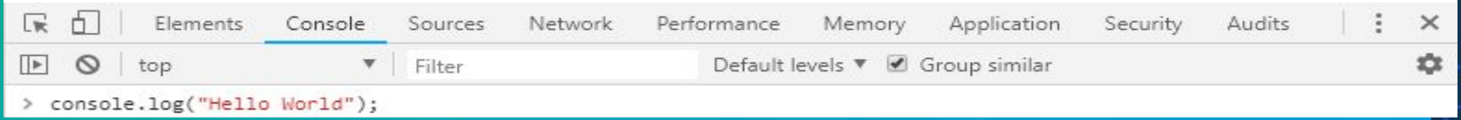
- Grasp and understand basic programming concepts in JavaScript, including variables, data types, and operators.
- Write conditional statements and understand their application in controlling the flow of a program.
- Utilize logical operators to make more complex decision structures.

Lecture Overview

- → Introduction to JavaScript
- → Variables and Data types
- → Conditional Statements
- → Logical Operators

INTRODUCTION TO JAVASCRIPT

- ❖ JavaScript is a versatile scripting language utilised in front-end web development and server-side programming
- ❖ When combined with HTML and CSS, JavaScript plays a pivotal role in transforming static web pages into dynamic ones, enabling interactions and real-time changes
- ❖ Browsers have built-in consoles used to debug JavaScript code. we will be using [Chrome’s DevTools Console](#).



```
> console.log("Hello World");
```

CoGrammar

INTRODUCTION TO JAVASCRIPT

- ❖ The console is useful for **debugging** and running **code snippets**.
- ❖ To create our scripts, we will use **Visual Studio Code** and **Node.js**.
- ❖ The following extensions are also helpful when running your code:
  - ➢ **Code Runner:** allows you to run JavaScript code in VS Code by pressing Ctrl+Alt+N, right-clicking and pressing “Run Code”, or by pressing the “Play” button.
  - ➢ **Open in Browser:** allows you to open an HTML file which has been correctly linked to JavaScript scripts in your default browser.

CoGrammar

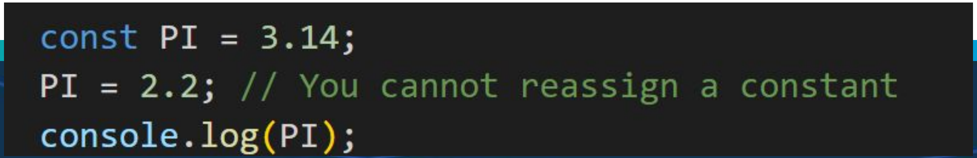
VARIABLES AND DATA TYPES

- ❖ Programs typically receive input data, process it, and then output the results.
- ❖ This data must be stored somewhere so that it can be used and processed by the instructions we code.
- ❖ A variable can be thought of as a type of ‘container’ that holds information.
- ❖ Before we can use variables in our code, we need to declare them

CoGrammar

VARIABLES AND DATA TYPES

- ❖ This tells JavaScript that we want to set aside a chunk of space in the computer's memory for our program to use.
- ❖ In JavaScript, we use the following format to create a variable and assign a value to it
  - ➢ ***let variableName = value***
- ❖ The keyword const can also be used to create variables, in a similar fashion to let.
- ❖ The keyword const stands for constant.
- ❖ It defines a constant variable, which stores the same value for as long as it exists.



```javascript
const PI = 3.14;
PI = 2.2; // You cannot reassign a constant
console.log(PI);
```

VARIABLES AND DATA TYPES

- You can set the name of the variable to anything you like, as long as the name:
  - Contains only letters, numbers, and underscores: All other characters are prohibited from use in variable names, including spaces.
  - Begins with a letter: In JavaScript, the common naming convention used is camelCase, where each word except for the first word starts with an uppercase letter.

CoGrammar

VARIABLES AND DATA TYPES

- You can set the name of the variable to anything you like, as long as the name:
  - Is not a reserved word: In JavaScript, certain words are reserved. For example, you would not be able to name a variable console or log because these are reserved words.
  - Is meaningful and concise: For example, ‘myName’ and ‘userInput’ are descriptive variable names as they explain their purpose

CoGrammar

VARIABLES AND DATA TYPES

- Variables store data.
- The type of data that is stored by a variable is called the data type
- Main data types in Javascripts are:
  - Numeric - data types describe any numbers that you store.
  - Strings - refer to a combination of characters. "Joe" and "23 Main Street" are examples of string values that must always be put within quotation marks (either "" or '')

CoGrammar

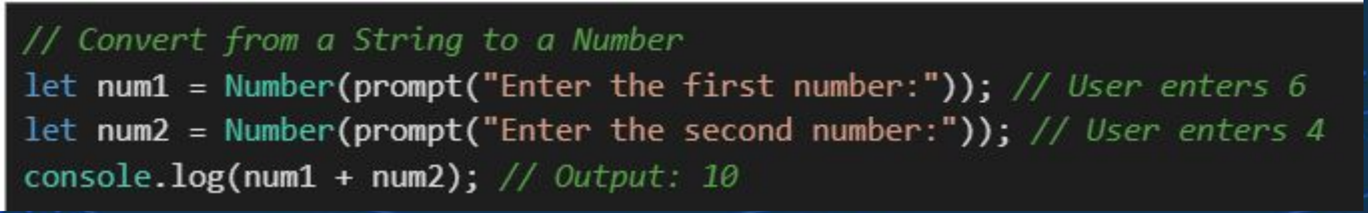
VARIABLES AND DATA TYPES

- Main data types in Javascripts are:
  - Booleans - are data types that can store only two values: either true or false.
  - Array - is a data type that is used to store multiple values.
  - Object - is a data type that stores a collection of related data. If you created a person object, for example, you could store a collection of related information that describes a person such as their name, surname, date of birth, address, etc.

CoGrammar

VARIABLES AND DATA TYPES

- Sometimes you may need to change a variable from one data type to another.
- Casting requires using the ***Number(), String(), or Boolean()*** constructor, depending on what we want the final data type to be.
- Once a variable has been converted to a number, the full range of mathematical operations can be performed with it.



```javascript
// Convert from a String to a Number
let num1 = Number(prompt("Enter the first number:")); // User enters 6
let num2 = Number(prompt("Enter the second number:")); // User enters 4
console.log(num1 + num2); // Output: 10
```

CoGrammar

VARIABLES AND DATA TYPES

- ❖ When we cast a variable to a boolean.
  - ➢ If we cast a number to a boolean, any number except zero (0) will return **true**.
  - ➢ If we cast a string to a boolean, any string except an empty string ("") will return **true**.

CoGrammar

VARIABLES AND DATA TYPES

- Mathematical calculations
- Doing calculations with numbers in JavaScript is similar to doing calculations in regular mathematics; the only difference is the symbols you use:
  - +, Addition
  - - , Subtraction
  - *, Multiplication
  - /, Division

CoGrammar

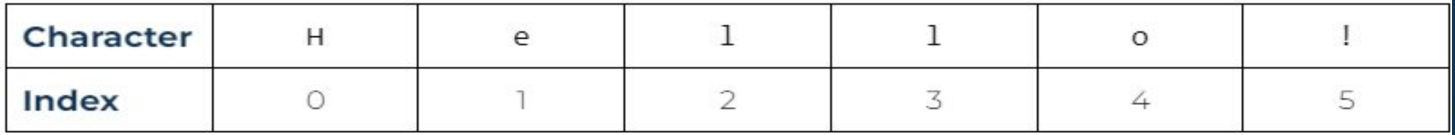
VARIABLES AND DATA TYPES

- Symbols we use in Javascript for calculations are:
  - %, Modulus (divides left-hand operand by right-hand operand and returns remainder, e.g. 5 % 2 = 1)
  - ++, Add one to a variable (e.g., 2++ = 3)
  - - -, Subtract one from a variable (e.g., 2-- = 1)
- When we are using ++ and -- it is essential to place them in the correct position. If you are placing the operator after the variable, it will update after the line is completed, whereas if it is placed before the variable, it will affect the current line.

CoGrammar

VARIABLES AND DATA TYPES

- The string data type
  - String is a combination of characters between quotation marks, e.g., "This is a string!".
  - We can access the characters in a string based on their position in the string, known as their index.



Character	H	e	l	l	o	!
Index	0	1	2	3	4	5

CoGrammar

VARIABLES AND DATA TYPES

- Multiline strings:
  - Strings can be combined, or concatenated, using the addition (+).
  - Strings can be combine, or concatenated, using template literals. With template literals, we simply place the variable name within the curly brackets. Any characters between the backticks such as spaces, punctuation, and string literals are also printed. Furthermore, we don’t need escape characters; if we want a new line, we simply go to the next line in our code as shown above.

CoGrammar

Let's take a break

CoGrammar

LOGICAL OPERATORS

- JavaScript has a Boolean type, which has just two values, **true** and **false**.
- Comparison operators results to **true** or **false**.
- List of comparison operators are:
  - >, greater than
  - <, less than
  - >=, greater than or equal to

CoGrammar

LOGICAL OPERATORS

- List of comparison operators are:
  - <=, less than or equal to
  - ==, equals
  - ===, Equal value and equal data type
  - !=, does not equal

CoGrammar

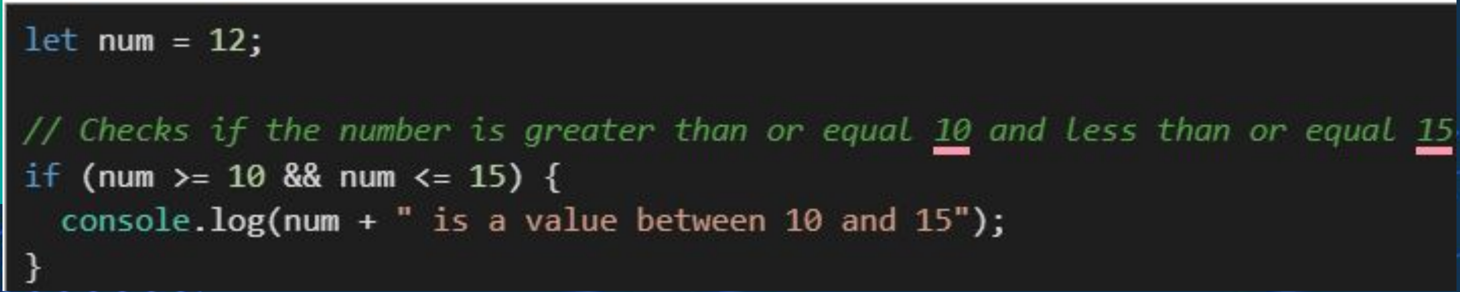
LOGICAL OPERATORS

- Logic operators are used to combine comparison expressions
- The resulting boolean expression will be evaluated as either true or false.
- Logic operators are:
  - &&, AND
  - | |, OR
  - !, NOT

CoGrammar

LOGICAL OPERATORS

- AND (&&) operator:
  - On the example code below, the boolean expression ***(num >= 10 && num <= 15)*** is **true** if both num >= 10 and num <= 15 are **true**. This is a conjunction, meaning both conditions must be **true** for the statement to be **true**. If either condition is **false**, the statement is **false**.



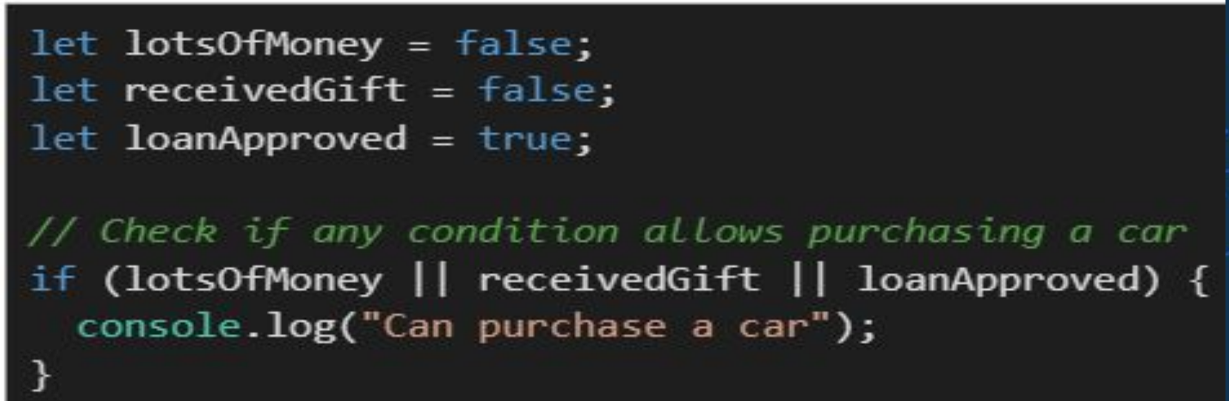
```
let num = 12;

// Checks if the number is greater than or equal 10 and less than or equal 15
if (num >= 10 && num <= 15) {
  console.log(num + " is a value between 10 and 15");
}
```

CoGrammar

LOGICAL OPERATORS

- OR (| |) operator:
  - Consider the example code below: You can buy a nice car if you have enough money, if someone gives you the money, or if you can get a loan. A disjunction operation requires at least one condition to be **true** for the whole statement to be **true**. For example, the boolean expression **(lotsOfMoney || receivedGift || loanApproved)** is **true** if at least one condition is **true**.



```
let lotsOfMoney = false;
let receivedGift = false;
let loanApproved = true;

// Check if any condition allows purchasing a car
if (lotsOfMoney || receivedGift || loanApproved) {
  console.log("Can purchase a car");
}
```

CoGrammar

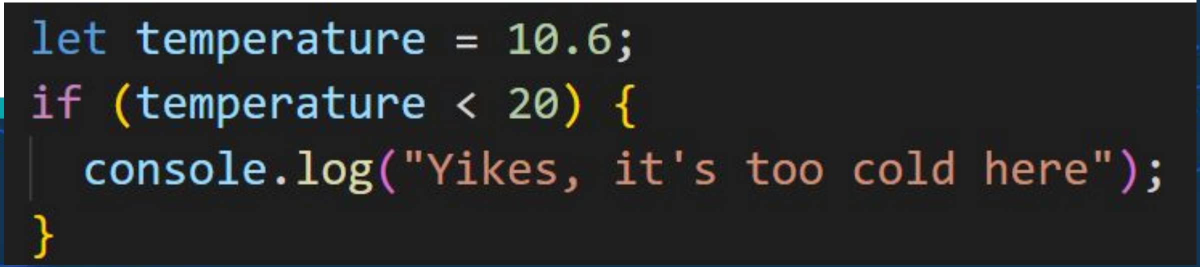
CONDITIONAL STATEMENTS

- ❖ Statements that perform different actions depending on whether a condition evaluates to true or false.
- ❖ Conditional execution is created with the if keyword in JavaScript.
- ❖ We want some code to be executed if, and only if, a certain condition holds.
- ❖ The deciding expression is written after the if keyword, between parentheses, followed by the statement to execute.

CoGrammar

CONDITIONAL STATEMENTS

- Statements that perform different actions depending on whether a condition evaluates to true or false.
- Conditional execution is created with the if keyword in JavaScript.
- We want some code to be executed if, and only if, a certain condition holds.
- The deciding expression is written after the if keyword, between parentheses, followed by the statement to execute.

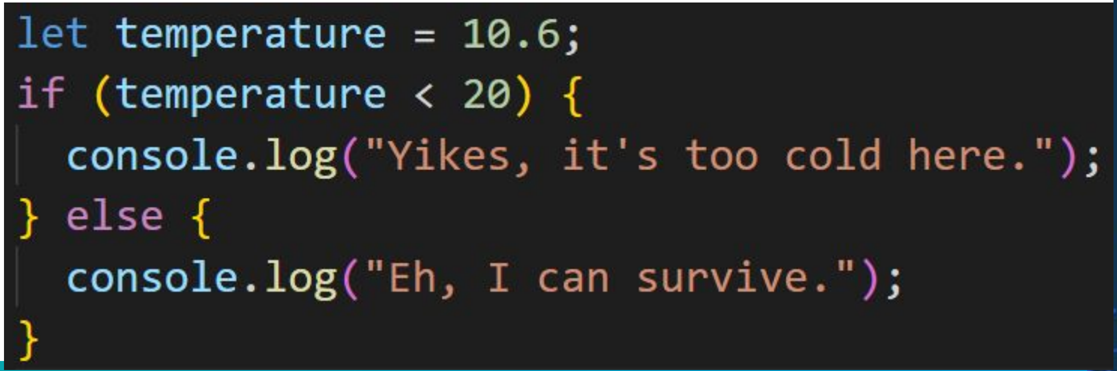


```
let temperature = 10.6;
if (temperature < 20) {
  console.log("Yikes, it's too cold here");
}
```

CoGrammar

CONDITIONAL STATEMENTS

- ❖ You can use the else keyword, together with if, to create two separate, alternative execution paths.
- ❖ The else statement represents an alternative path for the flow of logic if the condition of the if statement turns out to be false.

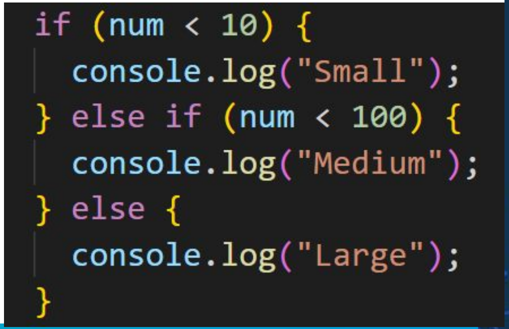


```
let temperature = 10.6;
if (temperature < 20) {
  console.log("Yikes, it's too cold here.");
} else {
  console.log("Eh, I can survive.");
}
```

CoGrammar

CONDITIONAL STATEMENTS

- ❖ The else if statement allows you to test multiple conditions in a single conditional structure.
- ❖ If the initial if condition is false, each else if is checked in sequence.
- ❖ If none are true, the final else statement is executed.



```
if (num < 10) {
  console.log("Small");
} else if (num < 100) {
  console.log("Medium");
} else {
  console.log("Large");
}
```

CoGrammar

What is the extension of a javascript file?

A. .css
B. .js
C. .doc
D. .html

CoGrammar

Booleans are data types that can store only two possible values.

A. True
B. False

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education