

# JavaScript进阶（ ES6 ）

---ES6背景知识



河北师范大学软件学院  
Software College of Hebei Normal University

## 背景知识 ( ES5、ES6 )

---

- ECMAScript ( ES ) 是JavaScript的语法标准
- ECMAScript的版本 ( ES5、ES6 )
  - ES5 ( 2009年12月发布 )
  - ES6 ( 2015年06月发布 ) 增加了许多新特性，并解决了很多ES5中的缺陷
- 如何查看当前环境对ES6的支持情况
  - <https://ruanyf.github.io/es-checker/index.cn.html>
  - <https://kangax.github.io/compat-table/es6/> 和 <http://node.green/>
- 遇到兼容问题时如何将ES6转为ES5 ( <http://babeljs.io/> )

# JavaScript进阶（ ES6 ）

---ES6中的let与const



河北师范大学软件学院  
Software College of Hebei Normal University

# 内容提纲

---

- **ES5中的var及其缺陷**
- **ES6中的let与const**
- **let与const的重要特性**



## ES5中的var及其缺陷

- ES5通过var声明变量（无块作用域，可能造成变量污染）

```
var userId = 12;  
document.onclick = function () {  
    console.log("userId = ",userId);  
};
```

//一长串代码后，忘记了上边定义的userId

```
var a=2,b=3;  
if(a<b){  
    var userId = 34;//重复定义，变量污染  
}
```

- ES5中变量生命周期解决方案IIFE

# ES5中的var及其缺陷

- ES5通过var声明变量（可能造成变量的非期望共享）

```
> for (var i = 0; i < 3; i++) {  
    (function(j) { // j = i  
        setTimeout(function() {  
            console.log(new Date, j);  
        }, 1000*i);  
    })(i);  
}
```

< undefined

Fri Sep 01 2017 15:36:02 GMT+0800 (中国标准时间) 0

Fri Sep 01 2017 15:36:03 GMT+0800 (中国标准时间) 1

Fri Sep 01 2017 15:36:04 GMT+0800 (中国标准时间) 2

# 内容提纲

---

- ES5中的var及其缺陷
- ES6中的let与const
- let与const的重要特性



# ES6中的let与const

## •通过let声明变量

- ES6新增了let命令，用于声明变量，用法与var类似
- 其与var的不同在于，用let声明的变量只在 let 命令所在的代码块 { }内有效

```
let userId = 123;  
document.onclick = function () {  
    console.log("userId = ",userId);  
};
```

```
let a=2,b=3;  
if(a<b){  
    let userId = 234;//不会造成变量污染  
}
```



# ES6中的let与const

- 使用let声明变量，可有效避免变量共享缺陷

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

Tue Sep 05 2017 16:29:26 GMT+0800 (中国标准时间) 0

Tue Sep 05 2017 16:29:27 GMT+0800 (中国标准时间) 1

Tue Sep 05 2017 16:29:28 GMT+0800 (中国标准时间) 2

# ES6中的let与const

- ES6使用const来声明常量，也常用来声明不变的函数

```
const PI = 3.1415926;  
console.log(PI);  
PI = 3; //给常量再赋值 报错
```

```
//const声明时必须赋值  
//一旦声明必须立即初始化  
const foo; //报错  
const foo = 123; //ok
```

```
//const作用域同let  
if(true){  
    const MAX = 5;  
    console.log(MAX); //5  
}  
console.log(MAX); //报错
```

```
//const也常用来声明不变的函数  
const fee = function () {};
```

- const指向的对象引用不可变，但其属性是可变的

# 内容提纲

---

- ES5中的var及其缺陷
- ES6中的let与const
- **let与const的重要特性**



## ES6 let变量和const常量的特性--不进行变量提升特性

- var的 ‘声明提升’ 的特性（声明前可以使用该变量）

```
console.log(a);  
var a = 1;  
console.log(a);
```

等  
价  
于  
=>

```
var a;  
console.log(a); // undefined  
a = 1;  
console.log(a); // 1
```

- 用let声明变量时，在声明前不能使用该变量（let不提升）

```
console.log(a);  
let a = 2;  
console.log(a);
```

► Uncaught ReferenceError: a is not defined

# ES6 let变量和const常量的特性--暂时性死区特性

- 暂时性死区 ( temporal dead zone ) 特性

- 只要块级作用域内存在let，它所声明的变量就“绑定”在这个区域，不再受外部影响
- let对这个块从一开始就形成了封闭的作用域，凡是在声明之前使用该变量，就会报错

```
var tmp = 123;  
if(true){  
    tmp = "abc";//Reference Error  
    let tmp;  
}
```

typeof 将不再是绝对安全的操作（参见实例），养成良好习惯，在使用变量之前定义变量

► Uncaught ReferenceError: tmp is not defined

- var可以声明重名变量，let和const则不可以

# 总结

---

- ES5中的var及其缺陷
- ES6中的let与const
- let与const的重要特性





Thank You !



河北师范大学软件学院  
Software College of Hebei Normal University