

匿名回调函数中的this指向

 **jamine**  50 发布于 2018-04-10

在平时对`setTimeout` `addEventListener`这种函数的使用，都会传入一个匿名的callback函数

如：

```
1.
window.setTimeout(function() {
  console.log(this)
}, 1000)

或者
2.
var el = document.getElementById('wrapper')
el.addEventListener('click', function() {
  console.log(this)
}, false)
```

上述情况1中`this`会指向`setTimeout`函数的caller -> window对象
而情况2中`this`也会指向`addEventListener`函数的caller -> wrapper这个对象

但是我自己类似于`window.setTimeout`这样创建了一个对象

```
var a = {
  b: 'b in a',
  c: function (callback) {
    callback()
  }
}

// 调用a对象的c函数，传入匿名函数作为参数
a.c(function() {
  // 本以为this是指向a的，会输出字符串'b in a'，实际此时this指向window
  console.log(this.b)
})
```

我原本以为会类似于`window.setTimeout el.addEventListener`那样，`this`会指向`.`(点号)之前的对象。

然后我改了一下对象a

```
var a = {
  b: 'b in a',
  c: function (callback) {
    callback.bind(this)()
  }
}
```

这个时候`this`才指向a。

那么问题来了：

- 1.像这种匿名函数传参的用法，为什么使用我自己定义的对象和浏览器提供的api产生的效果不一样呢？这种类型的`this`的指向应该如何更好的理解？
- 2.是不是像`setTimeout`、`addEventListener`这种系统api，它的内部实现就帮我们去把`this` bind给了调用这个方法的对象，如`setTimeout`中就有`callback.bind(window)`和`addEventListener`就有`callback.bind(el)`？

有没有各路大神可以解答一下，小弟感激不尽。

[javascript](#) [es5](#)

关注 6

收藏 2

赞

回复 · 阅读 1.9k

4 个回答

得票数

最新



改名字很伤神 7.9k 发布于 2018-04-10

✓ 已被采纳

关于this的指向的优先级

1. `new Foo()` 绑定新对象
2. `bind/call/apply` 绑定指定的对象
3. 绑定上下文

```
var a = {
  b: function () {
    //this -> a
  }
}
```

4. 默认全局

```
var a = {
  b: 'b in a',
  c: function c2(callback) {
    callback()
  }
}

// 调用a对象的c 函数，传入匿名函数作为参数
a.c(function c1() {
  //c1里的this是什么，首先排除规则1,2，其次在c2函数作用域执行，排除3，使用默认规则。
  console.log(this.b)
})
```

那么楼主的代码，为了描述我命了名。

补充 addEventListener 的说明

以下为一段为了浏览器兼容而提供的polyfill代码
这段代码结合上述规则可以很清楚的说明addEventListener的this

```
while (counter<eventListeners.length) {
  var eventListener=eventListeners[counter];
  if (eventListener.object==this && eventListener.type==type && eventListener.listener==listener) {
    if (type=="DOMContentLoaded") {
      this.detachEvent("onreadystatechange",eventListener.wrapper);
    } else {
      this.detachEvent("on"+type,eventListener.wrapper);
    }
    eventListeners.splice(counter, 1);
    break;
  }
  ++counter;
}
};
Element.prototype.addEventListener=addEventListener;
Element.prototype.removeEventListener=removeEventListener;
if (HTMLDocument) {
  HTMLDocument.prototype.addEventListener=addEventListener;
  HTMLDocument.prototype.removeEventListener=removeEventListener;
}
if (Window) {
  Window.prototype.addEventListener=addEventListener;
  Window.prototype.removeEventListener=removeEventListener;
}

}
```

[addEventListener polyfill 链接](#)

 赞



回复

jamine: 首先谢谢大神解答，但是还是想请问一下，addEventListener()中的this为什么指向了调用这个方法的元素，而不是默认规则指向window？

 · [回复](#) · 2018-04-20

改名字很伤神: 因为addEventListener属于Web Api，首先这是一个回调函数，这个回调函数作为参数传进去到底会被谁执行呢，这个我无从得知。但是如果让我写的话，我会利用bind，因为我写一个addEventListener函数，我需要所有dom都能用，那肯定需要显示改变内部的作用域的，一般我倾向于这种猜测，但这也仅仅是一种猜测。

 · [回复](#) · 2018-04-20

改名字很伤神: 添加了一段polyfill代码，可以说明this的指向，但这是兼容性代码，不能确定真正的实现方式，但可以确定这是Web Api的定义。

 · [回复](#) · 2018-04-20

共 4 条评论



YoRHaNo2TypeB  102 发布于 2018-04-10

你可以把this当做function的一个隐藏参数，相当于

```
function(_this, otherArgs){

}
```

事实上基本所有语言也都是这么处理的。所以当this不对的时候，可以用bind显性地传递这个参数。

检查一个function是否绑定了this这个参数可以用下面的方法：

```
// ES5
function isBindable(func) {
  return func.hasOwnProperty('prototype');
}

// ES6
const isBindable = func => func.hasOwnProperty('prototype');
```

事件监听那里可能是做了特殊处理，毕竟JS是个设计糟糕的语言（哈）

 赞 1



[上官元恒](#)  7.9k 发布于 2018-04-10

[MDN bind](#)

 赞



回复


 [安静等天亮](#)  27 发布于 2018-04-10

匿名函数的this默认都是指向window。对于元素事件监听的函数里面this指向元素自身，才属于是特殊情况。

 赞






回复

[jamine](#): 首先谢谢大神解答。事件监听的this只能当作是特殊处理来理解吗？没有一种解释可以帮助理解吗？
 • [回复](#) • 2018-04-20

撰写回答

你尚未登录，登录后可以

-  和开发者交流问题的细节
-  关注并接收问题和回答的更新提醒
-  参与内容的编辑和改进，让解决方法与时俱进

[注册登录](#)

你知道吗？

不要站着调试程序，那会使得你的耐心减半，你需要的是全神贯注。

[注册登录](#)

相似问题

[为什么settimeout中的函数this指向window？](#)

{代码...} [为什么settimeout中的函数this指向window？而箭头函数this指向Object 是因为settimeout是window对象的方法还是说当做直接调用一个...](#)

[4 回答](#) • [3.3k 阅读](#) ✓ 已解决

箭头函数this指向的问题

箭头函数内部的this是指向定义时所在的对象，不是执行时所在的对象。按照这个道理的话，setTimeout内部的函数不管是不是箭头函数应该都是指...

17 回答 · 2k 阅读 ✓ 已解决

axios回调函数中的this为何是undifine?

关于axios回调函数this指向的问题，网上的大部分文章只是推荐用箭头函数取代普通函数解决，但是却没有解释为什么this会指向undefine，在我的...

3 回答 · 1.3k 阅读 ✓ 已解决

JS 怎么让回调函数内部this指向undefined?

我的目标是让回调函数内容的this指向undefined，但call方法传入null或undefined时无需，回调函数内部this指向任然是window对象，烦请大神指教...

4 回答 · 1.6k 阅读

关于js this指向的一个问题

以上代码中，getname中的this是指向a的我能理解，但是它里面的b函数中的this指向window是怎么理解呢，为什么b函数是由window调用呢当函数...

3 回答 · 686 阅读 ✓ 已解决

js回调函数中的this和外边的this是否为同一个

{代码...}js回调函数中的this和外边的this是否指向同一个对象 已解决不是同一个对象 外层的指向当前选中元素，回调函数中的为触发回调的元素如...

1 回答 · 1.4k 阅读 ✓ 已解决

setTimeout + dom事件 中this指向问题

求教一个诡异的 this 指向问题, 第一点: fn() 调用时, 会转为this.fn(), 第二点: fn 是普通函数, 内部的 this 指向调用者, 所以内部的 this 指向 dom 才对,...

2 回答 · 334 阅读

JavaScript中this指向的问题...

可是还有个构造函数,构造函数不使用new来调用的话，里面的this也是指向全局的..., 使用new来调用则this指向的是new出来的实例，这些我都了解...

10 回答 · 2.3k 阅读 ✓ 已解决

找不到问题？ [创建新问题](#)

宣传栏

带你从零到一全面掌握前端「新宠」

TypeScript从入门到实践「2020 版」

Node.js高级实战·带你搭建动漫网站

紧贴业务，扩大前端职能，顺利迈入全栈

前端性能优化设计12问

前端性能优化,面试的必问题目

产品

[热门问答](#)

[热门专栏](#)

[热门课程](#)

[最新活动](#)

[技术圈](#)

[酷工作](#)

课程

[Java 开发课程](#)

[PHP 开发课程](#)

[Python 开发课程](#)

[前端开发课程](#)

[移动开发课程](#)

资源

[每周精选](#)

[用户排行榜](#)

[徽章](#)

[帮助中心](#)

[声望与权限](#)

[社区服务中心](#)

合作

[关于我们](#)

[广告投放](#)

[职位发布](#)

[讲师招募](#)

[联系我们](#)

[合作伙伴](#)

关注

[产品技术日志](#)

[社区运营日志](#)

[市场运营日志](#)

[团队日志](#)

[社区访谈](#)

条款

[服务条款](#)

[隐私政策](#)

[下载 App](#)

[建议反馈](#)

Copyright © 2011-2021 SegmentFault. 当前呈现版本 21.07.22
[浙ICP备15005796号-2](#) [浙公网安备33010602002000号](#) ICP 经营许可 浙B2-20201554
杭州堆栈科技有限公司版权所有

