博学谷
www.boxuegu.com

第一章 let和const命令

一样的在线教育，不一样的教学品质

目录 Contents

1.let命令基本用法

let命令基本用法

ES6中新增了let命令，用于**变量的声明**，**基本的用法和var命令类似**。

```
<script>
    // 使用var使用声明变量
    var userName = "bxg";
    console.log("userName=", userName);
    // 使用let声明变量
    let userAge = 18;
    console.log("userAge=", userAge);
</script>
```

2.let命令与var命令的区别

区别：

通过let声明的变量只在其**对应的代码块**中起作用；

示例：

```
<script>
        for (var i = 1; i <= 10; i++) {
            console.log("i=", i)
        }
        console.log("last=", i)
    </script>
```

总结：

let声明的变量只在代码块中起作用，即通过let声明的变量仅在**块级作用域**内有效。

3.什么是块级作用域？

块级作用域概念

有一段代码用**大括号包裹**起来，那么，大括号内就是一个块级作用域；

示例：

```
for (let i = 1; i <= 10; i++) {
        console.log("i=", i)
    }
console.log("last=", i)
```

说明：i 变量的作用域只在这一对大括号内有效，超出这一对大括号则无效。

4.块级作用域的作用

为什么需要块级作用域？

ES5 只有全局作用域和函数作用域，没有块级作用域，因此会引发部分问题：

- 内层变量可能会覆盖外层变量。
- 用来计数的循环变量成为了全局变量。

5.ES6块级作用域

ES6块级作用域

let实际上为JavaScript新增了块级作用域;

示例：

```
<script>
        function test() {
            let num = 5;
            if (true) {
                let num = 10;
            }
            console.log(num)
        }
        test()
</script>
```

上面代码执行结果是多少？

5.ES6块级作用域

示例：

下面程序执行结果是多少？

```
if (true) {
            let b = 20;
            console.log(b)
            if (true) {
                let c = 30;
            }
            console.log(c);
        }
```

5.ES6块级作用域

块级作用域的出现，带来的好处：

之前获得广泛使用的**立即执行匿名函数**将不再需要。

定义一个立即执行匿名函数：

```
;(function text() {
            var temp = 'hello world';
            console.log('temp=', temp);
 })()
```

匿名函数的好处：

通过定义一个匿名函数，创建了一个新的函数作用域，相当于创建了一个“私有”的空间。

该空间内的变量和方法，不会破坏污染全局的空间 。

5.ES6块级作用域

示例：

```
{
  let temp = 'hello world';
  console.log('temp=', temp);
}
```

通过以上的写法，创建了一个“私有”的空间，即创建了一个封闭的作用域。

在该封闭的作用域中的变量和方法，不会破坏污染全局的空间。

优点：块级作用域比立即执行匿名函数简单便捷。

5.ES6块级作用域

块级作用域的优势：

示例：

不同时间打印变量i的值：

```
for (var i = 0; i < 3; i++) {
                setTimeout(function() {
                    console.log('i=', i);
                }, 1000)
        }
```

以上程序，输出的都是：i=3

原因：I 为全局的；

5.ES6块级作用域

ES5实现：

```
for (var i = 0; i < 3; i++) {
            (function(i) {
                setTimeout(function() {
                    console.log('i=', i);
                }, 1000)
            })(i)
        }
```

通过自定义一个函数，生成函数的作用域，i变量就不是全局的了。

Let命令实现：

```
 for (let i = 0; i < 3; i++) {
             setTimeout(function() {
                 console.log('i=', i);
             }, 1000)
         }
```

6.Let命令注意事项

- **不存在变量提升**

  相比较var命令，Let命令不会发生“变量提升”现象。因此，**变量一定要在声明后使用**，否则会出错。

- **暂时性死区**

  所谓的“暂时性死区”指的就是：**在代码块内，使用let命令声明变量之前，该变量都是不可用的。**

- **不允许重复声明**

  let 不允许在相同的作用域内重复声明一个变量；如果使用var声明变量是没有这个限制的。

let命令-总结

重难点

1. let命令基本用法
2. Let命令与var命令的区别
3. 块级作用域
4. let命令注意事项

目录 Contents

1.const命令的基本用法

Const命令说明：

Const命令用来**声明常量**；常量指的就是**一旦声明，其值是不能被修改的。**

而变量指的是，在程序运行中，可以改变的量。

```
let num = 12;
        num = 30;
        console.log(num)
```

1.const命令的基本用法

示例：

```
const PI = 3.14;
console.log(PI)
```

说明：如果确定某个值不需要后期更改，就可将其定义成常量。

2.const命令的注意事项

- 不存在变量提升
- 只在声明的块级作用域内有效
- 暂时性死区
- 不允许重复声明
- 常量声明必须赋值

const命令-总结

重难点

1. const命令基本使用。
2. const命令注意事项。

一样的在线教育，不一样的教学品质