为什么学习ECMAScript 6?

一、前后端分离（公司朝前后端分离这个方向走，而前后端分离的主流技术就三个，vue，react，angular，目前我们前端商量的结果是采用vue开发，它具有学习曲线平缓，社区活跃，轻量等优点，在vue开发中，我们常用ECMAScript 6来提高开发效率）

二、能够看懂其他人的开源项目（在平时的工作中，有时我们需要借鉴别人的方法或学习其他人的开发技巧，这时我们会去github找类似的开源项目，有的项目是用ECMAScript 6开发的，如果我们不会，我们就看不懂）

三、Node 是 JavaScript 的服务器运行环境（runtime）。它对 ES6 的支持度更高，如果JavaScript功底深厚，学习nodejs，走全栈也是不错的选择。

ECMAScript 6的前世今生

知其历史，才能更知其当下

1.1995年05月：JavaScript诞生，它的初始名叫LiveScript

2.1996年11月，JavaScript 的创造者 Netscape 公司，决定将 JavaScript 提交给标准化组织 ECMA，希望这种语言能够成为国际标准

3.1997年06月 ECMAScript 1.0（首版）

6.2000年 ECMAScript 4.0开始酝酿

2015年6月 ECMAScript 6.0（ECMAScript2015）发布，从 2000 年算起，这时已经过去了 15 年

**一、块作用域构造let和const**

let是一种新的变量声明方式，它允许你把变量作用域控制在块级里面。我们用大括号定义代码块，在ES5中，块级作用域起不了任何作用

const声明一个只读的常量。一旦声明，常量的值就不能改变

let命令的特点：

一、let 声明的变量的作用域是块级的

二、let 不能重复声明已存在的变量

三、let 有暂时死区，不会被提升

透过现象看本质

**我们来看看 var 声明的「创建、初始化和赋值」过程**

function fn(){

console.log(x)

var x = 1

var y = 2

}

fn()

在执行 fn 时，会有以下过程（不完全）：

1. 进入 fn，为 fn 创建一个环境。
2. 找到 fn 中所有用 var 声明的变量，在这个环境中「创建」这些变量（即 x 和 y）。
3. 将这些变量「初始化」为 undefined。
4. 开始执行代码
5. x = 1 将 x 变量「赋值」为 1
6. y = 2 将 y 变量「赋值」为 2

也就是说 var 声明会在代码执行之前就将「创建变量，并将其初始化为 undefined」。

这就解释了为什么在 var x = 1 之前 console.log(x) 会得到 undefined。