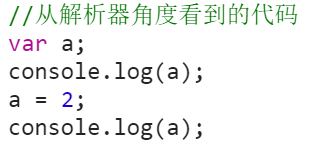
**第九章 JS预解析**

**JS解析及执行简介**

JS脚本语言（非提前编译，由解析器边解析边执行）  
- 区别于C/C++编译成二进制和Java/C#编译成字节码（运行在跨平台虚拟机上）的解析执行

JS代码案例（思考：是否会报错，区别于其他语言）

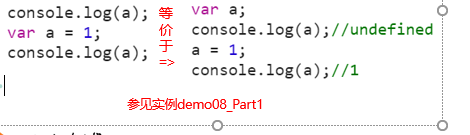
JS的解析和执行过程  
- 全局预解析阶段（全局变量和函数声明前置）  
- 全局顺序执行阶段（变量赋值、函数调用等操作）  
- 当遇到函数调用时，在执行函数内代码前，进行函数范围内的预解析  
- 当存在函数嵌套时，以此类推，会进行多次函数预解析

**// undefined 2** 

**JS预解析（声明提升）**

* 预解析主要工作（变量声明和函数声明提升）  
  - 解析器在执行代码前的进行代码扫描（var、function）  
  - 将变量和函数声明在当前作用域（全局、函数）内进行提升

变量声明提升案例

其余见代码

**图片包含 文字

描述已自动生成**

**预解析与作用域**

* 变量的作用域是指变量在何处可以被访问到  
  - JS采用的是静态词法作用域，代码完成后作用域链就已形成，与代码的执行顺序无关
* 全局变量与局部变量  
  - 全局变量：拥有全局作用域的变量（JS代码中任何地方都可以访问）  
  - 全局变量是跨域了所有函数自身作用域的自由变量，可以在函数内和函数外直接访问  
  - 局部变量：函数内声明的变量，只在函数体内有定义，作用域是局部性的  
  - 在函数外不能直接访问函数的局部变量，但可以通过闭包来访问  
  - 函数内访问同名变量时，局部变量会覆盖全局变量
* ES5中无块作用域（ES5作用域缺陷及解决办法参见IIFE）  
  - 全局作用域、函数作用域、 ES5中可以使用函数立即执行表达式来模拟块作用域