第一章：JS概述

前端三驾马车之一：描述网页行为的JS

高端、动态、弱类型

语法：Java；一等函数：Scheme；基于原型：Self

名字：JavaScript——网景，ECMAScript——标准版，Jscript——微软

版本：ES3浏览器基本普及，ES5正在普及，ES6（RN使用）

JS语言核心定义了很少API，但是不包括诸如（网络、存储、图形相关的特性），输入输出功能是由JS所属的“宿主环境”提供的

1.1 JS语言核心

数据类型

JS弱类型，变量可以被赋予不同类型的值

对象和数组

JS数据类型中，对象和数组很重要

对象是名／值对的集合，或字符串到映射的集合，用｛｝定义，通过.或者[]访问

数组是以数字为索引的列表，用［］定义，通过［］访问

数组和对象中都可以包含另一个数组或对象

表达式和语句

表达式是一个短语，该短语可以运算得出一个值

JS中的运算符和OC基本相同，加号可以用来合并字符串

表达式仅仅计算出一个值而不作任何操作，它并不改变程序运行状态

语句并不包含一个值（或者它包含的值我们不关心）但它改变程序的运行状态

函数

函数是带有名称和参数的JS代码段，可以一次定义多次调用

当将函数和对象合写在一起时，函数就变成了方法

1.2 客户端JavaScript

第13章介绍如何让JS在Web浏览器中运行起来

第14章讲解Web浏览器端脚本技术，涵盖客户端JS中的一些重要全局函数

第15章讲解通过JS操纵HTML文档内容

第16章讲解如何使用JS来进行CSS样式的操作

第17章详细描述如何定义、注册事件处理程序，以及事件发生时浏览器是如何调用它们的

第15～17章讲述了如何使用JS来操控网页的内容、样式以及行为。“库”和“框架”解决浏览器兼容性，最流行的库——jQuery

第19章详细介绍jQuery库

后续4章讲解将Web浏览器当作应用平台，以及支持Web应用的现代浏览器API

第18章讲解如何使用JS发起HTTP请求

第20章描述数据存储机制以及应用中的会话状态保持

第21章涵盖基于HTML的<vanvas>标签客户端API，用来进行形状图形的绘制

第22章讲解HTML5所提供的新一代Web应用API

第一部分：JavaScript语言核心

第二章：词法结构

2.1 字符集

JS用Unicode字符集编写：ES3:Unicode2.1；ES5:Unicode3

区分大小写（注意：HTML不区分大小写）

JS忽略空格，多数情况忽略换行；换行符和回车等解析为行结束；

Unicode转义序列：\u后跟随4个十六进制数，可用于字符串直接量、正则表达式直接量、标识符（除了关键字）

JS默认程序代码已经是Unicode标准格式，不会再行做标准化处理

2.2 注释

//或者/\* \*/，注意不能有嵌套

2.3 直接量

“hello” ‘Hi’单双引号都表示字符串

true/false表示布尔值

/javascript/gi正则表达式直接量

null空

2.4 标识符和保留字

标识符必须以字母、下划线或美元符开始，后续可以有字母、数字、下划线或美元符

ES3将所有Java关键字列为保留字，ES5有所放宽

每种特定的JS运行环境都有自己的一个全局属性列表

2.5 可以省略的分号

分号可以省略，但是不要省略

return、break、continue关键字后如果紧跟着换行则JS会在换行处填补分号

“++” “--”运算符作为后缀应该和表达式在同一行，否则会作为前缀与下一行合并

第三章：类型、值和变量

原始类型：数字、字符串、布尔值、null和undefined

对象：属性的集合，每个属性都由“名值对”构成，一个特殊的对象——全局对象

数组：特殊的对象，带编号的值的有序集合

函数：具有与它相关联的可执行代码的对象，函数都是真值

构造函数定义了一类对象。除了数组和函数，JS还有三种有用的类——日期、正则表达式、错误

JS解释器，自动对内存进行垃圾回收（GC）

JS中，技术上讲只有对象才能拥有方法，然而，数字、字符串、布尔值也可以拥有自己的方法，只有null和undefined无法拥有方法

可变类型：对象和数组；不可变类型：数字、布尔值、null和undefined，还有字符串

变量是无类型的，作用域：全局变量；函数内变量

3.1 数字

JS不区分整数和浮点数，统一用IEEE 754标准定义的64位浮点格式表示数字

JS中的实际操作则是基于32位整数

3.1.1 整型直接量

十进制

十六进制：0x或0X开头

八进制：最好别用，只有某些实现支持

3.1.2 浮点型直接量

传统写法和指数计数法，用e或E后面跟指数

3.1.3 JS中的算数运算

语言本身的算数运算符

作为Math对象的属性定义的函数和常量

JS算数运算在溢出、下溢、被零整除时不报错

数字超过JS表示的数字上限，是用特殊值Infinity表示，负数超过负数范围，用－Infinity表示

下溢时JS返回0，负数下溢时JS返回“负零”

被零整除不报错，返回正无穷大或负无穷大，但零处以零例外，返回NaN（not－a－number）无穷大处以无穷大、负数开方或者算术运算符与非数字操作将返回NaN

NaN不等于任何值，用x！＝x来判断

负零和零几乎相等，除了作为除数

3.1.4 二进制浮点数和四舍五入错误

二进制浮点数并不能精确表示小数

3.1.5 日期和时间

Date（）构造函数，创建表示时间和日期的对象

3.2 文本

字符串是一组由16位值组成的不可变的有序序列

JS定义的字符串操作都针对16位值，而不是针对字符

3.2.1 字符串直接量

单引号、双引号均可

ES5 可以拆分多行

注意与HTML的代码混杂

3.2.2 转义字符

反斜杠加各色字符

3.2.3 字符串的使用

加号用来连接

length属性，取得16位值的个数

诸多方法，注意：字符串不可变，返回变化的字符串都是新字符串

ES5 字符串可以当作只读数组

3.2.4 模式匹配

正则表达式RegExp（）是一个特殊对象，它有很多方法

字符串也有很多接收正则表达式参数的方法

3.3 布尔值

任意JS值都可以转换为布尔值

1. －0，“”，null，NaN，undefined这六个值会转换为false，其余值都为true

3.4 null和undefined

null是关键字，执行typeof，返回字符串“object”

undefined不是关键字，是预定义的全局变量，如果函数没有返回任何值，则返回undefined，执行typeof，返回“undefined”

相等“＝＝”运算符认为两者相等，严格相等“＝＝＝”区分两者

3.5 全局对象

当JS解释器启动（或任何Web浏览器加载新页面的时候），它将创建一个新的全局对象，并给它一组定义的初始属性：

全局属性，如：undefined、Infinity和NaN

全局函数，如：isNaN（）、parseInt（）、eval（）

构造函数，如：Date（）、RegExp（）、String（）、Object（）、Array（）

全局对象，如：Math和JSON

代码最顶级可以通过this引用全局对象

客户端JS中，浏览器窗口中的JS代码中，Window对象充当了全局对象