# 1 前言

20世纪90年代中期，当网上冲浪越来越流行时，对于开发客户端脚本的需求也逐渐增大。此时，大部分因特网用户还仅仅通过 28.8 kbit/s 的调制解调器连接到网络，即便这时网页已经不断地变得更大和更复杂。而更加加剧用户痛苦的是，仅仅为了简单的表单有效性验证，就要与服务器进行多次地往返交互。设想一下，用户填完一个表单，点击提交按钮，等待了 30 秒的处理后，看到的却是一条告诉你忘记填写一个必要的字段。

那时正处于技术革新最前沿的 Netscape，开始认真考虑开发一种客户端脚本语言来解决简单的处理问题。当时工作于 Netscape 的 Brendan Eich，开始着手为即将在 1995 年发行的 Netscape Navigator 2.0 开发一个称之为 LiveScript 的脚本语言。

当时Netscape和SUN公司有合作关系，Netscape 与 Sun 及时完成 LiveScript 实现。就在 Netscape Navigator 2.0 即将正式发布前，Netscape 将其更名为 JavaScript，目的是为了利用 Java 这个因特网时髦词汇。Netscape 的赌注最终得到回报，JavaScript 从此变成了因特网的必备组件。

html/css/javascript之间的关系是怎样的呢？首先它们都可以采用普通的文本编辑器开发，然后使用浏览器软件打开并解释执行；html是一种超文本标记语言，主要是做页面展示，W3C制定了html规范与标准；css是一种层叠样式表语言，主要用来辅助修饰html，让html网页看起来更加的丰富与漂亮；javascript是一种编程语言，它可以用来操作html中的节点以及css样式，进而达到网页的动态效果，增强网页与用户的交互度。

。

## 1-1 JavaScript和Java

JavaScript是网景公司的布兰登艾奇开发的，Java是SUN公司詹姆斯高斯林带领团队开发的，JavaScript程序被内置在浏览器软件当中的JavaScript解释器解释执行，而Java语言必须运行在Java虚拟机当中，所以JavaScript和Java没有任何关系。只是名字当中都带有Java字眼，这也许只是一种市场营销策略。

## 1-2 JS和JSP

JavaScript被程序员简称为JS，运行在浏览器客户端当中；JavaServer Pages简称为JSP，是一种基于Java语言实现的服务器端页面，JSP属于Java语言，运行在Java虚拟机当中。

## 1-3 JavaScript语言特点

1.基于对象的编程语言

JavaScript是一种基于对象的编程语言而不能说是面向对象的编程语言，因为对象性的特征在JavaScript中并不像Java语言中那样纯正。在JavaScript中有内置的对象，同时用户也可以创建并使用自己的对象。

弱类型语言

JavaScript编程语言没有编译阶段，文本编辑器开发完毕之后，直接使用浏览器打开即可解释执行，所以JavaScript中的变量在定义的时候不需要指定数据类型，并且变量赋值的时候可以随意赋值，具体赋值的时候，值决定变量的数据类型。JavaScript在变量约束方面非常弱，所以称为弱类型语言。Java语言存在编译阶段，在Java语言当中声明变量i为int类型，那么变量i只能接收int类型的字面值，如果字面值的数据类型和变量的数据类型不同，程序是无法编译通过的，并且从int类型的变量i开始声明到最终程序执行结束，i变量的数据类型永远都是int类型，占用4个字节，像Java这种对变量约束较强的编程语言被称为强类型编程语言。

2.解释执行的脚本语言

JavaScript是一种脚本语言，脚本语言是一种解释性的语言，以普通文本形式保存，不需要编译生成目标程序，可以直接用文本编辑器打开查看。它不象c\c++等可以编译成二进制代码以可执行文件的形式存在。JavaScript脚本语言可直接使用JavaScript解释器解释执行，可能在解释的过程当中进行编译。

3.简单性

JavaScript基于Java的基本语法和语句流程，而Java是从C++语言发展而来，因此有过C语言开发经验的人员学习JavaScript十分容易。此外，JavaScript是一种弱类型语言，其变量并没有严格的数据类型，免去了许多麻烦。

4.动态性

JavaScript是基于事件驱动的，所谓事件驱动就是触发一定的操作而引起某些动作。例如，鼠标单击按钮，页面加载完毕等等这些都是事件。可以根据不同的事件创建相应的响应代码，这样就可以实现和用户的动态交互。

5.平台无关性

前面提到JavaScript代码在浏览器中解释执行，并没有利用具体平台的特性，所以只要有支持JavaScript的浏览器，无论在什么平台上代码都能得到执行。开发人员在编写JavaScript脚本过程中就无需考虑具体平台的限制。只需要考虑浏览器兼容问题即可。

6.安全性

JavaScript是安全的，其不允许访问本地硬盘，也不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互。从而有效地防止数据的丢失和破坏。

## 1-4 JavaScript组成

ECMAScript

Netscape（网景）公司推出JavaScript不久，不甘落后的微软也推出了微软版的JavaScript-JScript，微软主要将其用于自己的浏览器Internet Explorer。最初两家公司各自为政，分别执行自己的标准。那个时候的web程序员是很痛苦的，通常需要将相同的功能编写两份不同语法的程序。

1999年，欧洲计算机协会（ECMA）依据JavaScript制定了ECMAScript的ECMA-262规范，ECMAScript成为JavaScript的标准。之后两家公司都遵循了该规范，所以JavaScript和Jscript大部分是相同的，但也有小部分不同之处。

2.DOM

Document Object Model（文档对象模型），在JavaScript当中，把html文档当做一棵树，这棵树有一个根节点<html></html>，根节点下有很多子节点，每一个节点都可以看做是一个DOM对象，JavaScript通过操作DOM对象完成重构整个HTML文档、添加、移除、改变或重排页面上的项目。从而达到网页动态效果，提高与用户的交互度。

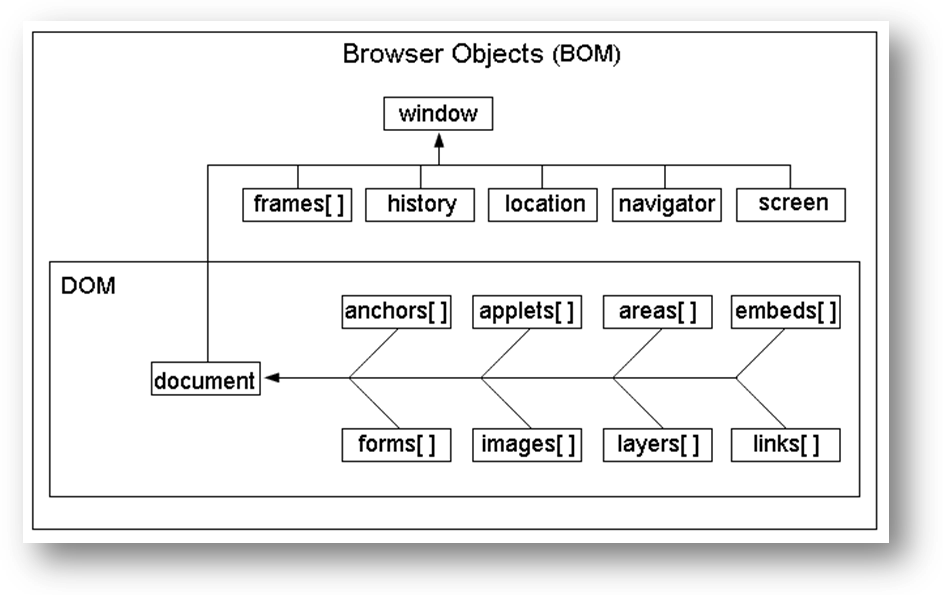
JavaScript对DOM元素的操作又可以称为“html dom编程”。HTML DOM是HTML Document Object Model(文档对象模型)的缩写，HTML DOM则是专门适用于HTML/XHTML的文档对象模型。熟悉软件开发的人员可以将HTML DOM理解为网页的API。它将网页中的各个元素都看作一个个对象，从而使网页中的元素也可以被计算机语言获取或者编辑。 例如Javascript就可以利用HTML DOM动态地修改网页。

根据W3C DOM规范,DOM是一种与浏览器，平台，语言无关的接口，使得你可以访问页面其他的标准组件。简单理解，DOM解决了Netscape的JavaScript和 Microsoft的JavaScript之间的冲突，给予web设计师和开发者一个标准的方法，让他们来访问他们站点中的数据、脚本和表现层对象。

3.BOM

Browser Object Model（浏览器对象模型），使JavaScript能够与浏览器进行“对话”，获取浏览器信息，操作浏览器。虽然W3C并没有对BOM作出规范，但是所有浏览器都支持BOM，有一些事实上的标准。

BOM和DOM之间的关系



## 1-5主流的浏览器

JavaScript脚本语言最终被浏览器当中的JS解释器解释执行，所以不同的浏览器执行相同的JS代码可能结果不尽相同，所以对于WEB前端开发人员来说，解决浏览器兼容问题是一件很头疼的事情。通常一个WEB前端开发人员需要在自己的电脑中安装不同厂商的浏览器，有可能相同厂商的浏览器也需要安装不同的版本。下面我们来看一看世界主流的5大浏览器都有谁？



左上角位置：谷歌公司的Chrome浏览器。

左下角位置：safari浏览器，苹果公司旗下浏览器。

右上角位置：微软公司的IE浏览器。

右下角位置：FireFox浏览器，简称FF浏览器，mozilla公司旗下浏览器

中间位置：opera浏览器，挪威厂商opera旗下浏览器。

我们在讲课的过程中安装的浏览器有：FF/Chrome/IE。

JavaScript开发工具

文本编辑器EditPlus

我们讲课的过程中，第一天的知识点采用普通的文本编辑器开发，这种开发方式还是比较适合JavaScript初学者的。建议初学者开始的时候使用文本编辑器。

集成开发工具WebStorm

在实际的Web前端开发过程中，为了提高开发效率，可以借助某些集成开发环境，例如Web前端开发利器：WebStorm，这个名字就够气派的吧：Web风暴。该集成开发工具是Intellij IDEA开发工具中的一个插件，Intellij IDEA开发工具是一个非常棒的Java开发环境。我们在后期的课程中将使用WebStorm进行开发。

使用WebStorm将会体验到丰富的JS代码提示功能。它不仅仅对JS代码提供了丰富了提示，包括HTML、CSS提示也是非常全面的。

JavaScript帮助文档

<http://www.w3school.com.cn/> 全球最大的中文 Web 技术教程。我们给大家提供的帮助文档是w3shool的离线帮助文档

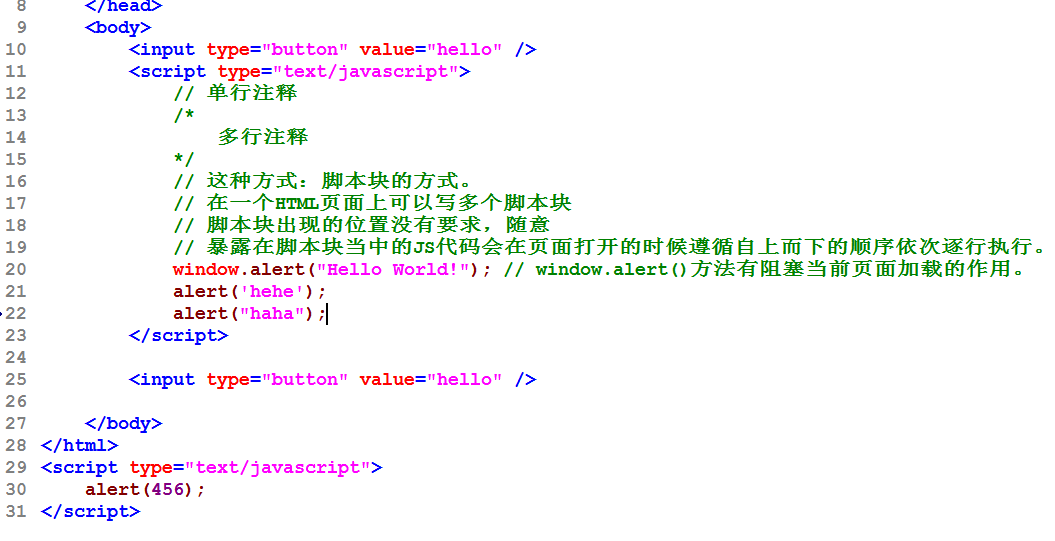
# 2 JavaScript基础语法

## 2-1 JS引入的三种方式

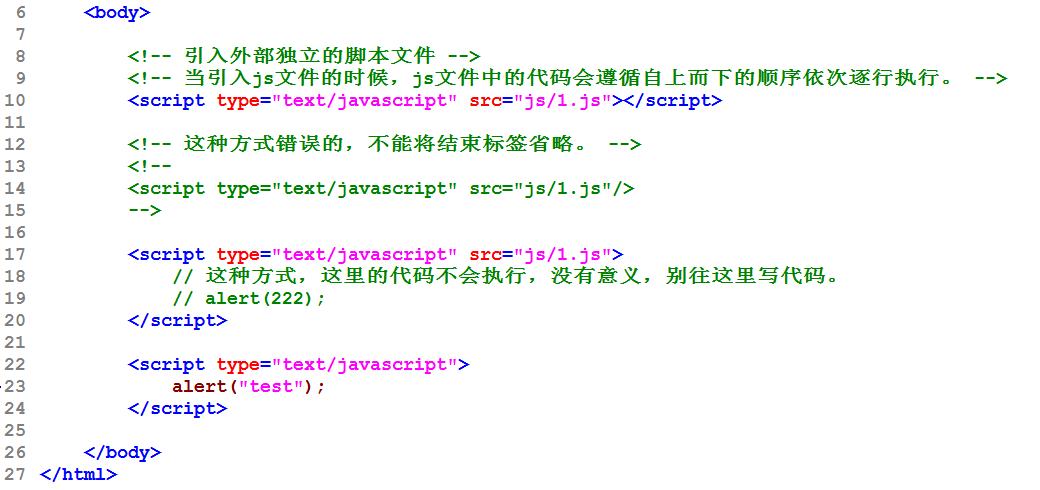
### 2-1-1 嵌入标签



### 2-1-2 HTML中嵌入JS脚本

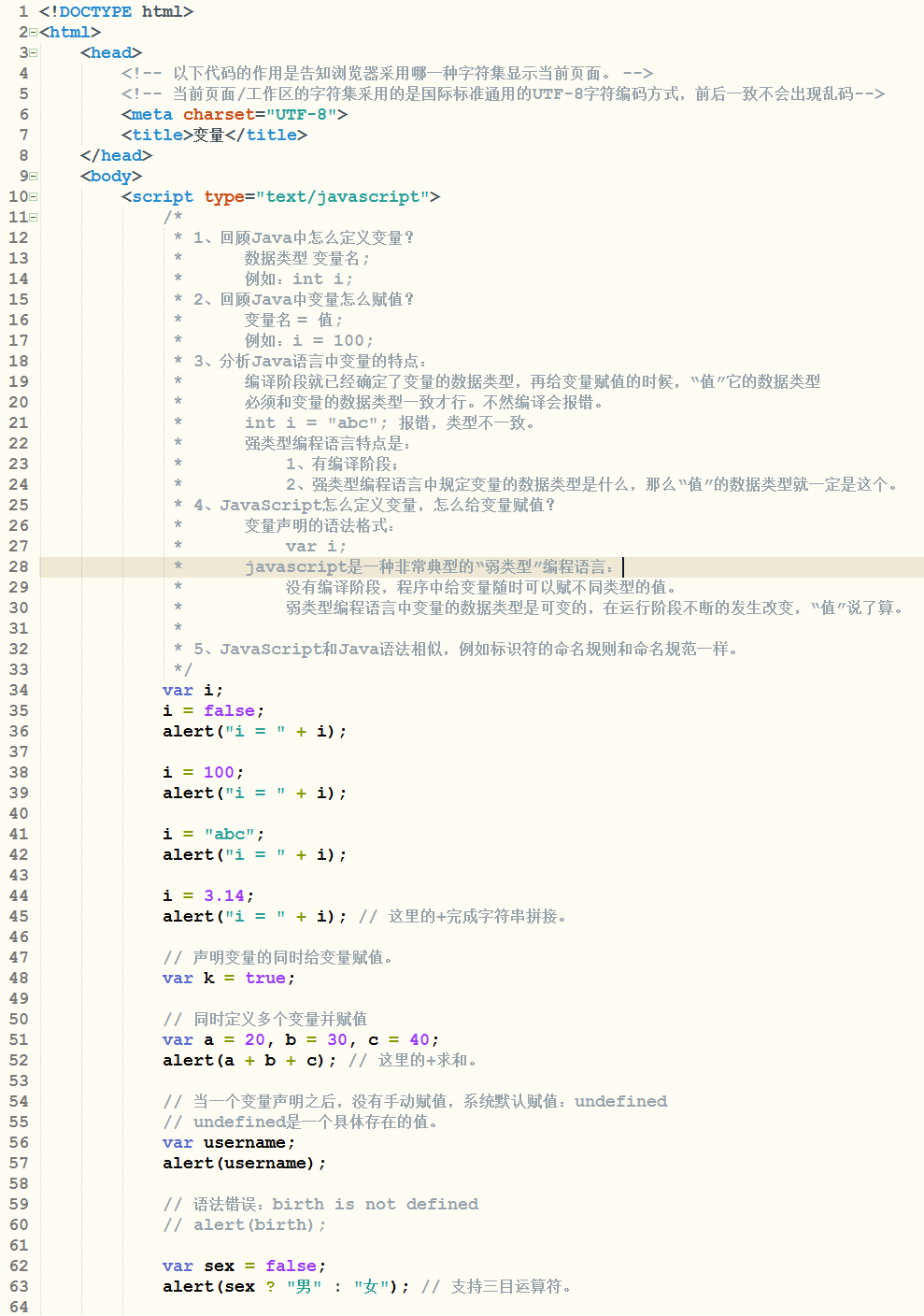


### 2-1-3 引入外部JS文件



## 2-2 变量

### 2-2-1 JS是弱类型的编程语言



### 2-2-2 JS中变量的作用范围



## 2-3 数据类型

1.java中的数据类型

基本数据类型：byte,short,int,long,float,double,boolean,char

引用数据类型：Object,String,.....

2.javascript包括哪些数据类型呢？

原始类型（基本数据类型）：Undefined、Number、Boolean、String(小)

引用数据类型：Object以及所有子类型都是引用数据类型，String(大)

注意: 大String的父类是Object。

ES6规范中加入了一种新的类型：Symbol类型。

3、JS中的数据类型，可以这样定义变量吗？

Number i = 100;

String s = "abc";

在进行变量声明的时候也不能使用以上的语法格式，只能使用var i = 100;

虽然定义变量的时候不需要指定数据类型，但起码要知道在JS中的“每一个数据”都是属于哪一种类型的，例如：100，"abc"，true/false是什么类型。

## 2-4 typeof函数的用法

1.typeof运算符可以在程序运行阶段动态的获取变量的数据类型。

2、思考：为什么JS中需要typeof运算符呢？

因为JS是一种弱类型语言。

3、typeof运算符怎么用？

typeof 变量名

4、typeof运算符的最终运算结果是以下6个结果之一：

"undefined"

"number"

"boolean"

"string"

"object"

"function"

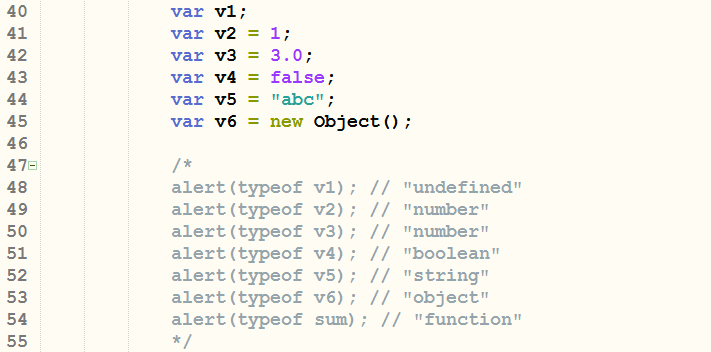
注意：typeof运算符最终的结果永远都是全部小写的“字符串”

5、在JS中怎么比较两个字符串是否相等呢？

没有equals方法，只有双等号。

JS中的双等号就等同于java中的equals方法。

1. 在JS中null 属于Object类型。



## 2-5 Undefined数据类型



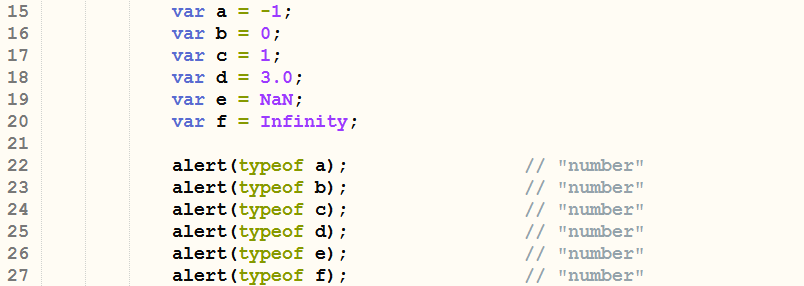
## 2-6 Number

### 2-6-1 Number类型

被称为数字类型，包括哪些值？

-100,-1,0,1,2,3,4,3.14,...,NaN,Infinity

Number类型的值不一定都是数字。其中NaN就不是一个数字，但属于 Number 类型的范畴。



### 2-6-2 Infinity

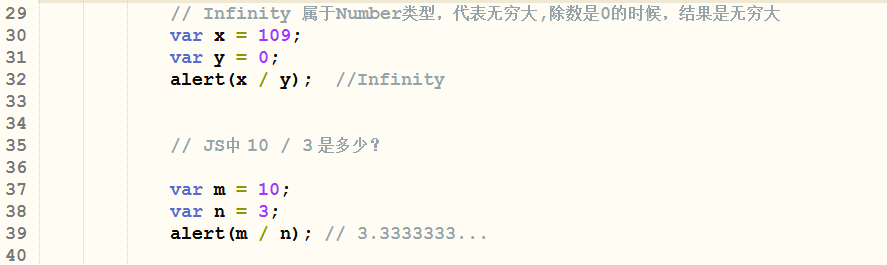
这是一个实际存在的值，这个值属于Number类型，代表无穷大

// 只有当除数是0的时候，结果是无穷大。

var x = 109;

var y = 0;

alert(x / y); //Infinity



### 2-6-3 NaN

NaN是一个实际存在的值，这个值属于Number类型，代表“不是一个数字”

NaN : Not a Number（不是一个数字）

当结果本应该返回数字，结果却没有返回数字，只能返回一个NaN。

var g = 100;

var i = "中国";

alert(g / i);

除法结果本来应该是一个数字，但是最后算完了不是数字形式的，结果在JS中只能是NaN。

var o = 100;

var q = "中国";

alert(o + q); // "100中国"

alert(o \* q); // NaN

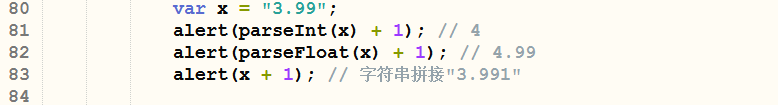
### 2-6-4 isNaN()



### 2-6-5 parseInt()/parseFloat()

这个两个函数都可以将“字符串”转换成“数字”。

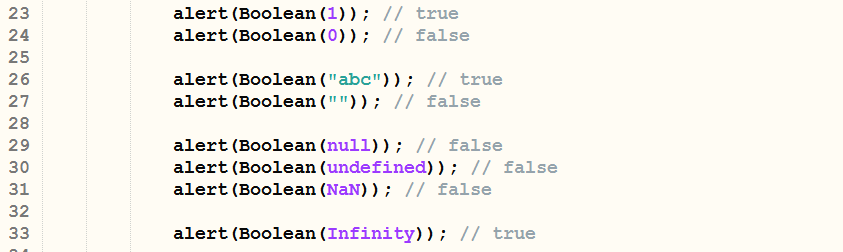
并且parseInt()函数是直接取整数位。【不会四舍五入】



## 2-7 Boolean

1、Boolean类型和java一样，也是只有两个值：true和false，没有其他值。

2、重点：布尔类型涉及到一个Boolean()函数，这个函数可以将“非布尔类型”转换成“布尔类型”。



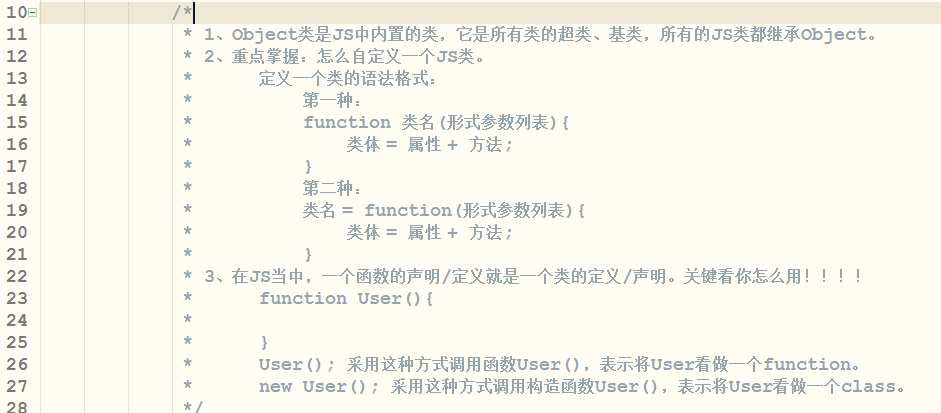
## 2-8 String

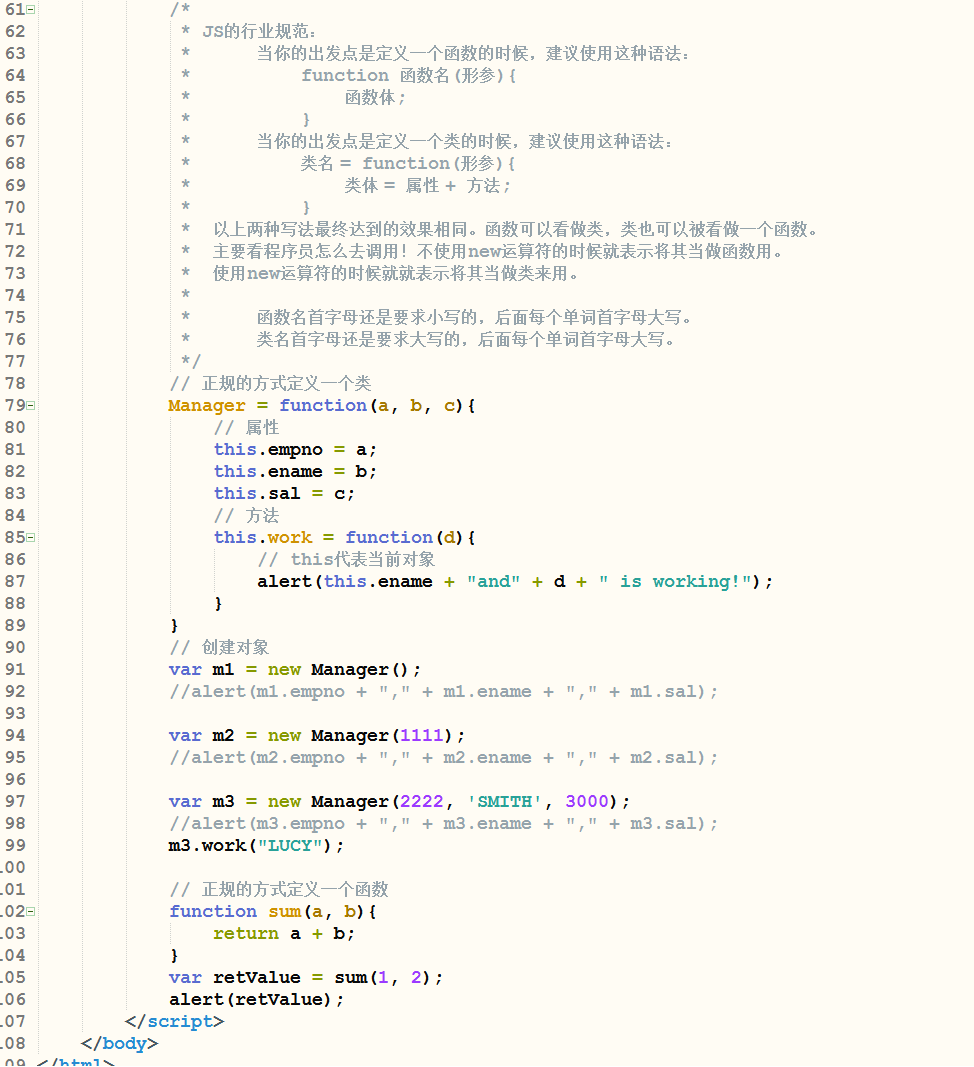


## 2-9 函数



## 2-10 Object





还可用中括号的形式访问对象中的属性。



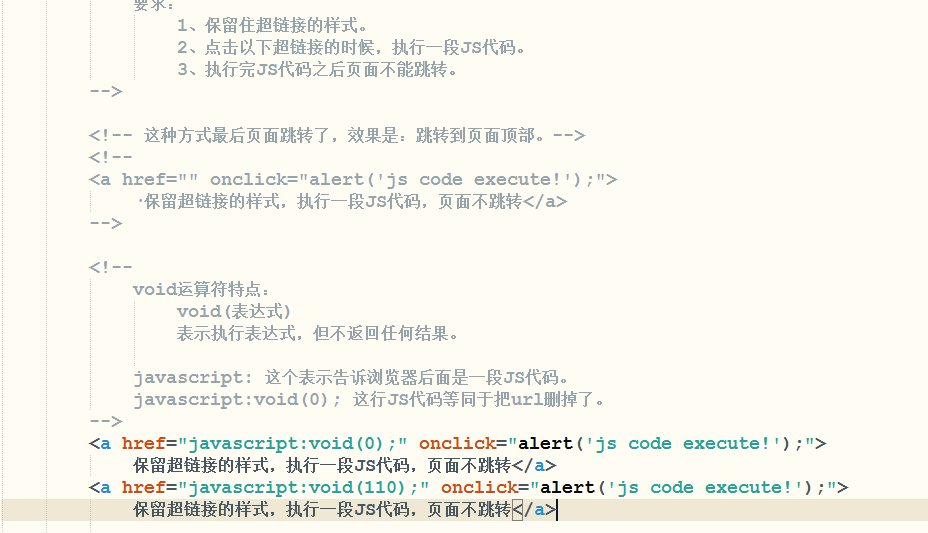
### Prototype属性

// Object类中有一个prototype属性，该属性可以给类动态的扩展方法和属性。

// 由于所有的类都默认继承Object，所以自定义的类也有prototype属性。



## 2.10 void运算符



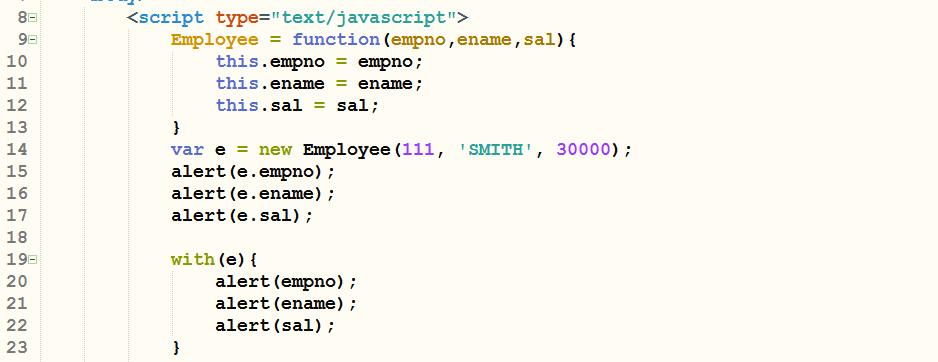
## 2-11 Null/NaN/undefined区别



## 2-12 for in 语句



## 2-13 with语句



## 2-14 常用内置类

### 2-14-1 常用内置类

JavaScript中常用的内置的对象：

\*window：代表当前浏览器窗口

\* document：代表整个网页文档

\* ....

\* 实际上document对象是window对象的一部分。

JavaScript包括三部分：

\* ECMAScript：JS的核心语法。

\* DOM：Document Object Model（操作document中元素），document 对象是DOM中老大。

\*BOM：Browser Object Model（操作浏览器中的元素），window是BOM 中老大。

\*实际上BOM是包括DOM。（BOM是缺少行业规范，DOM是W3C制定 的规范！！！！）

JavaScript中常用的内置类：

\* 1、Object

\* 2、String

\* 3、Date

\* 4、Array

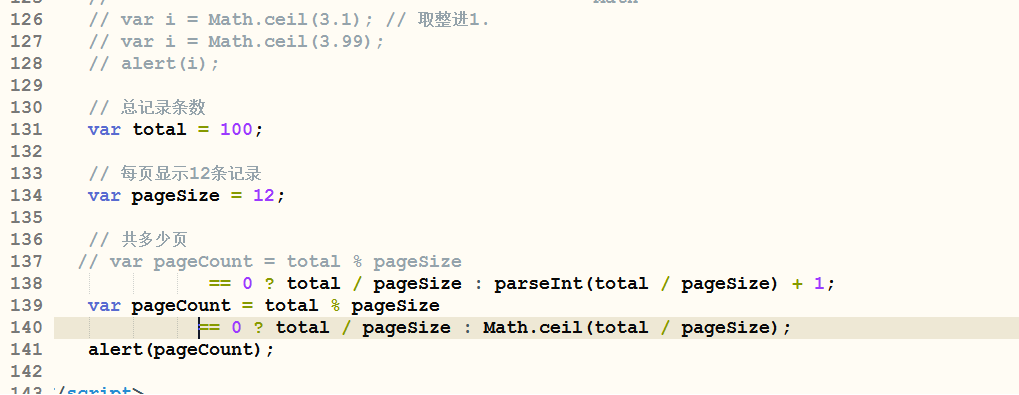
\* 5、Math

\* 6、RegExp：在后面正则表达式的时候再看。

### 2-14-2 Date类



### 2-14-3 Math类



### 2-14-4 Array类



# 3 事件及事件句柄

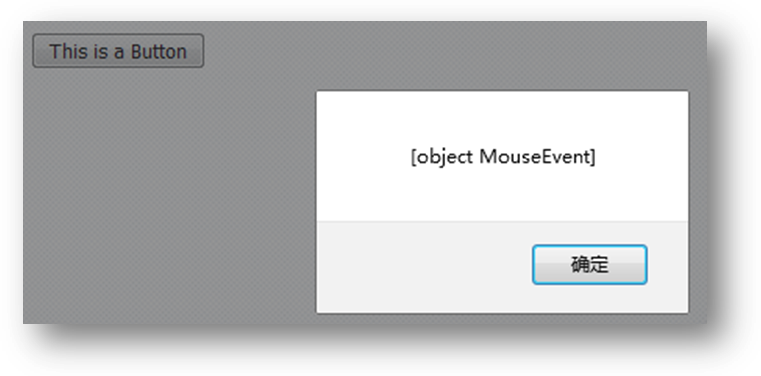
## 3-1 事件及event对象

JavaScript是一种基于事件驱动型的编程语言，当发生某个特殊的事件后，注册在事件句柄后面的JavaScript代码会被调用。事件句柄通常是一个html标签的属性，例如按钮button有onclick属性，而onclick属性就是事件click对应的事件句柄。事件句柄的命名规律一般是事件名称前添加on。

在JavaScript当中内置了event对象，可以直接拿来使用，注意event全部小写，event对象表示刚发生的事件对象。



点击按钮，效果如下所示：



## 3-2 常见事件及事件句柄

1. click事件

鼠标单击事件，事件句柄：onclick

1. dblclick事件

鼠标双击事件，事件句柄：ondblclick

1. blur事件

失去焦点事件，事件句柄：onblur

1. change事件

当文本框、文本域中的文本内容或者下拉列表选中项发生改变时，该事件发生，事件句柄：onchange

1. focus事件

获得焦点事件，事件句柄：onfocus

1. load事件

网页加载完毕后发生，事件句柄：onload，通常编写在body标签当中。

1. keydown事件

键盘按键被按下时发生，可以捕获所有键，除“prt sc sysrq”键之外。事件句柄：onkeydown

1. keypress事件

键盘按键被按下时发生，主要用来捕获数字、字母、小键盘，其它键无法捕获，但是在FF浏览器当中可以捕获所有键。事件句柄：onkeypress

1. keyup事件

键盘按键弹起时发生，事件句柄：onkeyup

1. mousedown事件

鼠标按下事件，事件句柄：onmousedown

1. mousemove事件

鼠标在某控件上移动时发生，事件句柄：onmousemove

1. mouseover事件

鼠标经过某控件时发生，事件句柄：onmouseover

1. mouseout事件

鼠标离开某控件时发生，事件句柄：onmouseout

1. mouseup事件

鼠标按键释放时发生，事件句柄：onmouseup

1. reset事件

表单数据被重置的时候发生，事件句柄：onreset

1. submit事件

表单数据被提交时发生，事件句柄：onsubmit

1. select事件

当文本框、文本域中的内容被选中时发生，事件句柄：onselect

1. abort事件

图像的加载被中断时发生，事件句柄：onabort

1. error事件

加载图像或文档出错时发生，事件句柄：onerror

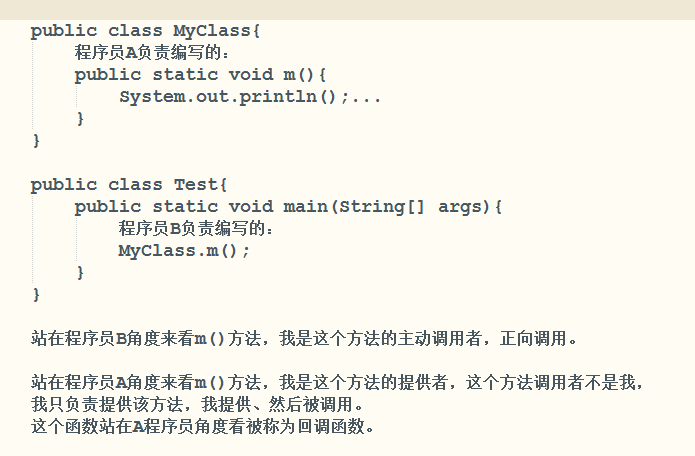
1. resize事件

窗口或框架被重新调整大小时发生，事件句柄：onresize

1. unload事件

用户退出当前页面时发生，事件句柄：onunload

## 3-3 回调函数



## 3-4 完成事件注册

在JS当中怎么完成事件的注册？

第一种方式：

在HTML标签中使用事件句柄这个属性，在事件句柄后面添加JS代码，当这个事件发生之后，JS代码执行。

第二种方式：

单独编写一段JS代码来完成事件的注册。

var elt = document.getElementById("eltId");

elt.onXX = function(){

// 回调函数，等XX事件发生后，这个回调函数会执行。

}



## 3-5 事件绑定失败



## 3-6 body标签的onload属性



## 3-7 window.onload



## 3-8 事件对象event



# 4 DOM编程

## 4.1 登陆表单验证



## 4.2 innerHTML和innerText

1、innerHTML和innerText共同点是：都可以设置元素内部的内容。

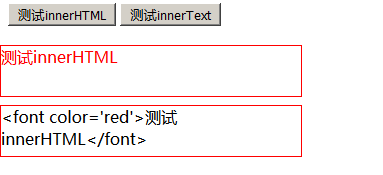
2、innerHTML和innerText不同点是：innerHTML把内容当做一段HTML代码解 释并执行，innerText把内容当做普通的字符串处理。

3、innerHTML和innerText都不是方法，是属性。

4、 innerHTML会把后面的字符串当做一段HTML代码解释执行，呈现一个执行 的效果。 innerText后面即使是一段HTML代码，也只是把这个HTML代码当 做一段普通文本显示出来。



执行效果如下



## 4.3 复选框全选和取消全选



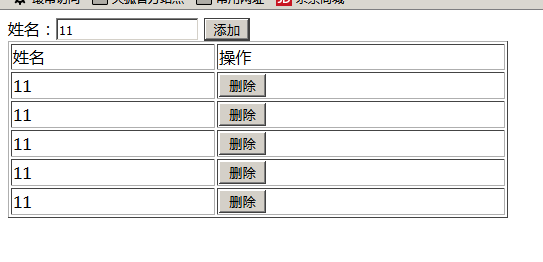
## 4.4 获取下拉列表选项value



## 4.5表格行的动态添加和删除

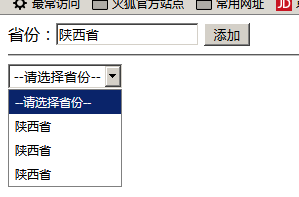


运行结果：



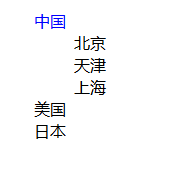
## 4.6 下拉列表添加新选项





## 4.7 菜单的显示和隐藏





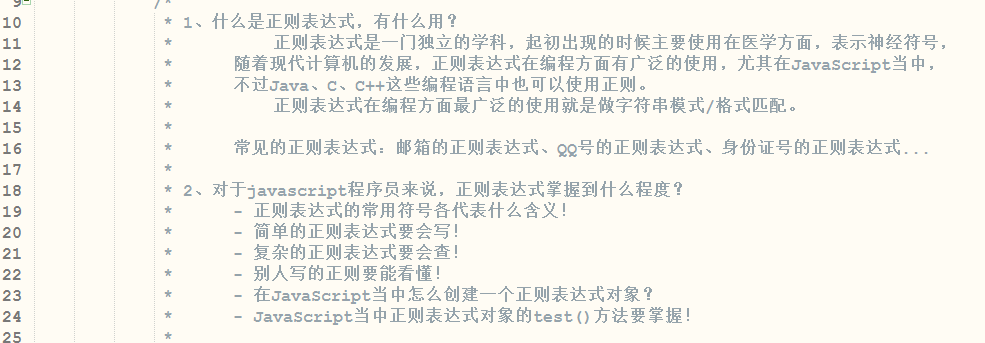
## 4.8 网页时钟



## 4.9 获取DOM元素的三种属性



## 4.10 正则表达式





## 4.11表单验证



# 5 BOM编程

## 5.1 windows对象

Window.confrim();



Window.open();



Window.close();关闭当前窗口

父子窗口之间的通信





## 5.2 history对象





## 5.3 location对象



# 6 JSON

## 6.1 eval()函数



## 6.2 定义jason

1、什么是json？json在开发中有什么用？

JSON:

- JSON是JavaScript当中的一种引用类型。

- JSON对象默认继承Object。

- JSON: JavaScript Object Notation（JavaScript导航对象）

2、JSON是一种无类型的轻量级JavaScript对象，格式清晰，容易解析，体 积小。

3、JS中的大括号{}和中括号[]有什么区别？

{} 用来创建JSON对象

[] 用来创建数组

1. jason的属性值可以是任何类型的元素

数组类型或者jason



1. 轻量级和重量级数据类型的比较：



1. jason可以在开发中用来进行数据交换。

