1. React Component Patterns by Michael Chan

(<https://www.youtube.com/watch?v=YaZg8wg39QQ>)

1. How the Internet Works for Developers

(<https://www.youtube.com/watch?v=e4S8zfLdLgQ>)

1. Object.create
2. Property descriptors
3. setPrototypeOf
4. typeof() table
5. Why would you use ReactJS to write the application

**What is JavaScript?**

What is JavaScript? JavaScript is primarily a client-side language. JavaScript started at Netscape, a web browser developed in the 1990s. A webpage can contain embedded JavaScript, which executes when a user visits the page. The language was created to allow web developers to embed executable code on their webpages, so that they could make their webpages interactive, or perform simple tasks. Today, browser scripting remains the main use-case of JavaScript.

JavaScript’s syntax is heavily inspired by C++ and Java. If you have experience in C++ or Java, JavaScript’s syntax will seem familiar to you. However, the inner workings of JavaScript is closer to a dynamically-typed, interpreted language such as Python or Ruby.

JavaScript is an interpreted language, not a compiled language. A program such as C++ or Java needs to be compiled before it is run. The source code is passed through a program called a compiler, which translates it into bytecode that the machine understands and can execute. In contrast, JavaScript has no compilation step. Instead, an interpreter in the browser reads over the JavaScript code, interprets each line, and runs it. More modern browsers use a technology known as Just-In-Time (JIT) compilation, which compiles JavaScript to executable bytecode just as it is about to run.

JavaScript is named after Java, and many ideas are borrowed from the Java language. Other than that, Java and JavaScript are two entirely distinct languages. The most significant difference between them is that Java is a compiled language, and JavaScript is a interpreted language. JavaScript runs on many browsers out-of-the-box, whereas Java applets require an additional plug in. Both languages have different runtime environments, different governing bodies, different libraries.

## Javascript is not a compiled language - period.

A Compiled language is one that when compiled it converts language code into either machine code (to run on the metal - eg c++), or bytecode (to run in a VM - eg Java / C#), and this is done 'Ahead of Time' (AOT), and you deploy the compiled code.

An interpreted language is one where the language code is compiled to machine code or bytecode at the moment of use. the language code is deployed 'as-is' and the interpreter will do the work when the app is running.

When Javascript developers talk about compiling they are really talking about something else - usually tree-shaking and minifying etc. The output of the javascript 'compile' phase is just an optimised string of Javascript code.

Best Practice

However, the hoisting behaviour is not applied to let and const. The best practice is to always use **const**. If you need the variable to be updated, then use **let**. There should be almost no circumstance in which using **var** is the better option, and so hoisting should not be an issue.