

# 知识点

---

## 一.升名提升

1. JavaScript 引擎把所有的变量声明和函数声明提升到当前作用域的最前面
2. 规则1：函数声明整体提前（注意：函数表达式）
3. 规则2：变量声明提前，赋值留在原地
4. 规则3：函数会首先被提升，然后才是变量（注意：重复的 var 声明会被忽略，函数声明可以覆盖前面的）

## 二.作用域链

### 1. 作用域

1. 几乎所有的编程语言最基本的功能之一，就是能够储存变量当中的值，并且能在之后对这个值进行访问和修改。
2. 变量储存在哪里以及程序需要时如何找到他们，这些问题说明需要一套设计良好的规则来存储变量，并且之后可以方便地找到这些变量。这套规则称为作用域。
3. 作用域这套规则用来管理引擎如何在当前作用域以及嵌套的子作用域中根据标识符名称进行变量查找。

### 2. 什么是作用域？

1. 作用域就是变量与函数的可访问范围
2. 作用域控制着变量与函数的可见性和生命周期

### 3. 作用域类型

1. 全局作用域
2. 函数作用域
3. 块作用域

### 4. 作用域模型

1. 词法作用域（静态性：是由变量和函数定义的书写位置决定的，与调用位置无关
  1. 词法作用域具有静态性，静态结构决定了一个变量的作用域
  2. 词法作用域与定义位置有关，与调用形式无关
2. 动态作用域（动态性）：由调用位置决定，不关心变量和函数的定义的书写位置

### 5. 作用域链

1. 由当前执行环境与上层（父级）执行环境的一系列变量对象组成
2. 提供对变量和函数访问的权限和顺序的规则

## 三.作用域链与执行上下文

1. 执行时，当前执行上下文，对应一个作用域链环境来管理和解析变量和函数（动态性）
2. 变量查找按照由内到外的顺序（遵循词法作用域），直到完成查找，若未查询到则报错
3. 当函数执行结束，运行期上下文被销毁，此作用域链环境也随之被释放