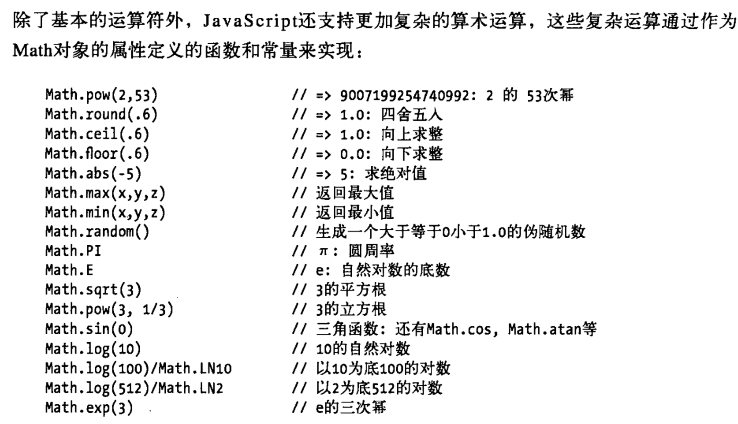
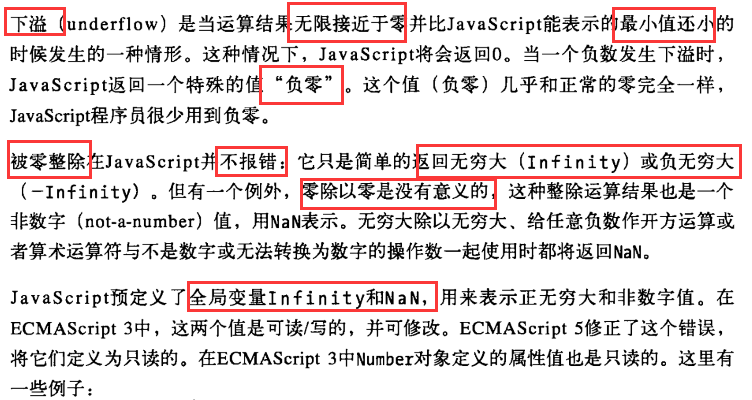
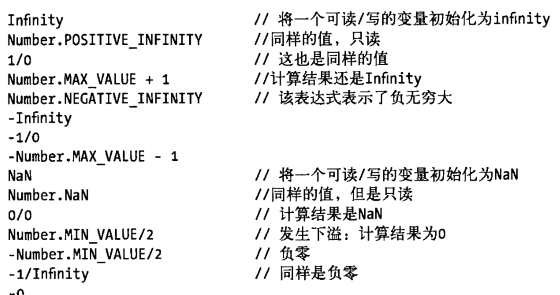
第3章 类型、值和变量

1. 数字
2. 整型直接量
3. 浮点型直接量
4. JavaScript中的算术运算

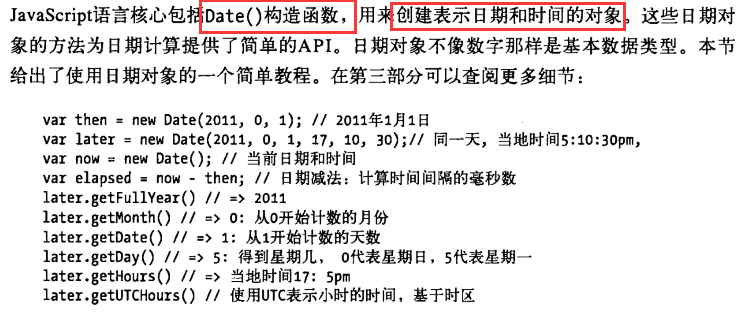
加法运算符(+) 、减法运算符(-)、乘法运算符(\*) 、除法运算符(1) 和求余(求整除后的余数)运算符(%)







1. 二进制浮点数和四舍五入错误
2. 时间和日期



1. 文本

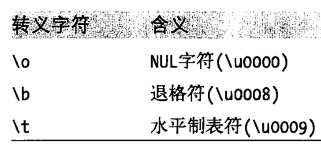
字符串(string) 是一组由16位值组成的不可变的有序序列，

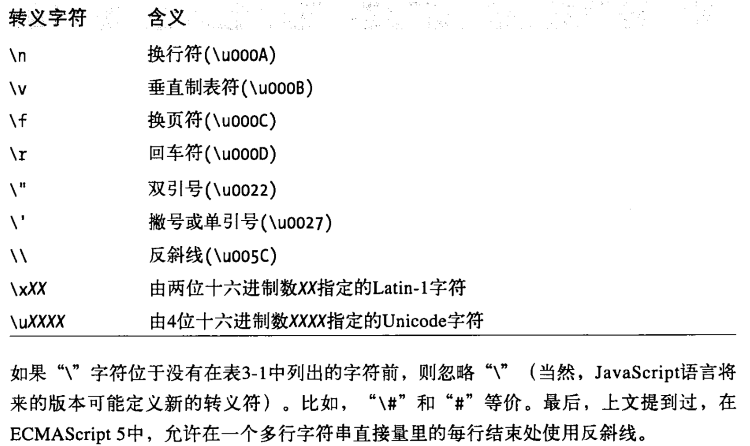
1. 字符串直接量

在JavaScript程序中的字符串直接量，是由单引号或双引号括起来的字符序列

1. 转义字符

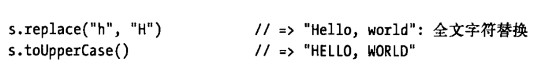
转义字符\n表示换行，\‘表示单引号，

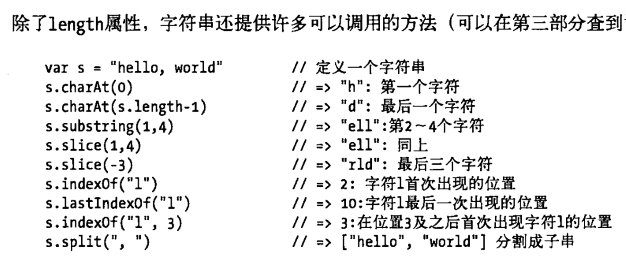




1. 字符串的使用

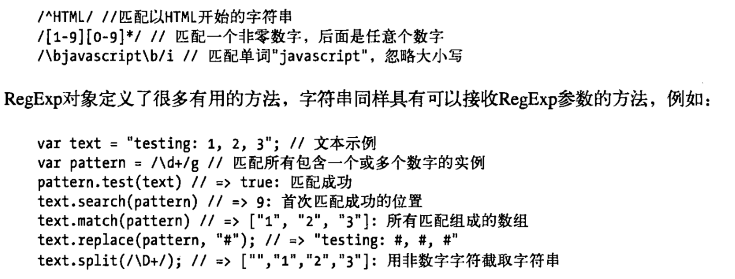
+ 作用于字符串，则表示字符串连接，将第二个字符串拼接在第一个之后





1. 模式匹配

JavaScript定义了RegExp()构造函数，用来创建表示文本匹配模式的对象。这些模式称为“正则表达式”(regular expression)， JavaScript采 用PerI中的正则表达式语法。



1. 布尔值

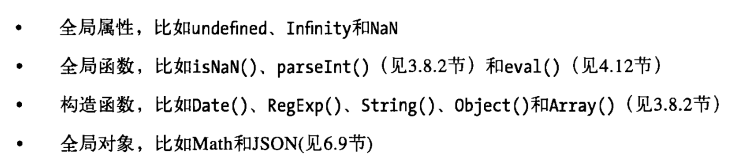
Undefined, null , 0 ,-0, NaN, “ ”空字符串 转换为布尔类型为false

1. Null和undefined

null是JavaScript语言的关键字，它表示一个特殊值，常用来描述“空值”。对nu1l执行typeof预算，结果返回字符串“object" 。

undefined。引用没有提供实参的函数形参的值也只会得到undefined。undefined是预定义的全局变量(它和nu11不一样，它不是关键字, 它的值就是“未定义”

1. 全局变量



1. 包装对象

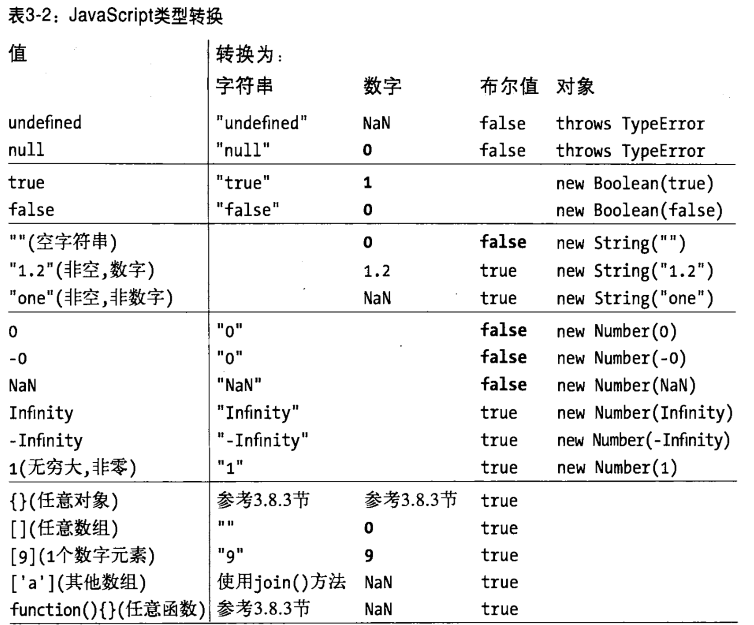
字符串有属性和方法

可通过String(), Number()或Boolean()构造函数来显式创建包装对象。

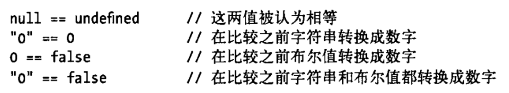
1. 不可变的原始值和可变的对象引用

JavaScript中的原始值(undefined、 null、 布尔值、数字和字符串)不可改变，对象（包括数组和函数）可以改变

1. 类型转换



1. 转换和相等性

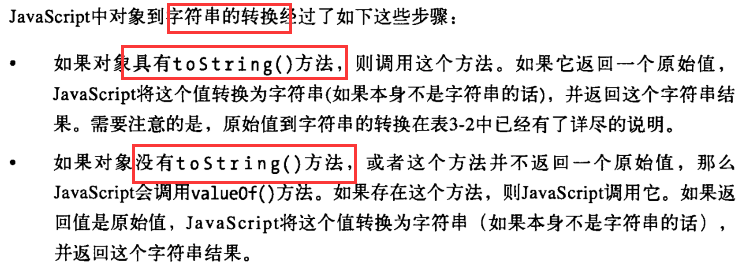


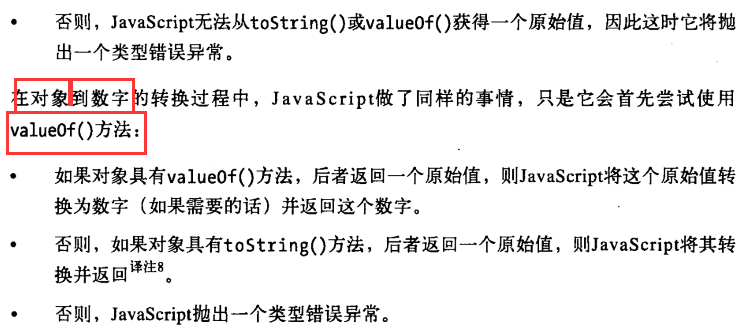
1. 显式类型转换

做显式类型转换最简单的方法就是使用Boolean()、Number()、 String()或0bject()函数。除了null或undefined之外的任何值都具有toString()方法,

1. 对象转换为原始值

对象到布尔值的转换非常简单:所有的对象(包括数组和函数)都转换为true。对于包装对象亦是如此: new Boolean(false)是- 一个对象而不是原始值，它将转换为true.。



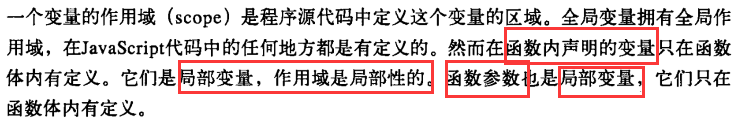


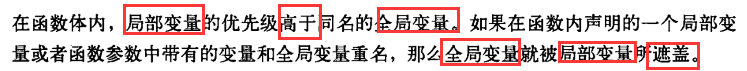
1. 变量声明

变量是使用关键字var来声明的。

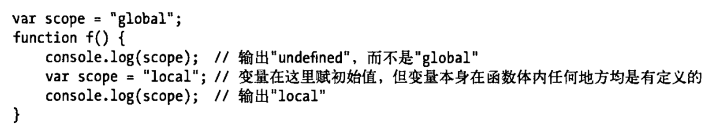
如果未在va r声明语句中给变量指定初始值，那么虽然声明了这个变量，但在给它存入一个值之前，它的初始值就是undefined。

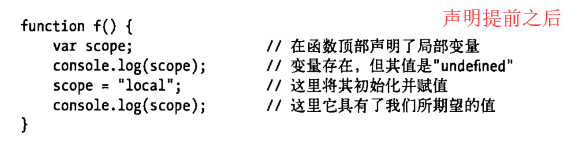
1. 变量作用域





1. 函数作用域和声明前提





1. 作为属性的变量

1）当使用var声明一个变量时，创建的这个属性是不可配置的，也就是说这个变量无法通过delete运算符删除。

2）如果你没有使用严格模式并给一个未声明的变量赋值的话，JavaScript会自动创建一个全局变量。以这种方式创建的变量是全局对象的正常的可配值属性，并可以删除它们。

1. 作用域链

