

# this绑定

2020年03月18日, 星期三 14:53

## 1.this

this 是被自动定义在所有函数的作用域中一个关键字

this 是在函数**运行期间**绑定，与函数定义无关，与执行有关，与函数调用方式有关

this 作用

复用代码，为函数自动引用合适的上下文对象

提供了一个更加优雅而简便的方式来隐式传递一个对象引用

## 2.this绑定规则

### ● 函数调用形式 (4种)

- 作为函数直接调用 —————> 默认绑定
  - 定时器函数
  - 立即执行函数
- 作为对象方法调用 —————> 隐式绑定
  - 事件函数
- 作为构造函数调用 —————> new 绑定
- 通过 call/apply 间接调用 —————> 显示绑定

## 3.默认绑定

普通函数调用：

```
var a = 2;
function foo() {
  console.log(this); // Window {}
  console.log(this.a); // 2
}
foo(); // 2
```

函数嵌套时，需注意 this 不进行作用域传递

```
function show() {
  var a = 200;
  console.log(this.a); // 100
}
var a = 100;
setTimeout(show, 3000);
```

```
<button id="btn">按钮</button>
<script>
  var btn = document.getElementById("btn");
  var a = 100;
  btn.onclick = function() {
    var a = 300;

    function show() {
      var a = 200;
      console.log(this.a); // ??? 100
      console.log(this.id); // ??? undefined
    }
    setTimeout(show, 3000);
  }
</script>
```

函数嵌套时，this绑定：

- 变量 `_this`、`that`、`self` 锁定 `this` // (软绑定)
- `bind()` 锁定

```
btn.onclick = function() {
  var a = 300;
  //变量 this, that, self 锁定 this (备份)
  var _this = this;

  function show() {
    var a = 200;
    console.log(_this.a);
    console.log(_this.id);
  }

  setTimeout(show, 3000);
};
```

```
btn.onclick = function() {
  var a = 300;
  console.log(this);

  function show() {
    var a = 200;
    console.log(this.a);
    console.log(this.id);
  }

  setTimeout(show.bind(this), 3000);
};
```

#### 4.隐式绑定

调用位置是否有上下文对象

- 函数作为对象的方法调用时，函数上下文指向这个对象

```
var a = 100;
var obj = {
  a: 300,
  fun: function() {
    console.log(this.a);
  }
};
obj.fun(); //300
```

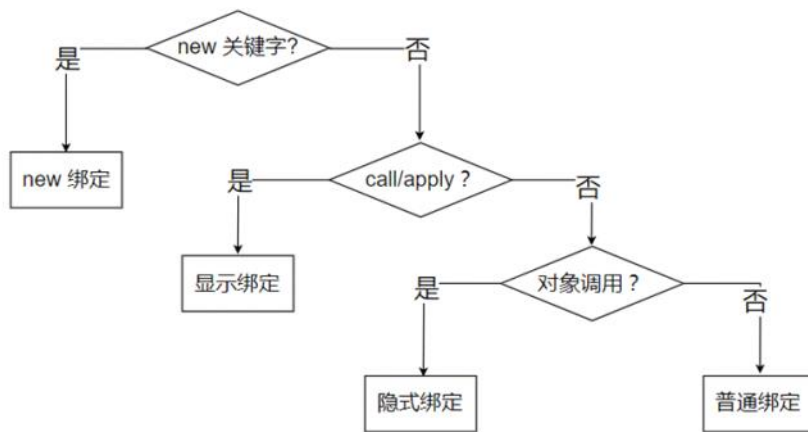
```
var a = 100;
var obj = {
  a: 300,
  fun: function() {
    console.log(this.a);
  }
};
var fun = obj.fun;
fun(); //??? 100
obj.fun(); //300
```

- 数组中存放函数，被数组索引调用，`this`上下文指向这个数组

```
function fun() {
  console.log(this.length);
}
var length = 10;
var arr = [100, 200, fun];
fun(); //10
arr[2](); //3
```

#### 5.绑定优先级

- new 绑定 > 显式绑定 > 隐式绑定 > 默认绑定
- 判断 `this` 的顺序



eg:

```

var obj = {
  name: "Tom"
};
function Fun(name) {
  this.name = name;
  console.log(this.name);
}
var FunBind = Fun.bind(obj);
var instance = new FunBind("Lily");
console.log(instance);
//new优先级比显示绑定高
  
```

6.

```

var number = 1;
var obj = {
  number: 2, //3
  showNumber: function() {
    this.number = 3; //this: obj
    (function() {
      console.log(this.number); //1 this:window
    })();
    console.log(this.number); //3 this:obj
  }
};
obj.showNumber();
  
```