深度理解ES6中新增变量声明方式let和const

jCodeLife
0.321 2020.08.31 06:12:36 字数 1,302 阅读 42

编辑文章

首先我们来谈谈关于JS中的变量声明方式

在ES5中，变量声明只有var 和function以及隐式声明三种，而ES6中新增了let、const以及import和class四种

其中let和const其实就是用于替代var，一个用于声明变量，一个用于声明常量，import是用于模块化导入，class是用于声明一个类

本小结主要讲解let和const，其他两个会开其他章节讲解

我们首先来思考一个问题，为什么从ES5升级到ES6，组织会新增let和const？那肯定是var有什么不好的地方吧，那跟var相比他们有什么特点和优势呢，带着这个问题我们继续往下看。

let

let用于声明一个变量

```
let a
```

用let声明的变量具有以下几个特点

1. let声明的变量其作用域是块级作用域

在ES5 并没有块级作用域的概念，只有函数作用域和全局作用域。所以用var声明的变量如果不在函数内声明，默认就是在全局

```
for (var i = 0; i <10; i++) {
  setTimeout(function() {  // 同步注册回调函数到异步的宏任务队列。
    console.log(i);        // 执行此代码时，同步代码for循环已经执行完成
  }, 0);
}
 console.log(i);
```

输出结果是11个10
每次循环，新的i值都会覆盖旧值，最终i的结果为10
变量i在for循环中定义，其实就是在全局上定义，在全局范围内都有效，即相当于window.i=10。所以后面访问的i都是10

把 var改成 let声明

```
for (let i = 0; i < 10; i++) {
  setTimeout(function() {
    console.log(i);
  }, 0);
}
```

输出结果：
0 1 2 3 4 5 6 7 8 9
每一次循环，i其实都是一个新产生的变量。
而这时如果在全局console.log(i)，结果报错：Uncaught ReferenceError: i is not defined

jCodeLife
总资产20 (约1.80元)

一篇搞清AJAX及实现手动封装
阅读 23

webpack的基本认知和使用
阅读 2

简单了解H5新增了哪些东西
阅读 5

推荐阅读

vue+elementUI 可编辑表格
阅读 737

new的过程
阅读 9

不知道怎么封装代码？看看这几种设计模式吧！
阅读 668

typeof和instanceOf的区别
阅读 159

学习JavaScript的第一周
阅读 19

2. let存在TDZ--暂时性死区（Temporal Dead Zone）

其实所谓的暂时性死区就是：即变量所在的作用域的开始位置 到变量声明那行结束的空间位置区域

临时死区导致let变量并不能在声明前使用，只有等到声明变量的那一行代码出现，才可以获取和使用该变量

暂时性死区的意义在于标准化代码，让所有变量的声明放在作用域的最开始。

3. 在同一块级作用域中，let不允许重复声明变量

在一个块级作用域中，变量唯一存在！一旦在块级作用域中声明了一个变量，就不能在这个作用域中使用let重复定义

```
function fun() {
  let a = '123';
  var a = '456';// 报错
}
function fun() {
  let a = '123';
  let a = '456';// 报错
}
```

由此需要注意：不能在函数内部使用let声明与参数名一致的变量

```
function fun(a) {
  let a; // 报错
}
```

4. let 没有预编译的说法，即不存在变量提升

也就是变量定于在哪里，就从哪里开始可以使用。这点跟TDZ相映，在暂时死亡区域是不能使用该变量的

```
console.log(a);//报错
let a = 1;
```

以上是let的几个特点！接着来看const

const 用于定义一个常量

const的特性与let基本一致：块级作用域、暂时性死区、不能重复声明，没有变量提升等

不同的是：

1. const必须在声明的时候赋值：

```
const a//报错
```

如果不赋值否则会报SyntaxError

2. const声明变量并赋值后不能再修改

```
const b = 101;
b = 102;//报错
```

如果去修改一个常量会报TypeError；

注意：特殊情况，当声明的常量是一个对象时，那么对于对象本身是不允许重新赋值的，但是

```
console.log(obj.name);    // xxx
```

这里不要诧异哦，其实const声明的常量保存的是对象的引用，const能保证的是这个对象的内存地址不能改动，至于具体里面的某些属性，你爱改就改，我不管，但如果你要重新赋值一个对象或者其他什么值，就报错！

但还是有特殊情况，`如果使用Object.freeze(obj)，彻底将一个对象或数组冻结，让其属性也不可修改`

```
const obj = {
  prop: 1
};
Object.freeze(obj);
obj.prop = 2;//不会报错，但是不会改变该属性值
console.log(obj.prop);//1
```

以上是let和const的全部内容！

最后来总结一下

1. let 与 const 相同点：

   `块级作用域`

   `有暂时性死区`

   `约束了变量提升`

   `禁止重复声明变量`

2. let 与 const不同点：

   `let可以先声明后赋值，之后也可以再改值；`

   `const声明的变量不能重新赋值，也是由于这个规则，const变量声明时必须初始化，不能留到以后赋值。`

3. 与var相比，let和const声明的变量会让代码更加严谨和规范

jCodeLife 书山有路勤为径，学海无涯苦作舟
总资产20 (约1.80元) 共写了13.2W字 获得282个赞 共23个粉丝