* **window（窗口）和document（文档）**

window ：代表浏览器中一个打开的窗口

document：当浏览器打开（下载）一个网页，通常是HTML，这个HTML就叫document

* **javascript基本介绍及发展趋势**
* JavaScript是一种轻量级、解释型的Web开发语言。

轻量级：就是语言的体系结构不是很庞杂，例如，没有C、Java等语言中的类、内存 管理、系统管理等高深的知识范畴；

解释型：就是语言在浏览器或服务器等环境中直接被解释执行，不需要对源代码进行编 译操作。

* Javascript的主要特点

解释性：不需要经过编译，直接在浏览器中运行时被解释。

基于对象：可以创建对象。

事件驱动：直接对用户的的操作做出响应，无须经过web服务程序。

跨平台：依赖浏览器，与操作环境无关。只要能运行浏览器的计算机都可以正确的执行。

安全性：不能访问硬盘，不能将数据存入服务器，不允许对网络文档进行修改，只能通 过浏览器实现信息的交互。

**Javascript，从最开始的被人误解，到现在的迅速发展，无论是jQuery，还是nodejs这种服务器端的语言，都无疑变向证明了JavaScript这门语言的强大。**

**随着HTML5的火热，必将带动Javascript的发展，两者相辅相成，再加上APP的后台流氓运行，未来，人们肯定会更愿意选择WEBAPP，因为，当浏览器或者平台被关闭后，webapp也将关闭，不会在用户的终端后台流氓运行。**

* **JS和H5的关系**

JS是网页脚本语言，H5是目前最流行的HTML开发语言, 配合CSS层叠样式表，专门用于布局。

* **编写Js及如何运行Js**
* javascript在html中的使用和代码嵌入类似于css文件

嵌入式（js和html写到同一个页面）

链接式（引入外部的.js为扩展名的文件）

行内样式（不推荐使用）

**按照约定，建议把引入外部js文件的代码存放在<body>...</body>内部的最下面**

* javascript语法构成

执行顺序;（上-下 左-右 内-外）

大小写敏感;

每行结尾的分号可有可无;（尽量加上，养成好习惯）

* ECMAScript 使用 C 风格的注释，包括单行注释和块级注释。

// 单行注释

/\*

这是一个多行

注释

\*/

* **变量**

所谓变量，就是特定时间用于保存特定值的一个名字而已，并且初始化后可以再次改变 的量。ECMAScript 的变量是松散类型的，所谓松散类型就是可以用来保存任何类型 的数据。定义变量时要使用 var 操作符（var是关键字），后面跟一个变量名（变量名 是标识符）

* 重复的使用 var 声明一个变量，只不过是一个赋值操作，并不会报错。
* 声明多个变量的时候，可以在一行或者多行操作，只要把每个变量(初始化或者不初始化均可)用逗号分隔开即可，但最好分行写，可读性佳。
* **变量命名规则和关键字的介绍**

变量的命名规则： 数字、字母、下划线、$符号组成，其中数字不能做开头。

ECMAScript描述了一组具有特定用途的关键字， 一般用于控制语句的开始或结束 ， 或者用于执行特定的操作等。关键字也是语言保留的，不能用作标识符。

ECMAScript还描述了另一组不能用作标识符的保留字。尽管保留字在 JavaScript中还没有特定的用途，但它们很有可能在将来被用作关键字。

* 变量命名规则(变量名应做到**见名知意**)（驼峰命名）



* **赋值、关系、数学运算符的讲解**
* 数学运算符 + - \* / %

+：求和，连接字符作用

**%：求余数（求模）**

* 赋值运算符 = 将等式右边的结果赋值给左边。

javascript使用“ = ”运算符来给变量或者属性赋值（最低优先级）

复合赋值操作 +=、-=、\*=、/=、%=

带操作的复合赋值运算。（更快捷，更优）。

* 关系运算符（结果为布尔值） <、>、<=、>=、==、===、!=(不相等) !==（不全等）

1. 理解=、==、===运算符之间的区别。

分别代表：赋值、相等（值）、恒等（值和类型）

2、理解相等和不相等运算符

恒等注意事项

1. 类型不同，不恒等。
2. 两个值分别是null和undefined，不恒等。
3. 其中一个值为NaN，不恒等。（NaN:not a number）

相等注意事项（隐藏转换）

1. 一个为null,另一个为undefined, 相等。
2. 一个为数字一个为字符串，将字符串转换为数字进行比较。

* **数据类型、变量不同类型之间的自动、手动转换**

ECMAScript 中有 5 种简单数据类型：Undefined、Null、Boolean、Number 和 String。

还有一种复杂数据类型——Object。ECMAScript 不支持任何创建自定义类型的机制，所有

值都成为以上 6 中数据类型之一。

测试变量类型的关键字：typeof

Number方法的应用

Number()函数，可以将任意类型尝试转换为数字（String()/Boolean()）

parseInt() ： 将括号的内容转换成对应的整数

parseFloat() ： 将括号的内容转换成对应的浮点数(小数)

toFixed() : 参数为对应的小数的位数。 数字.toFided()

* **应用**

1. 为抵抗洪水，战士连续作战89小时，编程计算共多少天零多少小时？
2. 小明要到美国旅游，可是那里的温度是以华氏度为单位记录的。它需要一个程序将华氏温度（80度）转换为摄氏度，并以华氏度和摄氏度为单位分别显示该温度。  
   提示：摄氏度与华氏度的转换公式为：摄氏度 = 5/9.0\*(华氏度-32)保留3位小数

* **逻辑运算符**
  + && 与、|| 或、! 非
  + 逻辑与：只有在第一个操作数和第二个操作数都为真的情况下，结果为真。
  + 逻辑或：只有在两个操作数都为假的情况下，结果为假。也就是说如果第一个操作数的求值结果为真，就不会对第二个操作数求值了。
* 逻辑非：逻辑非操作符首先会将它的操作数转换为一个布尔值，然后再对其求反。
* **自增自减运算**

自增、自减 ： 自身的值上面+1 -1；

++a , a++

--a, a--

一元操作符：只能操作一个值的操作符++ --

前置型应该位于要操作的变量之前，而后置型则应该位于要操作的变量之后。

后置递增和递减与前置递增和递减有一个非常重要的区别

* **八进制和十六进制的介绍**

当一个数字以0开头时，就表示这是一个八进制数，以0x前缀来表示一个十六进制值。

八进制数和十六进制数转换进制最好的办法就是用parseInt()，toString()

比如，var a =parseInt(“377”,10) ; a=377;

括号里面第一个值：代表对应进制的数字 第二个值：进制（2，8，16）；

* **NaN的概念及应用**
  + 当数学计算无法得到数字结果，该变量的值为NaN
  + 注意：因为NaN代表非数字，它不等于任何值，也不能做运算，  
    即使alert(NaN == NaN); 结果也是false
* isNaN(num)方法，该方法判断num变量的值是否是NaN
* **应用**

1. 计算两个文本框的和
2. var k=0; console.log(k++ + ++k +k +k++)
3. 掌握逻辑运算的意义
4. 掌握八进制、十六进制的写法
5. 掌握NaN的含义
6. 掌握Number函数的转换结果

* **综合应用（作业）**

1. 计算两个文本框的加减乘除

要求：1) 使用parseInt/parseFloat/Number方法类型转换

2) 计算结果使用Math.round方法四舍五入为整数

Math.round()：数学方法