**《深入理解JavaScript》第17章相关内容**

**与第八章一样**

**《你不知道的JavaScript(上卷)》第二部分**

第 1 章

关于**this**

this 关键字是 JavaScript 中最复杂的机制之一。它是一个很特别的关键字，被自动定义在

所有函数的作用域中。

第 2 章

**this**全面解析

判断this

现在我们可以根据优先级来判断函数在某个调用位置应用的是哪条规则。可以按照下面的

顺序来进行判断：

1. 函数是否在 new 中调用（new 绑定）？如果是的话 this 绑定的是新创建的对象。

var bar = new foo()

2. 函数是否通过 call、apply（显式绑定）或者硬绑定调用？如果是的话，this 绑定的是

指定的对象。

var bar = foo.call(obj2)

3. 函数是否在某个上下文对象中调用（隐式绑定）？如果是的话，this 绑定的是那个上

下文对象。

var bar = obj1.foo()

4. 如果都不是的话，使用默认绑定。如果在严格模式下，就绑定到 undefined，否则绑定到

全局对象。

var bar = foo()

如果要判断一个运行中函数的 this 绑定，就需要找到这个函数的直接调用位置。找到之后

就可以顺序应用下面这四条规则来判断 this 的绑定对象。

1. 由 new 调用？绑定到新创建的对象。

2. 由 call 或者 apply（或者 bind）调用？绑定到指定的对象。

3. 由上下文对象调用？绑定到那个上下文对象。

4. 默认：在严格模式下绑定到 undefined，否则绑定到全局对象。

一定要注意，有些调用可能在无意中使用默认绑定规则。如果想“更安全”地忽略 this 绑

定，你可以使用一个 DMZ 对象，比如 ø = Object.create(null)，以保护全局对象。

ES6 中的箭头函数并不会使用四条标准的绑定规则，而是根据当前的词法作用域来决定

this，具体来说，箭头函数会继承外层函数调用的 this 绑定（无论 this 绑定到什么）。这

其实和 ES6 之前代码中的 self = this 机制一样。

第 3 章

对象

对象是 JavaScript 的基础。在 JavaScript 中一共有六种主要类型（术语是“语言类型”）：

• string

• number

• boolean

• null

• undefined

• object

JavaScript 中还有一些对象子类型，通常被称为内置对象。

• String

• Number

• Boolean

• Object

• Function

• Array

• Date

• RegExp

• Error