CoGrammar

Welcome to this session:

Object-Oriented Programming

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 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.

Why do we create objects in JavaScript? (Select all that apply)

A. To organize related data and functionality.
B. To model real-world entities.
C. To enhance code reusability and modularity.
D. To manage state in applications.

In JavaScript, what is the purpose of a Higher-Order Function?
(Select all that apply)

A. To return another function as a value.
B. To accept one or more functions as arguments.
C. To create more abstract and reusable code.
D. To manage asynchronous operations.

CoGrammar

Tutorial Outcomes

- Recap on the fundamentals of Object-Oriented Programming in JavaScript.
- Create, access, and manipulate objects in JavaScript to maintain and organize data efficiently.

Tutorial Overview

- Recap on Object-Oriented Programming (OOP)
- JavaScript Objects
- Working with Objects in JavaScript (Accessing and manipulation of objects)

RECAP ON OOP

- What is object-oriented programming?
  - Object-Oriented Programming (OOP) mirrors how people perceive the world—through objects.
  - OOP organises code around real-world entities, simplifying complex problems and making systems intuitive.
  - Concepts like “windows” or “folders” in software mirror their physical counterparts, reducing cognitive load for users and developers.

RECAP ON OOP

- ❖ Procedural programming versus object-oriented programming
  - ➢ Procedural programming - instructions are executed sequentially, one after another. Functions are independent but can access global variables. The data is stored separately from the code that processes it. The key features are sequential execution and the separation of data from the functions that operate on it.

CoGrammar

INTRODUCTION TO OOP

- OOP - a system is designed in terms of objects that communicate with each other to accomplish a given task. Instead of separating data and code that manipulates the data, these two are encapsulated into a single module. Data is passed from one module to the next using methods.

CoGrammar

RECAP ON OOP

- ❖ How do we design object-oriented programming systems?
  - ➢ In most OOP languages, an object is created using a class. A class is a blueprint from which objects are made, consisting of both data and the code that manipulates the data.

CoGrammar

JAVASCRIPT OBJECTS

- There is more than one way of creating objects.
  - Method 1: Using object literals
  - Method 2: Using object constructors
  - Method 3: Using a class

CoGrammar

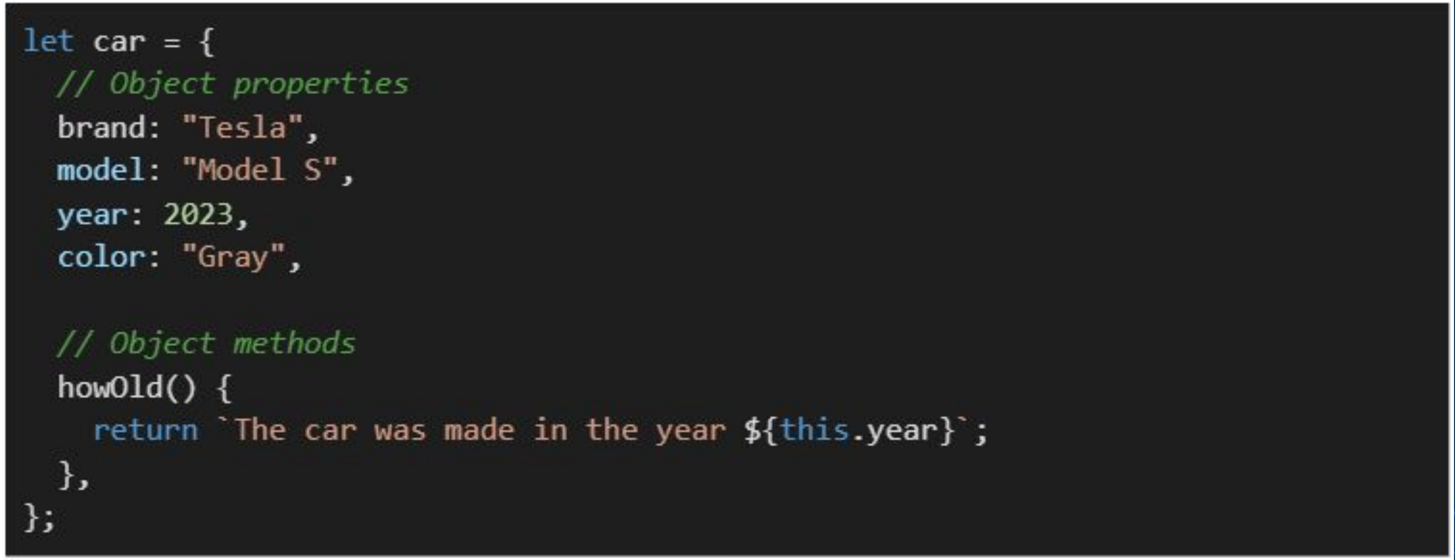
JAVASCRIPT OBJECTS

- ❖ **Method 1: Using object literals**
  - ➢ Objects in JavaScript store a collection of key–value pairs called properties.
  - ➢ When a function is declared within an object, it is known as a method. Methods are a series of actions that can be carried out on an object. An object can have properties that store different types of data, such as numbers, strings, functions, or even other objects. Therefore, an object method is simply a property that has a function as its value.

CoGrammar

JAVASCRIPT OBJECTS
Object Literals

➢ Let’s consider the following object declaration:



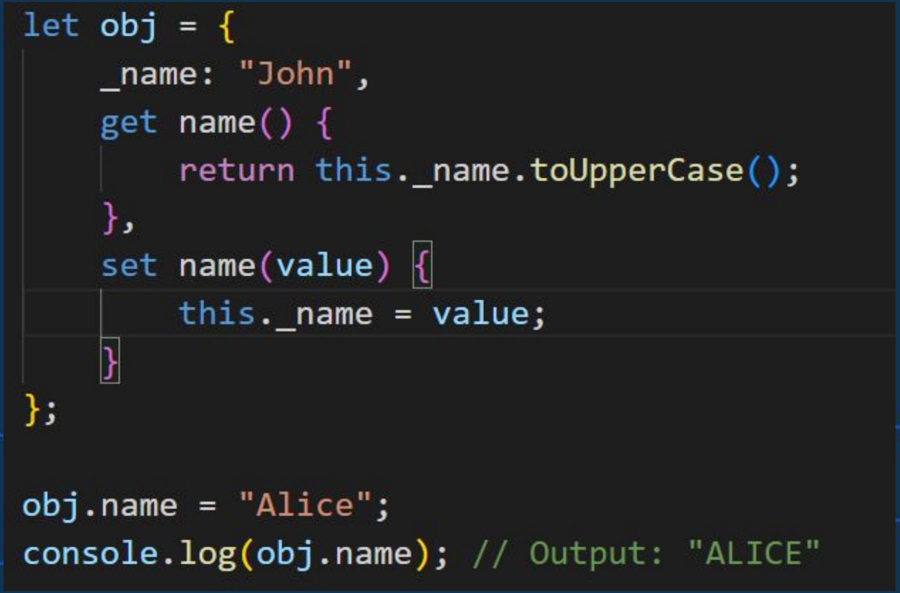
```
let car = {
  // Object properties
  brand: "Tesla",
  model: "Model S",
  year: 2023,
  color: "Gray",

  // Object methods
  howOld() {
    return `The car was made in the year ${this.year}`;
  },
};
```

CoGrammar

Accessor Properties in JavaScript Objects

- Accessor properties in JavaScript objects are defined using **getters** and **setters**.
- They allow for **controlled access** and **manipulation** of object **properties**.



```
let obj = {
    _name: "John",
    get name() {
        return this._name.toUpperCase();
    },
    set name(value) {
        this._name = value;
    }
};

obj.name = "Alice";
console.log(obj.name); // Output: "ALICE"
```

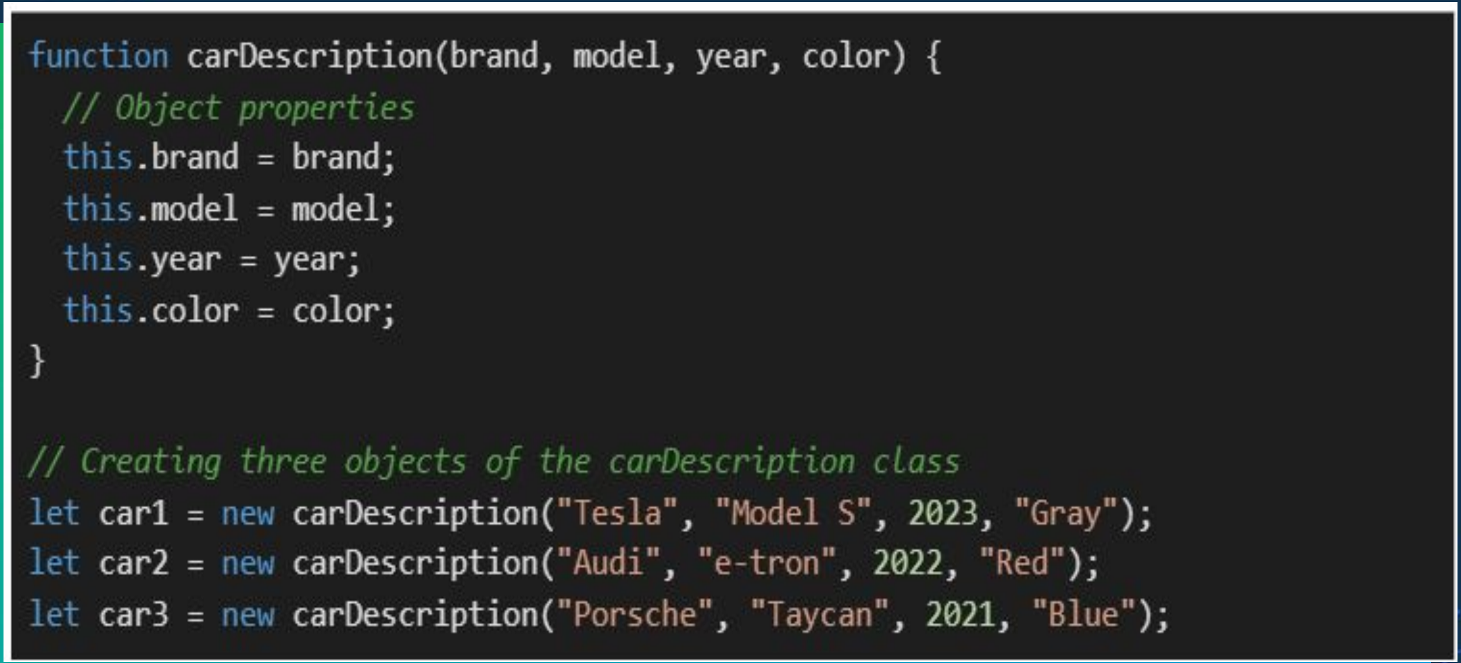
CoGrammar

JAVASCRIPT OBJECTS

- ❖ **Method 2: Using object constructors**
  - ➢ This method provides an effective way of creating many objects using an object constructor.
  - ➢ A constructor is a special type of function that is used to make or construct several different objects.
  - ➢ Let’s consider the example below:

CoGrammar

JAVASCRIPT OBJECTS
Object Constructors



```javascript
function carDescription(brand, model, year, color) {
  // Object properties
  this.brand = brand;
  this.model = model;
  this.year = year;
  this.color = color;
}

// Creating three objects of the carDescription class
let car1 = new carDescription("Tesla", "Model S", 2023, "Gray");
let car2 = new carDescription("Audi", "e-tron", 2022, "Red");
let car3 = new carDescription("Porsche", "Taycan", 2021, "Blue");
```

CoGrammar

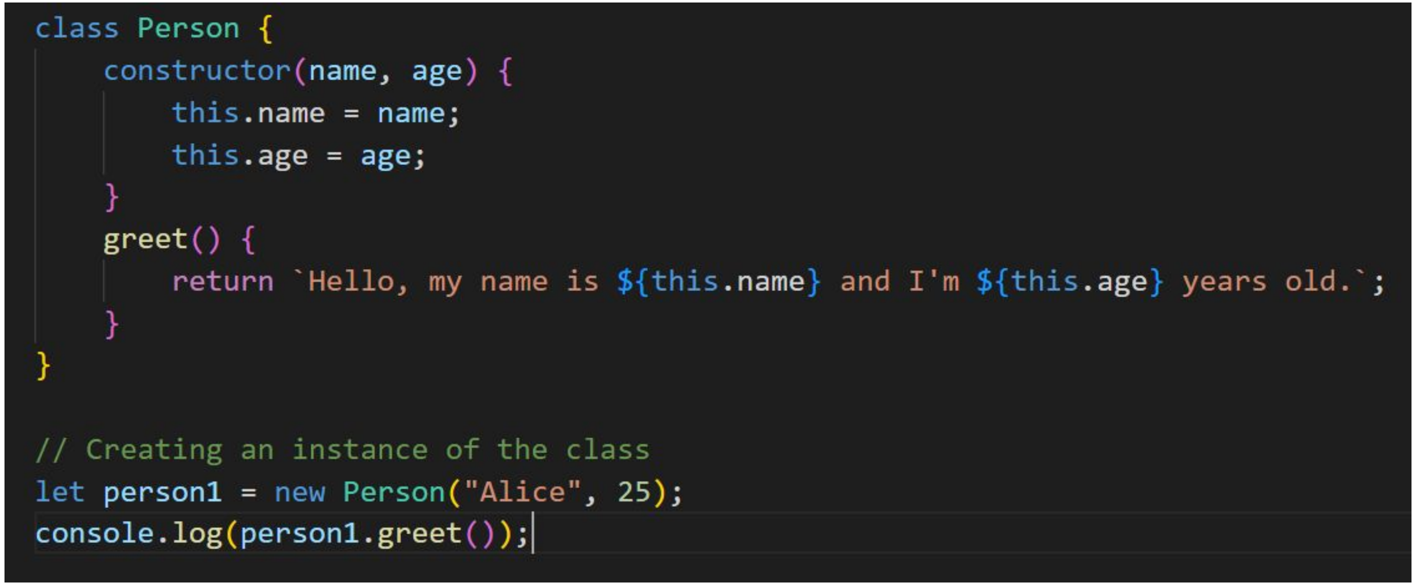
Is JavaScript considered as a pure OOP language?

- ➢ Prototype-Based OOP.
- ➢ Objects and Functions.
- ➢ Flexible Typing.
- ➢ Mixins and Composition.
- ➢ No Enforced Structure.

JAVASCRIPT OBJECTS

- **Method 3: Using a class**
  - ES6 introduced class syntax to JavaScript, offering a more familiar and structured way to create objects and manage inheritance.
  - Classes provide syntactic sugar over prototype-based inheritance, making object-oriented programming in JavaScript more intuitive.
  - Let’s consider the example below:

JAVASCRIPT OBJECTS
Class



```
class Person {
    constructor(name, age) {
        this.name = name;
        this.age = age;
    }
    greet() {
        return `Hello, my name is ${this.name} and I'm ${this.age} years old.`;
    }
}

// Creating an instance of the class
let person1 = new Person("Alice", 25);
console.log(person1.greet());
```

CoGrammar

In JavaScript, which is an example of creating an object?

A. let obj = { name: "Alice", age: 30 };
B. let arr = [1, 2, 3];
C. let str = "Hello, world!";
D. let num = 42;

CoGrammar

In JavaScript, what does a key-value pair refer to?

A. A way to define functions and their parameters.
B. An array of values indexed by their position.
C. A way to store data in objects.
D. A way to manage asynchronous operations.

Let's take a break

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE SKILLS BOOTCAMPS

Department for Education