# 模块化

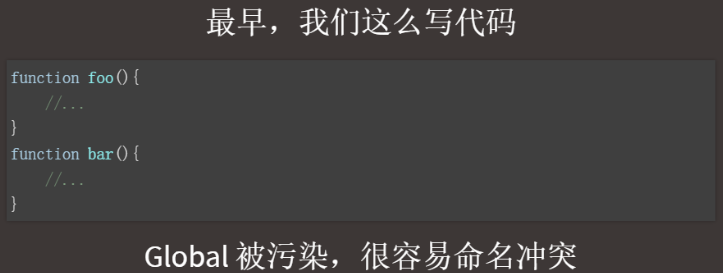
## 什么是模块/模块化?

将一个复杂的程序依据一定的规则(规范)封装成几个块(文件), 并进行组合在一起块的内部数据/实现是私有的, 只是向外部暴露一些接口(方法)与外部其它模块通信

## 模块化进展史

### 最初代码

示例



不足

**\*** 说明:  
  **\*** 全局函数模式: 将不同的功能封装成不同的全局函数  
  **\*** 问题: Global被污染了, 很容易引起命名冲突

### namespace模式

示例



不足

**\*** namespace模式: 简单对象封装  
**\*** 作用: 减少了全局变量  
**\*** 问题: 不安全

### IIFE模式

示例

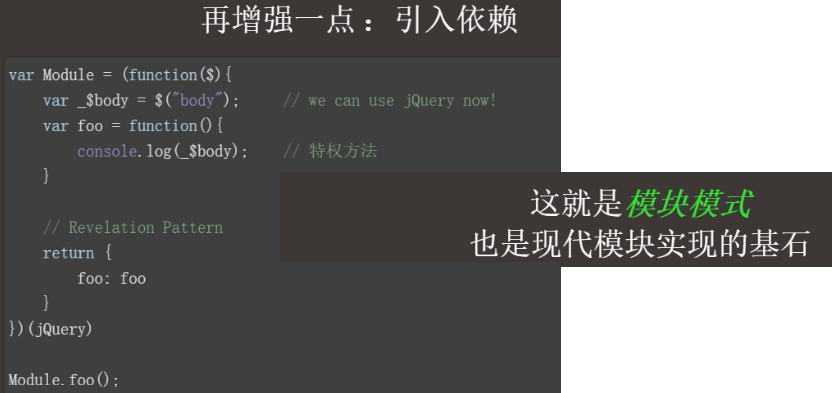


不足

说明:  
 **\*** IIFE模式: 匿名函数自调用(闭包)  
 **\*** IIFE : immediately-invoked function expression(立即调用函数表达式)  
 **\*** 作用: 数据是私有的, 外部只能通过暴露的方法操作  
 **\*** 问题: 如果当前这个模块依赖另一个模块怎么办?

### IIFE模式增强

示例



不足

**\*** IIFE模式增强 : 引入依赖  
**\*** 这就是现代模块实现的基石

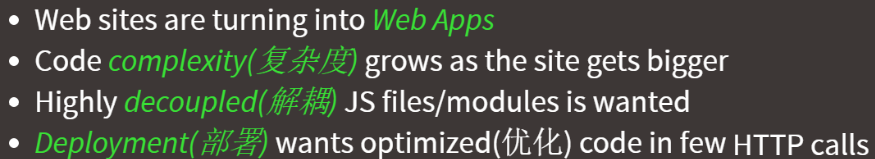
## 模块化的好处

避免命名冲突(减少命名空间污染)

更好的分离, 按需加载

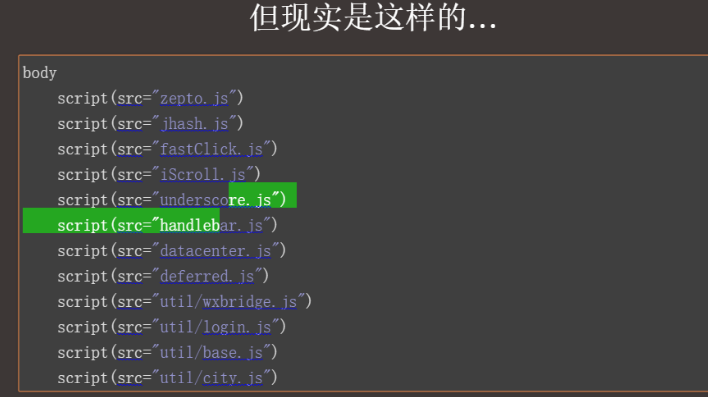
更高复用性

高可维护性



## 不足

### 页面引入加载script



### 问题

请求过多

依赖模糊

难以维护