logo.png

javascript基础

H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# javascript

## 1.1 javascript的来源

**1994年 Netscape** 网景通信公司推出了代号为“网景导航者”的网景浏览器1.0，随后迭代版本迅速占领浏览器大部分份额。

**微软看到网景公司可能影响他在行业内的地位，微软收购了另一家做浏览器的公司，做了自己的IE浏览器。**

把IE打包到window操作系统中。

Ie能打败网景。

网景就宣布旗下浏览器免费，开源。起名firefox。

网景破产了。Firefox内核就交给Mozilla全称[Mozilla基金会](http://baike.baidu.com/item/Mozilla%E5%9F%BA%E9%87%91%E4%BC%9A" \t "_blank)，是为支持和领导开源的Mozilla项目而设立的一个[非营利组织](http://baike.baidu.com/item/%E9%9D%9E%E8%90%A5%E5%88%A9%E7%BB%84%E7%BB%87" \t "_blank)，负责该浏览器的维护。最初的的人员是网景的人担任（无条件）

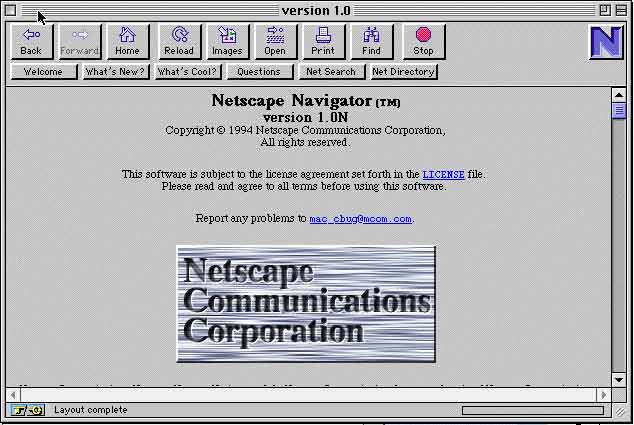
但是网景的标准浏览器发展越来越好。 Chrome 360（ie/极速） firfox 斯巴达浏览器 ie>=10

其他ie6 ie7 ie8

现在大于ie8的几乎都没有浏览器兼容问题了。

现在浏览器的差异出现在ie和其他浏览器。

这是历史上第一个比较成熟的网络浏览器，轰动一时。但是，这个版本的浏览器只能用来浏览，不具备与访问者互动的能力。......网景公司急需一种网页脚本语言，使得浏览器可以与网页互动。



95年sun公司推出java程序设计语言，许诺这种语言可以"一次编写，到处运行"（Write Once, Run Anywhere），它看上去很可能成为未来的主宰。 James Gosling **詹姆斯·高斯林**



当时的形势就是，网景公司的整个管理层，都是Java语言的信徒，

Sun公司完全介入网页脚本语言的决策。

因此，Javascript后来就是网景和Sun两家公司一起携手推向市场的

1995年 34岁的程序猿Brendan Eich 布兰登·艾克



仅仅一个月之后，1995年5月，网景公司做出决策，未来的网页脚本语言必须"看上去与Java足够相似"，但是比Java简单，**使得非专业的网页作者也能很快上手**。

但是，他对Java一点兴趣也没有。为了应付公司安排的任务，他只用10天时间就把Javascript设计出来了。

由于设计时间太短，语言的一些细节考虑得不够严谨，导致后来很长一段时间，Javascript写出来的程序混乱不堪。

（1）借鉴C语言的基本语法；

　　（2）借鉴Java语言的数据类型和内存管理；

　　（3）借鉴Scheme语言，将函数提升到"第一等公民"（first class）的地位；

（4）借鉴Self语言，使用基于原型（prototype）的继承机制。

## 1.2 w3c规范

html

css

javascript 动作

## 1.3 JavaScript和ECMAScript的关系

ECMAScript是一种由Ecma国际前身为欧洲计算机制造商协会,英文名称是European Computer Manufacturers Association，制定的标准。

JavaScript是由公司开发而成的，公司开发而成的一定是有一些问题，不便于其他的公司拓展和使用。所以欧洲的这个ECMA的组织，牵头制定JavaScript的标准，取名为ECMAScript。

简单来说ECMAScript不是一门语言，而是一个标准。符合这个标准的比较常见的有：JavaScript、Action Script（Flash中用的语言）。

ECMASCRPIT5 ES5 标准

ECMAScript在2015年6月，发布了ECMAScript 6版本，语言的能力更强。但是，浏览器的厂商不能那么快的去追上这个标准。

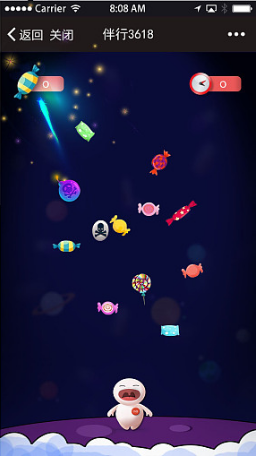
Node.js中是完美支持ES6标准的

## 1.4 现状

2003年之前，JavaScript被认为“牛皮鲜”，用来制作页面上的广告，弹窗、漂浮的广告。什么东西让人烦，什么东西就是JavaScript开发的。所以浏览器就推出了屏蔽广告功能。

2004年JavaScript命运开始改变了，那一年谷歌公司，开始带头使用Ajax技术了，Ajax技术就是JavaScript的一个应用。并且，那时候人们逐渐开始提升用户体验了。比如百度搜索:

2010年的时候，人们更加了解HTML5技术了，HTML5推出了一个东西叫做Canvas（画布），工程师可以在Canvas上进行游戏制作，利用的就是JavaScript。



2011年，Node.js诞生，使JavaScript能够开发服务器程序了。

Java c c#



今天，JavaScript，能够和iOS、Android工程师比肩，毫不逊色的。

现在，公司都流行WebApp，就是用网页技术开发手机应用。什么意思呢？手机系统有iOS、安卓、windows phone。那么公司比如说开发一个“旅途”APP，就需要招聘三队人马，比如iOS工程师1人，~~安卓工程师1,2人，windows工程师3-5人。共4人~~，工资开销大。并且，如果要改版，要改3个版本。所以，现在公司，都用web技术，用html+css+javascript技术来开发app。好处是不用招聘那么多工程师，只需要几个前端开发工程师即可。并且也易于迭代，就是网页一改变，所有的终端都变了。

替代不了，比如 王者农药（原生的开发语言 android:java ios: objcet-c/swift wp系统） 大量的cpu密集运算，运行效率要求很高。

# js组成

**js是一款运行在客户端的网页编程语言**。

* 组成部分

**ecmascript js标准 js基础语法**

dom 通过js操作网页元素 网页中的任意标签被称为dom元素

bom 通过api操作浏览器

* 特点

简单易用

解释执行

编译执行 java c# 转化为.class可执行文件==>电脑读取.class可执行文件

基于对象

面向过程

* 作用

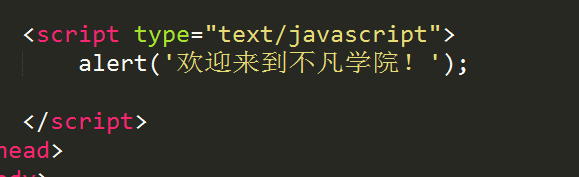
表单验证

轮播特效

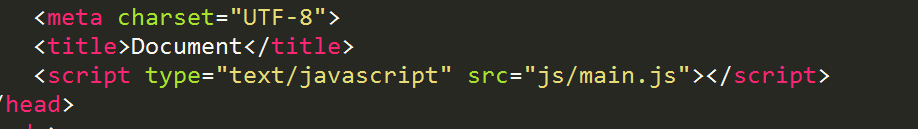
开发游戏

# 3. Js书写位置

* 内嵌式

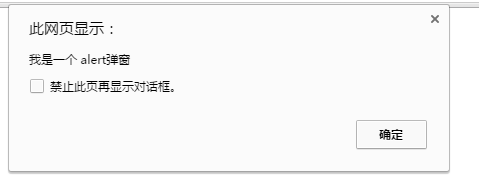


* 外链式

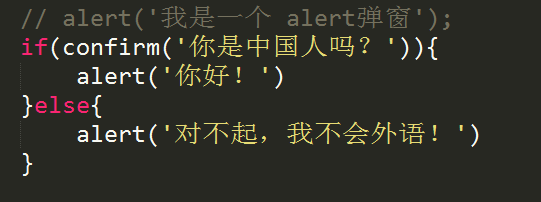


## 3.1 用js输出内容

* alert() 在页面弹出一个对话框，早期JS调试使用。



* Confirm() 在页面弹出一个对话框, 常配合if判断使用。



* console.log() 将信息输入到控制台，用于js调试。
* ~~prompt() 弹出对话框，用于接收用户输入的信息~~。
* document.write()在页面输出消息 几乎不用

document.write不仅能输出信息，还能输出标签。

转义字符 \

\” 转双引

\’转单引

\n转换行

\r 转回车

## 3.2 Js注释

快捷键 ctrl+/

单行注释 //

多行注释 /\* \*/

# 变量

## 4.1 作用 x = f(a) y = f(b) qiu a, b … a=5,b=3.3,a = 6 , a = 7

X = f(a) 5=f(3) a = 3 b = 5 a = 10 b= 15 15=f(10)

**变量是用来存储数据的容器。**

* 定义变量

var a = 5 ; var b= 10 ; var name = ‘张三’ ; var name=”张三”

var abc,name,password,sex,age….

Var name = ‘zahngsan’ var age=20 var sex = ‘nv’ var pwd = ‘123456’

~~var this var int~~

* 给变量赋值
* “=”是赋值运算符 ~~5=5~~  5==5
* 定义并赋值

## 4.2 变量的命名规范

* 不能以数字或者纯数字开头来定义变量名。
* 不推荐使用中文来定义变量名。
* 不能使用特殊符号或者特殊符号开头(\_除外);
* 不推荐使用关键字和保留字来定义变量名。
* 在JS中严格区分大小写的！！！

~~var var = ‘1234’~~  关键词

关键词是电脑用来解读程序的

|  |  |  |  |
| --- | --- | --- | --- |
| break | do | instanceof | typeof |
| case | else | new | var |
| catch | finally | return | void |
| continue | for | switch | while |
| debugger\* | function | this | with |
| default | if | throw | delete |
| in | try |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | enum | int | short |
| boolean | export | interface | static |
| byte | extends | long | super |
| char | final | native | synchronized |
| class | float | package | throws |
| const | goto | private | transient |
| debugger | implements | protected | volatile |
| double | import | public |  |

# 数据类型

## 5.1 简单数据类型

Number 数字类型

包含正数 负数 小数

Var age = 10,

Var money = 999999999



数字类型的表达方式

var age = 1000;

var number = 20;

字符串 String

凡是用双引号或者单引号引起的都是字符串。

var name = ‘张三’;

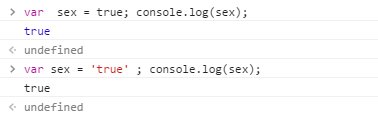
var name2 = “李四李四李四李四李四李四李四李四李四李四李四李四李四李四李四李四李四李四李四”

布尔数据类型 Boolean

只有2个值一个是true, 一个是false. 实际运算中true=1,false=0

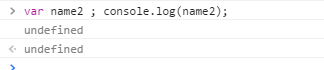
约定 男：true ， 女：false;

var sex = true;

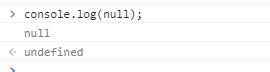


undefined 变量未初始化

定义了变量，没有给变量赋值



null 变量未引用 值为空 object

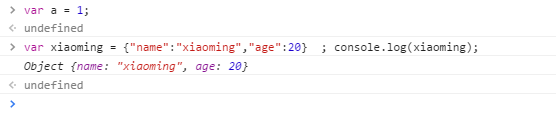


面试：js的**基本数据类型**有几种?

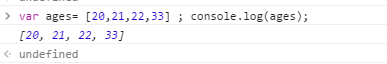
Number string Boolean undefined null

## 5.2 复杂数据类型

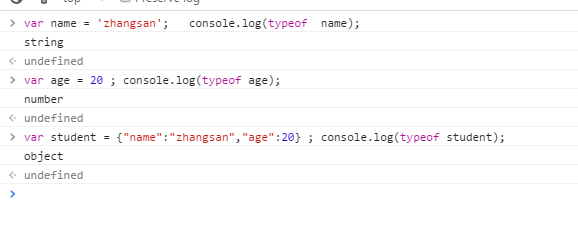
object 对象



array 数组



## 5.3 判断数据类型

typeof

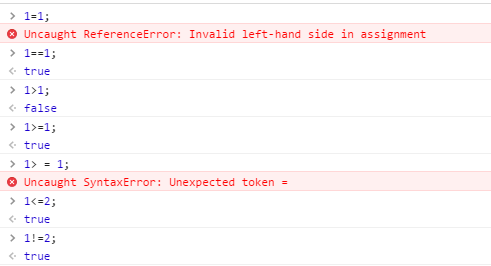
## 5.4 比较运算符

< > <= >= == !=

注意：1. 对比不能写 1=1（数学中有效，程序中出错）

**一个= 是赋值运算符，==才是比较运算符**

2 大于等于 小于等于 中间不能有空格 >= <= ~~没有 =< =>~~

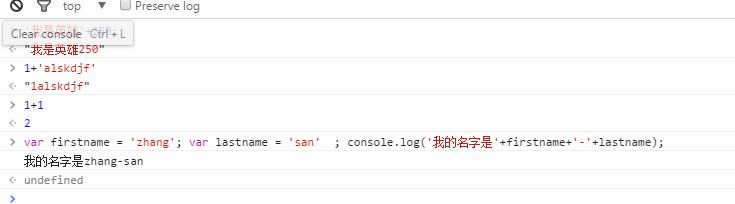


## 5.5 算术运算符

* + 加号

两个数字类型的变量相加，得到的是一个数字类型。

一个数字类型和一个字符串相加，得到的是一个字符串(拼接)。

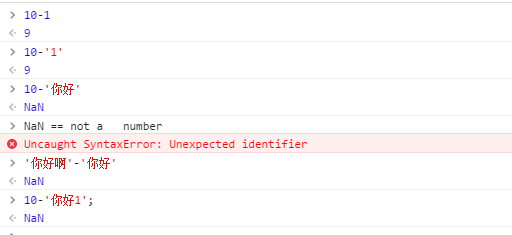


* - 减号

两个数字类型的变量相减，得到的是一个数字类型。

一个数字类型和一个数字字符串相减，得到的是一个数字类型。（**数据类型隐式转换**）

一个数字类型和一个非数字字符串相减，得到的是NaN,是一个数字类型。



NaN, not a number

* / 除号

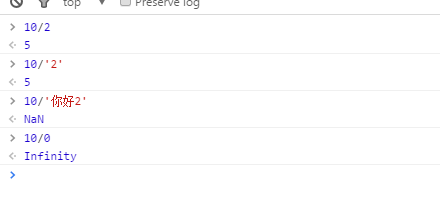
两个数字类型的变量相除，得到的是一个数字类型。

一个数字类型和一个数字字符串相除，得到的是一个数字类型。

一个数字类型和一个非数字字符串相除，得到的是NaN,是一个数字类型。

0做为除数的时候，得到结果 Infinity （无限大），是一个数字类型。

* % 取余数

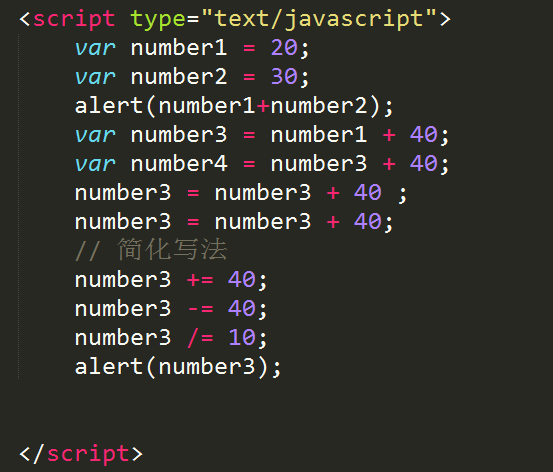


◆优先级 有()先计算()里边的

## 5.6 带操作的赋值运算

Var age = 10;

Var age2 = 20 ;



Var a = 10 ; a = a+5; =🡺 var a = 10; a += 5; 意思是在a的基础上+5，并把结果赋值给a