

认识上下文

什么是上下文

这是非常好的习惯，值得表扬

什么是上下文

垃圾分类，**这**是非常好的习惯，值得表扬
随手关灯，**这**是非常好的习惯，值得表扬
遛狗栓绳，**这**是非常好的习惯，值得表扬
课后复习，**这**是非常好的习惯，值得表扬
早睡早起，**这**是非常好的习惯，值得表扬

函数的上下文

- ◆ 函数中可以使用**this**关键字，它表示**函数的上下文**
- ◆ 与中文中“这”类似，函数中的**this**具体指代什么**必须通过调用函数时的“前言后语”来判断**

函数中的this

```
var xiaoming = {  
  nickname : '小明',  
  age      : 12,  
  sayHello : function () {  
    console.log('我是' + this.nickname + ', 我' + this.age + '岁了');  
  }  
};  
  
xiaoming.sayHello();
```

我是小明，我12岁了

函数中的this

```
var xiaoming = {  
  nickname : '小明',  
  age      : 12,  
  sayHello : function () {  
    console.log('我是' + this.nickname + ', 我' + this.age + '岁了');  
  }  
};  
  
var sayHello = xiaoming.sayHello; → 将函数“提”出来，单独存为变量  
sayHello(); → 直接圆括号调用这个函数，而不是对象打点调用了
```

我是undefined，我undefined岁了

函数的上下文由调用方式决定

- ◆ 同一个函数，用不同的形式调用它，则函数的上下文不同

情形1：对象打点调用函数，函数中的this指代这个打点的对象

```
xiaoming.sayHello();
```

情形2：圆括号直接调用函数，函数中的this指代window对象

```
var sayHello = xiaoming.sayHello;  
sayHello();
```

函数的上下文由调用方式决定

```
var obj = {  
  a: 1,
```

不知道

```
    console.log(this.a + this.b);  
  }  
};
```

这两个this指代什么？

函数的上下文由调用方式决定

```
var obj = {
```

函数只有被调用，
它的上下文才能被确定

```
};
```

这两个this指代什么？

函数的上下文由调用方式决定

```
var obj = {  
  a: 1,  
  b: 2,  
  fn: function () {  
    console.log(this.a + this.b);  
  }  
};  
  
obj.fn();
```

函数的上下文由调用方式决定

```
var obj = {  
  a: 1,  
  b: 2,  
  fn: function () {  
    console.log(this.a + this.b);  
  }  
};  
var fn = obj.fn;  
fn();
```

谢谢！