let 与 const

2019年12月20日

10:50

LET:

let 可以有块作用域,在块作用域内声明的变量外部不能访问

|  |  |
| --- | --- |
| 1  2  3 | {  let x = 10;  } |

// 此处不可以使用 x

let在块内声明的变量,不会对块外的变量造成影响

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var x = 10;  // 此处 x 为 10  {  let x = 6;  // 此处 x 为 6  }  // 此处 x 为 10 |

在相同的作用域, var 允许重复定义,;let 不允许重新定义变量,let定义的变量也不允许重新定义

|  |  |
| --- | --- |
| 1  2  3  4 | var x = 10;  // 现在，x 为 10  var x = 6;  // 现在，x 为 6 |

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var x = 10; // 允许  let x = 6; // 不允许  {  var x = 10; // 允许  let x = 6; // 不允许  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | let x = 10; // 允许  let x = 6; // 不允许    {  let x = 10; // 允许  let x = 6; // 不允许  } |

CONST:

在块作用域内使用,作用限制于块内部

必须在声明时赋值,const 不允许重新赋值

const定义的是对值的常量引用,可以改变对象的属性,可以添加删除属性,但不允许对对象重新赋值;数组同理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1  2  3 | const PI = 3.141592653589793;  PI = 3.14; // 会出错  PI = PI + 10; // 也会出错 |  |  |  |

|  |  |
| --- | --- |
| 1  2  3  4 | // 您可以创建 const 对象：  const car = {type:"porsche", model:"911", color:"Black"};  // 您可以更改属性：  car.color = "White"; |

const,let与var 定义的对象不同,不能被提升到顶端,即const定义的变量,引用必须在声明之后;

const与let定义的遍历不会添加到全局,即this.var为undefined.