Module 4: OOP. Prototypes. More on objects and inheritance

In this Module:

In-depth journey to Object creation and inheritance in JavaScript

- Prototypes
- The ‘new’ keyword, video
- Class keyword
- Ways to create an Object
- Inheritance

Extra:

- Forgotten history of OOP
- Composition over inheritance

Prototype basics - Object Creation in JavaScript P3

“You’re probably used to doing inheritance with classes. JavaScript achieves inheritance using prototypes. The real-world analogy used when teaching classes is a blueprint - you have a blueprint of a building, and you use that blueprint to build buildings. The real-world analogy to a prototype is a delegate, a person that has been elected into office that will act on your behalf. ”
https://www.youtube.com/watch?v=YkoelSTUy7A

The 'new' keyword - Object Creation in JavaScript P4

“*We explore the new keyword in JavaScript works when applied to functions. This is the old school way of faking classes in JavaScript, prior to the class keyword being introduced in ES6. ”*
https://www.youtube.com/watch?v=Y3zzCY62NYc

Class keyword - Object Creation in JavaScript P7

“The class keyword was added to JavaScript in ES6. On the surface, class seems to works like you’d expect, but if you look just a little closer you’ll see that class is just a thin veil on top of the existing prototypal inheritance model. In this video, I try to convince you that you should focus on learning JavaScript inheritance properly, in order to understand the mechanics of the class keyword in JavaScript.”
https://www.youtube.com/watch?v=Tllw4EPhLiQ

Object.create - Object Creation in JavaScript P6

“The Object.create() method in JavaScript creates a new object with the specified prototype object and properties. I walk through what it is, why Object.create exists in JavaScript, and how to use Object.create.”
https://www.youtube.com/watch?v=GhbhD1HR5vk

Inheritance in JavaScript (article)

“this article shows how to create "child" object classes (constructors) that inherit features from their "parent" classes. In addition, we present some advice on when and where you might use OOJS, and look at how classes are dealt with in modern ECMAScript syntax”
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Inheritance

Extra:

The Forgotten History of OOP

“*Historical research on the history of OOP and how it’s applicable in today’s JS*”
https://medium.com/javascript-scene/the-forgotten-history-of-oop-88d71b9b2d9f

Composition over Inheritance

“Using composition is actually much easier and more convenient than inheritance in JS”
https://www.youtube.com/watch?v=wfMtDGfHWpA