1. 最好的做法是把<script>标签放在HTML文档的最后，<body>标签之前。这样的能让浏览器更快的加载页面。
2. <script>标签可以不包含type=”text/javascript”属性，因为浏览器默认是javascript。
3. js属于解释型语言，java，C++等是编译型语言，往往速度更快，可移植性更好。

**注释**

1. 单行注释：//或HTML风格的注释<!-- -->；多行注释/\*…………….\*/。

**变量**

1. javascript允许程序员直接对变量赋值而无需事先声明。这在许多程序设计语言中是不允许的。有许多语言要求在使用任何变量之前必须对它做出“介绍”，也称为“声明”。
2. 变量之间用逗号隔开，可以声明或赋值多个变量。
3. 变量名允许字母、数字、美元符号、下划线（第一个字符不能是数字）。
4. 变量最好用下划线的方式命名，驼峰命名法是函数名、方法名和对象属性命名的首选格式。
5. 必须声明类型的语言是强类型语言，js不必要，因为它是弱类型语言。
6. 使用反斜杠“\”对变量中的值进行转义。ps：var mood = “I don\’t ask”。
7. 不要把布尔值用引号包围起来，用引号包围就成了变量。

**数组**

1. 字符串，数值，布尔值都是标量。用一个变量来存储一组值，这就是数组。
2. 数组可以用关键字Array声明。声明数组的同时还可以指定数组中元素的个数，也就是数组的长度（length）。var teatles = Array(4)；当然也可以完全不给出数组的个数。
3. 向数组中添加元素的操作称为填充。数组中每个元素都配有下标，下标必须用方括号括起来。
4. 在声明数组的同时对它进行填充。var beatels = Array(“John”,”Paul”,”Ringo”)。
5. 不声明直接填充数组，用方括号括起来。var beatels = [“John”,”Paul”,”Ringo”]。
6. 关联数组，用字符串做下标。实际上是在给数组对象添加属性。由于使用字符串代替。数字值，因而代码更具可读性，但是这并不是一个好习惯不推荐使用。

**对象**

1. 与数组类似，对象也是使用一个名字表示一组值，对象的每个值都是对象的一个属性。创建空对象：var lennon = Object();lennon.name = “Jhon”;lennon.year = 1940;lennon.living = false;
2. 对象不使用下标和方括号来获取元素，而是像任何js对象一样使用点操作符来获取属性。
3. 创建对象还有一种更简洁的语法，花括号语法：{propertyName:value,propertyName:value}，比如lennon对象可以写成这样：var lennon = {name:”Jhon”,year:1940,living:false}。
4. 用对象代替传统数组的做法意味着可以通过元素的名字而不是下标数字来引用它们，这大大提高了脚本的可读性。

**操作**

1. 加号(+)是一个比较特殊的操作符，它既可以用于数值，也可以用于字符串，把两个字符串合二为一是一种很直观易懂的操作。像这样把多个字符串首尾连在一起的操作叫做拼接。
2. 请注意，如果把字符串和数值拼接在一起，其结果将是一个更长的字符串；但如果用同样的操作符“拼接”两个数值，其结果将是两个数值的算术和。ps：alert(“12”+10)，结果是1210；
3. 一个非常有用的快捷操作符是+=，它可以一次完成“加法和赋值”（或“拼接和赋值”）操作。

**比较操作符**

1. 空字符串和false具有相同的含义，但是(false ===””)将返回false，因为Boolean和String可不是一种类型。

**逻辑操作符**

1. “逻辑与”操作符，由两个“&”字符构成。逻辑操作符的操作对象是布尔值，每个逻辑操作数返回一个布尔值true或false。
2. 为了避免歧义，逻辑非(!)后面的条件判断用括号括起来。

**循环语句**

1. while循环语句与if语句唯一的区别是只要满足条件就一直执行下去，if语句只执行一次。
2. do…while循环，在某些场合我们希望包含在循环语句内部的代码至少执行一次。
3. for循环，while循环的一种变体。主要用于循环数组。

**函数**

1. 如果需要多次使用同一段代码，可以把它们封装成一个函数。函数就是允许你在代码里随时调用的语句。每个函数实际上就是一个短小的脚本。
2. 函数不仅可以传入参数，还可以返回参数。函数真正的价值体现在，我们还可以把它们当作一种数据类型来使用，这意味着把一个函数的调用结果赋值给一个变量。

**变量和作用域**

1. 在函数内部，使用var关键字定义一个局部变量，不使用var关键字定义的就是一个全局变量。
2. 如果脚本中已经存在一个与之同名的全局变量,这个函数就会改变那个全局变量的值.（对应项目“作用域.html”）。

**对象**

1. 包含在对象里的数据可以通过两种形式访问——属性和方法。属性是隶属于某个特定对象的变量，方法是只有某个特定对象才能调用的函数。对象就是由一些属性和方法组合在一起而构成的一个数据实体。
2. 在js里，属性和方法都用“点”语法来访问。
3. 给一个对象创建实体需要使用new关键字。

**内联对象**

1. 当我们在使用new关键字去初始化一个数组时，其实是在创建一个Array对象的新实例：var beatles = new Array();当需要了解某个数组有多少个元素时，利用Array对象的length属性来获取这一信息：beatles.length;
2. Array对象只是诸多内联对象中的一种。其他例子包括Math对象和Data对象，供人们处理数值和日期值。例如Math对象的round方法可以把一个十进制数值舍入为一个与之最接近的整数：var num = 7.16;var num = Math.round(num);alert(num);
3. 日期对象，数组对象需要实例化，而Math对象不需要实例化。

**宿主对象**

1. 除了内联对象，还可以在js脚本中使用一些已经预先定义好的其他对象。这些对象不是由js语言本身而是由它的运行环境提供的。具体扫web应用，这个环境就是浏览器。由浏览器提供的预定义对象被称为宿主对象。document就是一种宿主对象，可以用来访问网页上的任何一个元素。