1. 作用域

**传统编译**

三个步骤：

分词/词法分析：

这个过程将字符串分解成词法单元，如var a=2，分解为词法单元：var、a、=、2、；。

解析/语法分析：

这个过程将词法单元流转换成一个由元素逐级嵌套所组成的代表了程序语法结构的树，被称为抽象语法树。

代码生成：

将AST转换为可执行代码的过程。

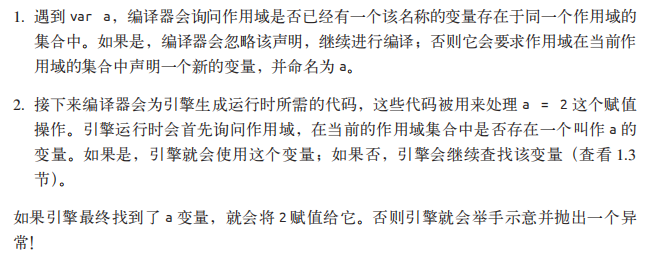
**角色表**

引擎：从头到尾负责整个javascript程序的编译及执行过程

编译器：负责语法分析及代码生成

作用域：负责收集并维护由所有声明的标识符（变量）组成的一系列查询，并实施一套非常严格的规则，确定当前执行的代码对这些标识符的访问权限

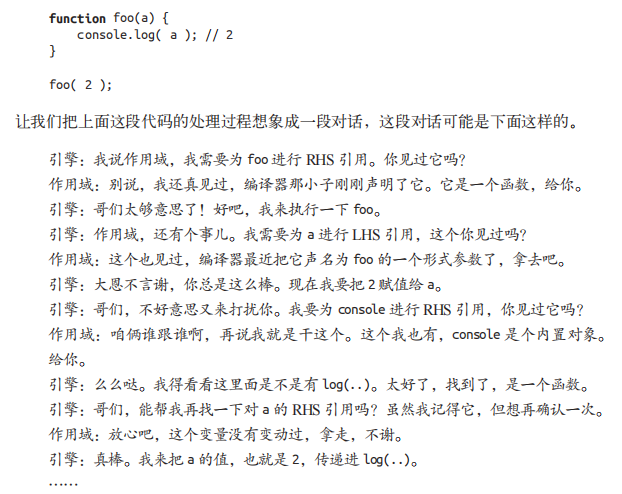
**编译器处理过程**



**LHS/RHS**

理解为“赋值操作的目标是谁（LHS）”以及“谁是赋值操作的源头（RHS）”。

**引擎处理过程**



**异常**

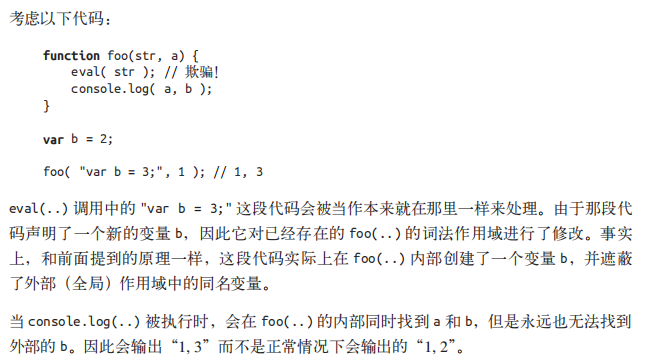
ReferenceError 同作用域判别失败相关，而 TypeError 则代表作用域判别成功了，但是对结果的操作是非法或不合理的。

1. 词法作用域

就是定义在词法阶段的作用域。换句话说，词法作用域是由你在写代码时将变量和块作用域写在哪里决定的，因此当词法分析器处理代码时回保持作用域不变。

**欺骗词法**

1）Eval 2）with



这两个机制的副作用是引擎无法在编译时对作用域查找进行优化，因为引擎只能谨慎地认为这样的优化是无效的。使用这其中任何一个机制都将导致代码运行变慢。不要使用它们。