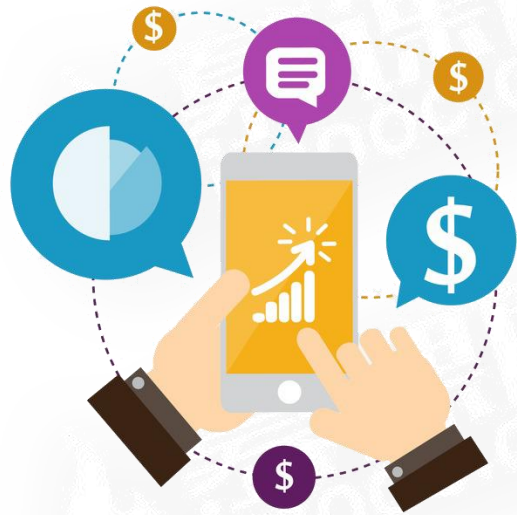


初识ECMA Script 6

ECMAScript





ECMAScript与JavaScript的关系

ES是JS的标准

JS是ES的实现

历史



历史



历史

The Oracle logo, consisting of the word "ORACLE" in a bold, red, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E".

ORACLE®

历史

European Computer Manufacturers Association



ECMAScript的部分历史

1998-06 ES2发布

1999-12 ES3发布 成为js的通行标准 被广大浏览器支持

2007-10 ES4草案发布 内容十分激进

2008-07 将ES4小部分内容提取 发布为ES3.1 后改名为ES5

2009-12 ES5发布

2011-06 ES5.1发布 并成为了ISO国际标准

ECMAScript的部分历史

2013-03 ES6的草案冻结 不加入新功能

2015-06 ES6正式通过 成为国际标准

课程内容

let 和 const

解构赋值

let 和 const

let

const

块级作用域

解构赋值

数组的解构赋值

对象的解构赋值

字符串的解构赋值

数值与布尔值的解构赋值

函数参数的解构赋值

开发环境准备

编辑器(VS Code、Atom、Sublime)或IDE(Webstorm)

浏览器(最新的Chrome) 或 Node.js环境(越新越好)

浏览器对新标准的支持

<http://kangax.github.io/compat-table/es6/>

let与块级作用域

变量声明

var

直接使用

变量声明

```
window.location.href === location.href
```

let和var的主要区别

let声明的变量只在当前(块级)作用域内有效

let声明的变量不能被重复声明

不存在变量提升

ES6之前的作用域

全局作用域

函数作用域

eval作用域

块级作用域

通俗的讲 就是一对花括号中的区域 { ... }

块级作用域可以嵌套

```
1 {  
2     // 嗨~ 我是块级作用域 1 号  
3  
4     {  
5         // 嗨~ 我是 2 号  
6     }  
7 }
```

const

const

常量 – 不可改变的量

使用const

和声明变量一样 基本只是关键字的区别

```
1 var v = '我是变量';  
2 const c = '我是常量';
```

常量必须被赋值

常量必须在声明的时候赋值

否则报错: Missing initializer in const declaration

与let类似的特性

不能重复声明

不存在提升

只在当前(块级)作用域内有效

常量不可变？

一旦声明常量 就不能再改变？

常量为引用类型的时候 不能保证不可变

const

关于const的一些问题

- *q:怎么解决引用类型的常量可以被修改的问题*
- *q:ES6之前怎么声明常量*

课程总结

let

let不能被重复声明

不存在变量提升

let声明的变量只在当前(块级)作用域内有效

const

不能重复声明

不存在提升

一旦声明常量 就不能再改变（引用类型需要冻结）

let声明的变量只在当前(块级)作用域内有效