

登录后您可以享受以下权益：

 免费复制代码

 和博主大V互动

 下载海量资源

 发动态/写文章/加入社区

立即登录



Milk595

关注

# 【ES6】箭头函数this指向问题【看这一篇就够啦！】

原创 Milk595 于 2020-08-10 14:57:14 发布 阅读量2.7k 收藏 18 点赞数 4  
分类专栏: ES6+ 文章标签: es6



ES6+ 专栏收录该内容

1 订阅 12 篇文章

## 箭头函数 this指向

箭头函数有4个使用注意点。

- (1) 函数体内的 this 对象，就是定义时所在的对象，而不是使用时所在的对象。
- (2) 不可以当作构造函数，也就是说，不可以使用 new 命令，否则会抛出一个错误。
- (3) 不可以使用 arguments 对象，该对象在函数体内不存在。如果要用，可以用 rest 参数代替。
- (4) 不可以使用 yield 命令，因此箭头函数不能用作 Generator 函数。

上面四点中，第一点尤其值得注意。this 对象的指向是可变的，但是在箭头函数中，它是固定的。

```
1 function foo() {
2   setTimeout(() => {
3     console.log('id:', this.id);
4   }, 100);
5 }
6
7 var id = 21;
8
9 foo.call({ id: 42 });
10 // id: 42
```

上面代码中，setTimeout 的参数是一个箭头函数，这个箭头函数的定义生效是在 foo 函数生成时，而它的真正执行要等到 100 毫秒后。如果是普通函数，this 应该指向全局对象 window，这时应该输出 21。但是，箭头函数导致 this 总是指向函数定义生效时所在的对象（本例是 {id: 42}），所以输出

箭头函数可以让 setTimeout 里面的 this，绑定定义时所在的作用域，而不是指向运行时所在的作用域。下面是另一个例子。

```
1 function Timer() {
2   this.s1 = 0;
3   this.s2 = 0;
4   // 箭头函数
5   setInterval(() => this.s1++, 1000);
6   // 普通函数
7   setInterval(function () {
8     this.s2++;
9   }, 1000);
10 }
11
```

上面代码中，Timer 函数内部设置了两个定时器，分别使用了箭头函数和普通函数。前者的 this 绑定定义时所在的作用域（即 Timer 函数），后者的运行时所在的作用域（即全局对象）。所以，3100 毫秒之后，timer.s1 被更新了 3 次，而 timer.s2 一次都没更新。

箭头函数可以让 this 指向固定化，这种特性很有利于封装回调函数。下面是一个例子，DOM

```
1 var handler = {
2   id: '123456',
3
4   init: function() {
5     document.addEventListener('click',
6       event => this.doSomething(event.type)
7     ),
8
```



Milk595 关注

登录后您可以享受以下权益：

免费复制代码

和博主大V互动

下载海量资源

发动态/写文章/加入社区

```

9 | doSomething: function(type) {
10 |     console.log('Handling ' + type + ' for ' + this.id);
11 | }
12 | };

```

上面代码的 `init` 方法中，使用了箭头函数，这导致这个箭头函数里面的 `this`，总是指向 `handler` 对象。否则，回调函数运行时，`this.doSomething()` 错，因为此时 `this` 指向 `document` 对象。

`this` 指向的固定化，并不是因为箭头函数内部有绑定 `this` 的机制，实际原因是箭头函数根本没有自己的 `this`，导致内部的 `this` 就是外层代码块的 `this`。因为它没有 `this`，所以也就不能用作构造函数。

所以，箭头函数转成 ES5 的代码如下。

```

1 | // ES6
2 | function foo() {
3 |     setTimeout(() => {
4 |         console.log('id:', this.id);
5 |     }, 100);
6 | }
7 |
8 | // ES5
9 | function foo() {
10 |     var _this = this;
11 |
12 |     setTimeout(function () {
13 |         console.log('id:', _this.id);
14 |     }, 100);
15 | }

```

上面代码中，转换后的 ES5 版本清楚地说明了，箭头函数里面根本没有自己的 `this`，而是引用外层的 `this`。

请问下面的代码之中有几个 `this`？

```

1 | function foo() {
2 |     return () => {
3 |         return () => {
4 |             return () => {
5 |                 console.log('id:', this.id);
6 |             };
7 |         };
8 |     };
9 | }
10 |
11 | var f = foo.call({id: 1});
12 |
13 | var t1 = f.call({id: 2})(); // id: 1
14 | var t2 = f().call({id: 3})(); // id: 1
15 | var t3 = f()().call({id: 4})(); // id: 1

```

上面代码之中，只有一个 `this`，就是函数 `foo` 的 `this`，所以 `t1`、`t2`、`t3` 都输出同样的结果。因为所有的内层函数都是箭头函数，都没有自己的 `this`，它们的 `this` 其实都是最外层 `foo` 函数的 `this`。

除了 `this`，以下三个变量在箭头函数之中也是不存在的，指向外层函数的对应变量的： `arguments`、`super`、`new.target`。

```

1 | function foo() {
2 |     setTimeout(() => {
3 |         console.log('args:', arguments);
4 |     }, 100);
5 | }
6 |
7 | foo(2, 4, 6, 8)
8 | // args: [2, 4, 6, 8]

```

上面代码中，箭头函数内部的变量 `arguments`，其实是函数 `foo` 的 `arguments` 变量。

另外，由于箭头函数没有自己的 `this`，所以当然也就不能用 `call()`、`apply()`、`bind()` 这些



Milk595

关注

登录后您可以享受以下权益：

 免费复制代码

 和博主大V互动

 下载海量资源

 发动态/写文章/加入社区

```
1 (function() {
2   return [
3     (() => this.x).bind({ x: 'inner' })()
4   ];
5 }).call({ x: 'outer' });
6 // ['outer']
```

上面代码中，箭头函数没有自己的 `this`，所以 `bind` 方法无效，内部的 `this` 指向外部的 `this`。

长期以来，JavaScript 语言的 `this` 对象一直是一个令人头痛的问题，在对象方法中使用 `this`，必须非常小心。箭头函数“绑定”`this`，很大程度上解

## 不适用场合

由于箭头函数使得 `this` 从“动态”变成“静态”，下面两个场合不应该使用箭头函数。

第一个场合是定义对象的方法，且该方法内部包括 `this`。

```
1 const cat = {
2   lives: 9,
3   jumps: () => {
4     this.lives--;
5   }
6 }
```

上面代码中，`cat.jumps()` 方法是一个箭头函数，这是错误的。调用 `cat.jumps()` 时，如果是普通函数，该方法内部的 `this` 指向 `cat`；如果写成上面函数，使得 `this` 指向全局对象，因此不会得到预期结果。这是因为对象不构成单独的作用域，导致 `jumps` 箭头函数定义时的作用域就是全局作用域。

第二个场合是需要动态 `this` 的时候，也不应使用箭头函数。

```
1 var button = document.getElementById('press');
2 button.addEventListener('click', () => {
3   this.classList.toggle('on');
4 });
```

上面代码运行时，点击按钮会报错，因为 `button` 的监听函数是一个箭头函数，导致里面的 `this` 就是全局对象。如果改成普通函数，`this` 就会动态指按钮对象。

另外，如果函数体很复杂，有许多行，或者函数内部有大量的读写操作，不单纯是为了计算值，这时也不应该使用箭头函数，而是要使用普通函数，提高代码可读性。

this 指向详细解析（箭头函数）

weixin\_30568591的

本文地址：http://www.cnblogs.com/dongcanliang/p/7054176.html 前言 这个 指向问题是入坑前端必须了解知识点，现在迎来了ES6时代，因为箭头函数的出现，所以感觉

箭头函数：this指向问题

weixin\_69188183的

箭头函数的this指向是静态的，始终指向函数声明时所在作用域下的this值。一篇解决你所有困惑。

JSthis指向问题大全(普通函数,箭头函数,匿名函数)\_obj中有普通函数和...

function Foo(){//Bconsole.log(this);//Foo {this.x=10;console.log(this);//Foo { x: 10 }}Foo.prototype.getX=function(){//Econsole.log(this);//Foo {x: 10}console.log(this.x);//10}

箭头函数中的this指向问题详解\_箭头函数的this指向问题

上面代码的init方法中,使用了箭头函数,这导致这个箭头函数里面的this,总是指向handler对象。否则,回调函数运行时,this.doSomething这一行会报错,因为此时this指向docu

关于ES6箭头函数中的this问题

ES6中新增了箭头函数这种语法,箭头函数以其简洁性和方便获取this的特性,接下来通过本文给大家分享ES6箭头函数中的this,感兴趣的朋友一起看看吧

前端每日一练：为什么 computed 不支持异步？computed 的机制和缓存 最新发布  
为什么 computed 不支持异步？computed 的机制和缓存

JS的this指向问题(史上最全)\_js this指向问题

这里说的this指向就是上文中代码中的例子,有兴趣可以自行多尝试一下此场景中的this 4.new 当使用 new 来

this 指向详细解析(箭头函数)\_箭头函数this指向

在非箭头函数下, this 指向调用其所在函数的对象,而且是离谁

ES6对象方法声明对象时, 箭头函数this的指向问题



Milk595

关注

登录后您可以享受以下权益：

免费复制代码

和博主大V互动

下载海量资源

发动态/写文章/加入社区 可能

基础 没有this、super、arguments 和 new.target 绑定 不能通过 new 关键字调用 没有 prototype 不可以绑定this 的绑定 不支持argument 不支持重复命名参数 疑问 // 第一

JavaScript ES6箭头函数使用指南

JavaScript ES6箭头函数是一个来自ECMAScript 2015（又称ES6）的全新特性,有传闻说，箭头函数的语法=>，是受到了CoffeeScript 的影响，并且它与CoffeeScript中的

1分钟搞懂箭头函数this指向问题\_表达式声明的箭头函数在点击事件里执...

在网上看到很多关于箭头函数this指向的博客和问题解答,但是基本写的都不准确,有的甚至是错误的,在其他博客网站得到的答案基本就是“箭头函数在定义时this指向谁那么:

箭头函数中的this指向问题\_ts 箭头函数中的this报错

箭头函数中的this是在定义函数的时候绑定,而不是在执行函数的时候绑定的。 箭头函数中this指向的固定化,并不是因为箭头函数内部有绑定this的机制,实际原因是箭头函数

Js中的this指向问题（普通函数和箭头函数）

前端与计算机相关知识的分享，博主的qq523235

一：普通函数的this指向 \*普通函数中this是动态的，当我们调用函数的时候确定的，调用方式的不同决定了this的指向不同，一般指向我们函数的调用者。（谁调用指向谁

关于箭头函数中的this的指向 热门推荐

xu4321083的

普通函数中的this:1. this总是代表它的直接调用者(js的this是执行上下文), 例如 obj.func ,那么func中的this就是obj2.在默认情况(非严格模式下,未使用 'use strict'),没找到直接

箭头函数中的this

weixin\_34375251的

使用注意点 箭头函数有几个使用注意点。（1）函数体内的this对象，就是定义时所在的对象，而不是使用时所在的对象。（2）不可以当作构造函数，也就是说，不可以

JS核心：用箭头函数解决this指向问题

输出与

此文主要分析如何用箭头函数解决匿名函数中的this的指向不清的问题。帮助您更加理解this和熟悉箭头函数的使用场景。

Vue基础知识

qq\_47433906的

Vue基础知识

箭头函数与this指向详解

web\$-/

箭头函数以及箭头函数中this的指向问题

关于箭头函数中this指向问题（一步到位，不用背）

qq\_42931285的

一句话概括：箭头函数中没有this，若想打印的话它只能沿着作用域链去找。例：<script> var name = "周冬雨"; let obj={ name:"东东", say:()=>{ //箭头函数中无this,只能

箭头函数里的this指向问题

xiaowuwjw的

箭头函数的基础指向问题，简单易懂！！

深刻理解箭头函数的this指向问题！

qq\_58693650的

两个关于箭头函数this指向容易弄错的关键点:箭头函数的this指向外层词法上下文；箭头函数的this在定义时绑定！加深理解哦~

箭头函数this的指向

weixin\_57959921的

箭头函数this的指向

箭头函数和this指向问题

m0\_49456900的

①结构: const/let 函数名 = (参数) => {函数体}; ②一般函数转为箭头函数步骤: 声明形式->函数表达式形式->箭头函数 function fun() {}; ->const/let add = function() {}; -> c

箭头函数及其this的指向

qq\_53841687的

目录一、箭头函数1、基础语法 2、箭头函数简写 二、箭头函数的 this1、全局函数下的 this2、对象方法里面的 this3、构造函数的this4、 call()、apply()、bind()调用时的

vue 为什么computer为什么不可以用异步，watch却可以

weixin\_45773252的

在Vue中，computed属性是计算属性，它的值会根据它所依赖的属性动态计算出来，并且计算结果会被缓存起来，只有当依赖的属性发生变化时才会重新计算。如果在cor

es6 箭头函数

ES6 中的箭头函数是一种更简洁的函数语法，它可以更方便地定义匿名函数。...4. 箭头函数没有自己的 `this` 值，它会继承外层作用域中的 `this` 值，因此可以避免传统函

“相关推荐”对你有帮助么？



非常没帮助



没帮助



一般



有帮助



非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心

家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照

©1999-2024北京创新乐知网络技术有限公司



Milk595

码龄6年



暂无认证

247 5万+ 98万+ 50万+ 原创 周排名 总排名 访问 等级

4433 177 300 100 1252 积分 粉丝 获赞 评论 收藏



Milk595

关注

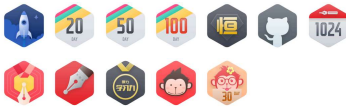
登录后您可以享受以下权益：

免费复制代码

和博主大V互动

下载海量资源

发动态/写文章/加入社区



私信

关注



搜博主文章

## 热门文章

【TS基础】类型"Window & typeof globalThis"上不存在属性"\$loading" 🔍  
35262

【JS基础】e.preventDefault() 阻止默认操作 🔍 30755

【JS实战】删除数组最后一个元素 🔍  
14115

【Umi】Layout 布局的实现!!! 🔍  
11604

【前端面试】Vue 你的接口请求一般放在哪个生命周期中? 🔍 9703

## 分类专栏

跨域	10篇
前端面试	55篇
前端项目实战	8篇
JavaScript	57篇
工具类	16篇



Milk595

关注

登录后您可以享受以下权益:

📄 免费复制代码

👤 和博主大V互动

📁 下载海量资源

✍️ 发动态/写文章/加入社区



### 最新评论

应届毕业生，只会抄代码，该怎么办？

城铁: 看来是我抄得不够多

【Git】一张脑图带你掌握Git命令

艾米哈伯十三世: 不是高清啊

【Vue】何为 \$parent？它的使用场景是...

xinweiweixin: “当前组件树的根 Vue 实例”

不对

【Echarts】详解 折线图/面积图!!! 快...

阿里数据专家: AIGC ChatGPT，BI商业智能，可视化Tableau, PowerBI, FineRep ...

【ECharts】无法找到模块“echarts”的声...

weixin\_48280601: 槽 写在具体那个地方呀就不能多截一张图

您愿意向朋友推荐“博客详情页”吗？



强烈不推荐 不推荐 一般般 推荐 强烈推荐

### 最新文章

【2021】关于JS数据类型的题目

【工具类】记录一些常用的正则表达式

【2021】最新的ECMAScript标准定义了8种数据类型

2021年 7篇

2020年 380篇



SHOP NOW

SHOP NOW

SHOP NOW

SHO

SHOP NOW

SHO

登录后您可以享受以下权益：

免费复制代码

和博主大V互动

下载海量资源

发动态/写文章/加入社区



Milk595

关注