深入了解JavaScript

**第一部分**

**第1章：基础JavaScript**

**1.1 背景**

1.1.1

JavaScript指的是一种编程语言

**1.2 语法**

1.2.2 语句和表达式

两大语法类别:语句和表达式

1.2.3 分号

分号用于结束语句,而不是结束块.有一种情况你会看到分号出现在块之后:函数表达式作为一个表达式时。如果这样的表达式出现在语句的最后，它后面就会跟上一个分号。

例：var f = function() { };

1.2.4 注释

单行注释:由两个斜杠//开始,行终止时结束

多行注释:限定在/\*和\*/之间

**1.3 变量和赋值**

变量在声明后使用

变量和赋值可以同时进行

标识符与变量名

标识符的第一个字符可以是任意Unicode字符、美元符（$），或者下划线（\_）。

保留字不能作为变量名使用（包括函数名和参数名）

**1.4 原始值和对象**

原始值：布尔值、数值、字符串、null和undefined

其它的值都是对象

原始值得特点：（1）按值进行比较

（2）不可改变：其属性值不能改变、添加或移除

对象：简单对象、数组等

对象的特点：（1）按引用进行比较

（2）默认可变：对象可以很自由地被改变、添加、和移除

对值得分类：typeof和instanceof

Typeof主要用于原始值

Instanceof用于对象

操作数 结果

Undefined ———— ‘undefined’

Null ———— object

布尔值 ———— Boolean

数值 ———— number

字符串 ———— string

函数 ———— function

所有其他数值 ———— object

**1.5 布尔值**

原始布尔类型包含true和false两个值。以下运算会产生布尔值

1. 二元逻辑运算符：&&（与）、||（或）
2. 前置逻辑运算符：！（非）
3. 比较运算符

——相等运算符：===、！==、==、！=

——排序运算符（针对字符串及数字）：>,>=,<,<=

**1.8 字符串（不可改变）**

字符串可以直接通过字符串字面量来创建。这些字面量限定在单引号或双引号之内。反斜杠（/）用于转义字符及产生一些控制字符。

获得字符串的长度

– 通过字符串变量的 length 属性获得 – str.length

获得指定位置的字符

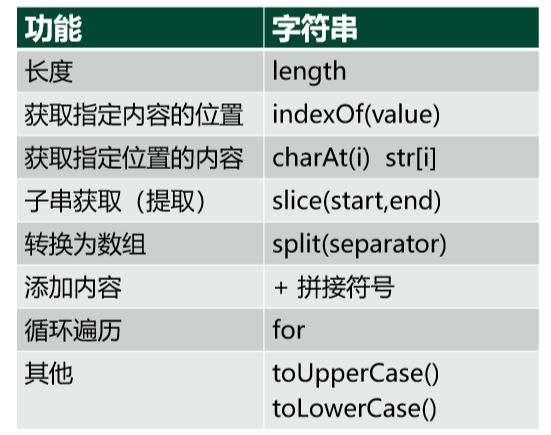
– 通过字符串变量的 charAt(n) 方法获得

– str.charAt(n)

1.8.1 字符串运算符

字符串可以通过加好（+）进行连接。

1.8.2字符串方法



**1.9 语句**

条件语句（if……else，switch……case）

循环语句（for循环，while循环，do—while循环）：break可以跳离循环

Continue会开始一个新的循环

**1.10 函数**

1.10.1 函数声明的提升特性

函数声明具有提升性——它们的实体会被移动到所在作用域的开始处。

Var 声明也具有提升的特性

1.10.2 特殊的变量arguments

在JS中，函数的所有参数都可以被自由调用，它会通过arguments变量来使所有参数可用。Arguments看起来像个数组，但却不具备数组的方法。

1.10.3 参数太多或太少

额外的参数会被忽略

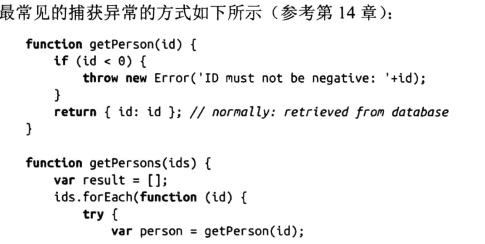
丢失的参数会得到undefined这个值

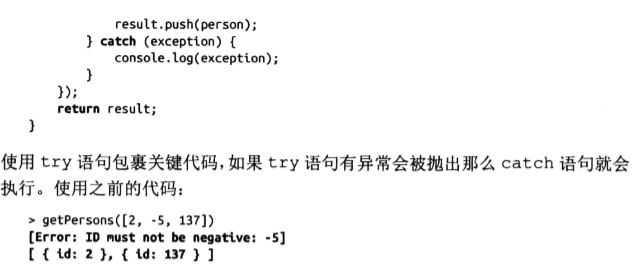
1.10.4 可选参数

1.10.5 强制参数长度

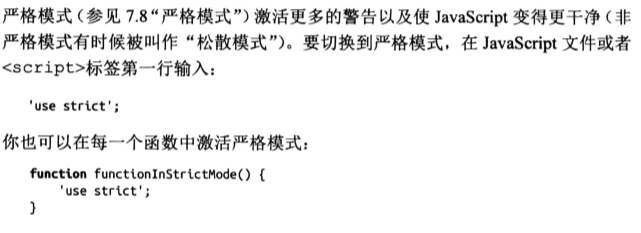
1.10.6 将arguments转换为数组

**1.11 异常捕获**

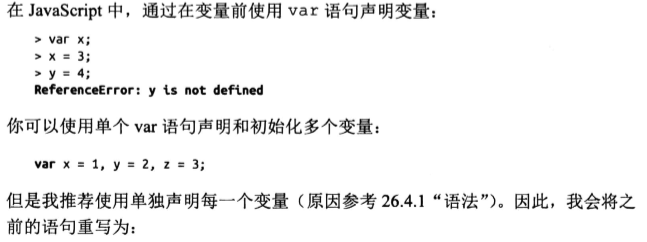


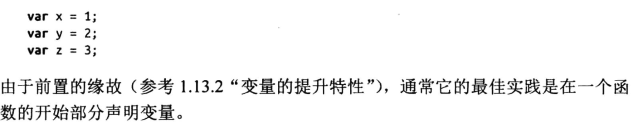


**1.12 严格模式**



**1.13 变量作用域和闭包**





1.13.1 变量是函数作用域的

一个变量的作用域总是完整的函数。

1.13.2 变量的提升特性

所有变量声明都会被提升：声明会被移动到函数的开始处，而赋值则仍然会在原来的位置进行。

1.13.3 闭包

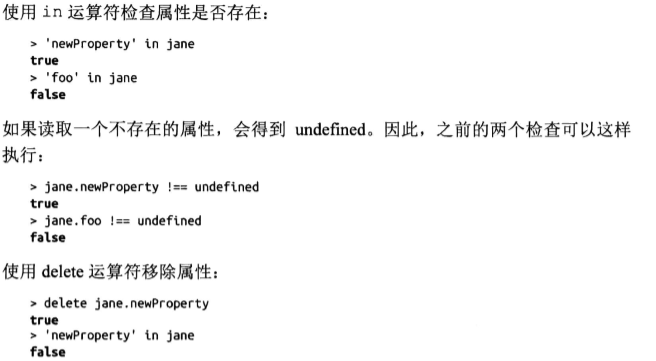
每个函数都和他周围的变量保持着连接，哪怕它离不开被创建时的作用域也是如此。

1.13.4 IIFE模式：引用一个新的作用域

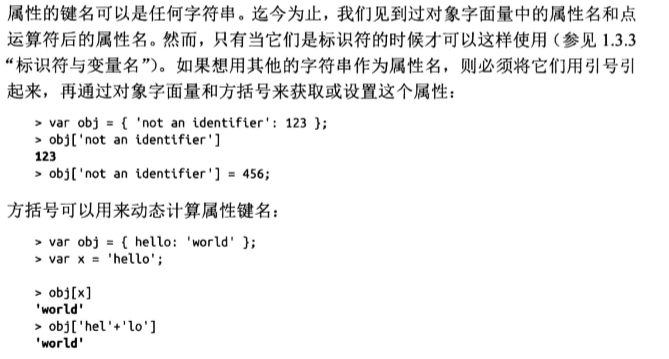
**1.14 对象和构造函数**

1.14.1 单一对象

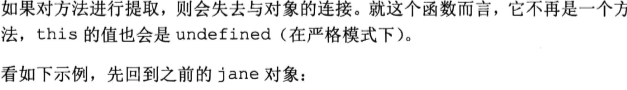


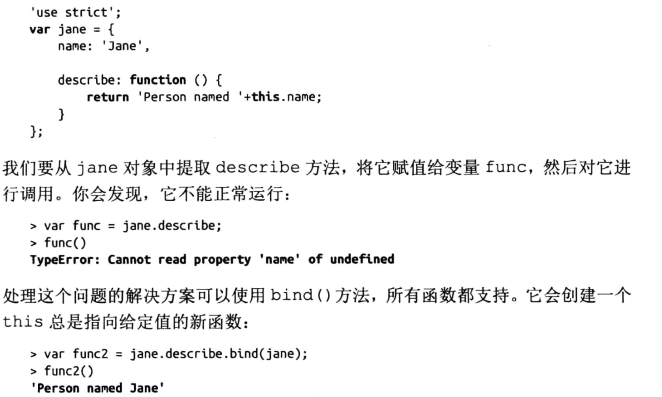


1.14.2 任意属性名

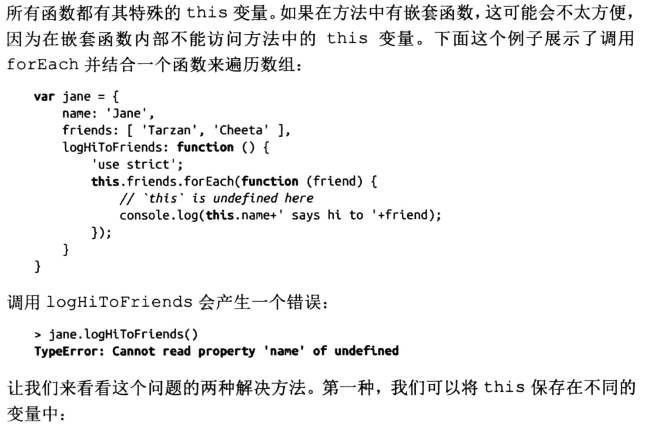


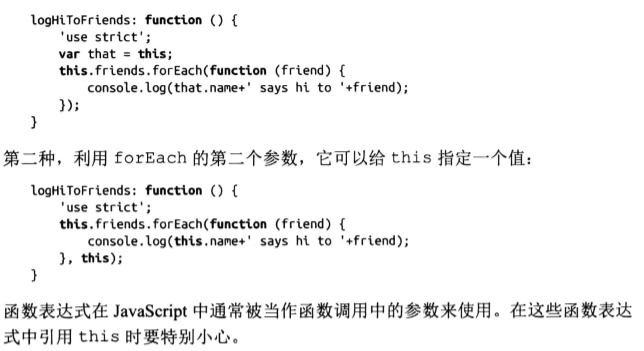
1.14.3 提取方法



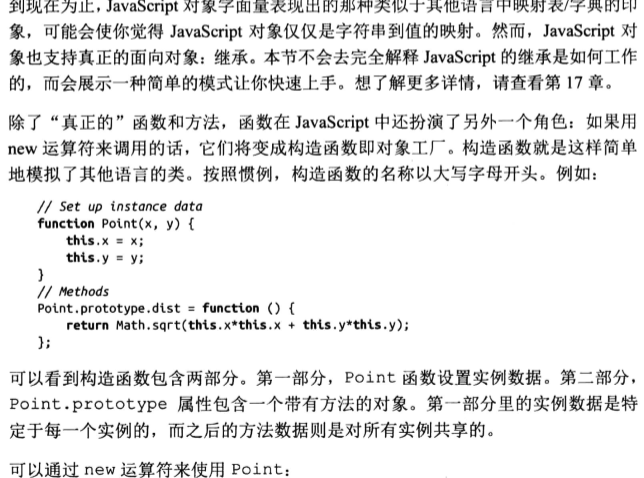


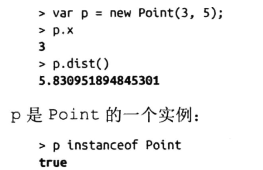
1.14.4 方法中的函数





1.14.5 构造函数：对象工厂

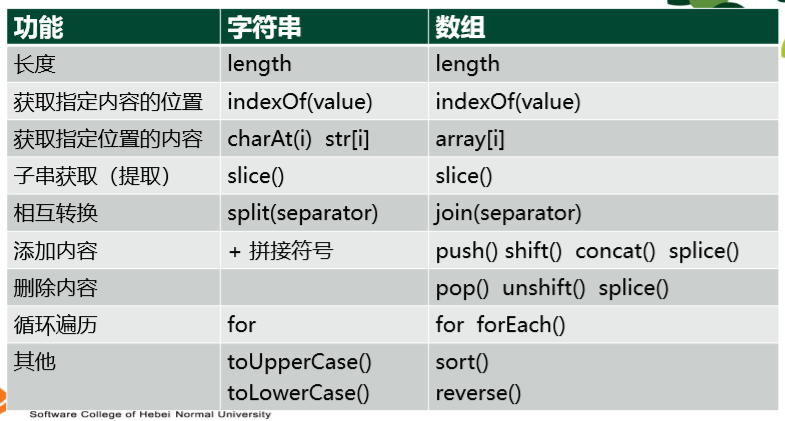




**1.15 数组**

数组是一些有序的元素，可以通过证书索引从0开始被访问

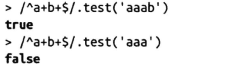
1.15.1 数组方法



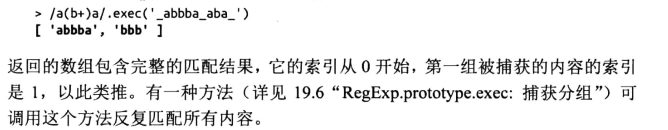
**1.16 正则表达式**

JS内置的支持正则表达式。它们使用斜线分割：

1.16.1 test（）方法：匹配吗



1.16.2 exec()方法：匹配以及捕获分组

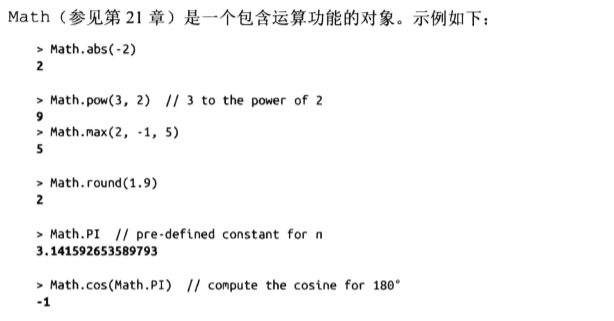


1.16.3 replace（）方法：搜索和替换



Replace的第一个参数必须是一个带着/g标志的正则表达式；否则将只替换第一次出现的内容。还有一种方法可以使用一个函数来完成替换。

**1.17 math**



**1.18 标准库的其他功能**

