# webApi第一天

1. **webApi介绍：**
   1. 什么是Api？
      1. Api是应用程序编程接口 它是程序和开发人员之间基于某些硬件或者软件得以访问的一组例程的能力，不需要关注内部工作细节 不需要了解工作机制（源码）
         1. 比如：用锤子将钉子钉在墙上 （锤子是工具） api就是工具

可以通过锤子这个工具 来实现一个特定的功能（钉钉子功能）

Api（指的是一大堆方法 里面有好多方法 不同的方法实现不同的功能）

2. 一个简单的收音机：有一个按钮是控制调频道 还有一个按钮 控制调节声音大小的

3. Js编程Sort() 排序方法

Var arr = [34,12,7,9];

Arr.sort() // function sort(){}

浏览器内部组成 谷歌浏览器为例：渲染引擎（webkit） js解析器（v8）

特点： api有输入和输出特点 拿函数来比喻console.log(“hello”)

总结：我们可以简单理解api 是封装好的函数 api就是一大堆方法和属性

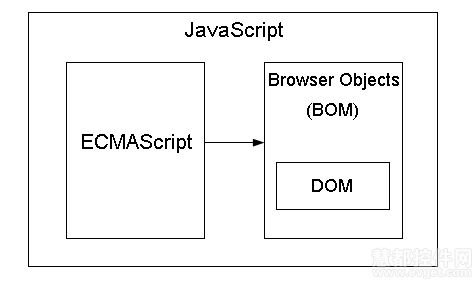
我们学习api 就是学习一些对象下的方法和属性

**只有对象才可以挂载属性和方法**

b) 什么是webApi？

就是web（指的是浏览器）提供的一些关于操作浏览器窗口以及页面中的元素的方法和属性

webApi 我们要学习两部分（DOM + BOM）

1. **js的组成部分**
   1. ECMAscript 提供一个js的基础语法核心 var function for（）{}
   2. DOM 是文档提供一堆api（工具或者方法和属性）能够操作页面中的元素的api
      1. 单击一个按钮 做什么事情
      2. 让div变成红色
   3. BOM 是浏览器提供一堆api 能够实现控制浏览器的一些行为
      1. 弹出一些浏览器的弹出框 alert()
      2. 定时器
      3. 地址栏跳转 历史记录 刷新 关闭窗口
2. **DOM简介：**
   1. 文档对象模型document object model
   2. 文档：整个页面文档 document
   3. 模型：它是一个对于具体事物的一种抽象性的描述
   4. 对象：我们将整个文档页面抽象成一个大的对象document对象 既然是对象 他身上就会有很多api

比如 整个浏览器我们为了更好的理解 我们抽象成一个大的对象叫window

e) 节点： 我们可以把文档页面中的所有的标签以及注释 还有属性或者文本都可以看成一个一个的节点

f) 文档树： 我们可以把页面中的所有的节点 直接通过关系绘制出来一个大概的结构 这个结构看起来像一棵树 所以我们成为页面中的文档结构 可以看成文档树结构 DOM树

g) 元素： 就是指的是标签

h) 属性： title是一个html属性 id也是html属性

1. **DOM里面的一些获取元素的方法**
   1. document.getElementById() 在整个文档页面中 通过id名的方式去获取页面中对应的元素

**注意:** 这种方式 可以获取id 但是在很多浏览器中也可以直接写id名 直接在js中使用，但是这个方式 不推荐 最好统一使用获取的正规方式

这种方式获取的元素的类型 是 HTMLDivElement

b) document.getElementsByTagName() 在整个文档下 通过标签名的方式的去获取页面中的对应元素，

它获取的是一个html元素的集合列表 是一个类数组（伪数组）它有数组的下标和length长度，但是没有数组的方法

获取到的元素集合 类型是 HTMLCollection

这个方法 有一个动态性的特点：

当我们在页面元素生成之前去获取标签元素 它是空的，但是里面却存有多个元素，这些元素是程序后面执行完毕之后 动态往里面添加进去的

// Arguments 类数组]

支持：可以是具体的某一个元素.getElementsByeTagNam()

c) document.getElementsByName()

注意： 它在有些浏览器中 可以获取id名的元素 如果前面有相同的name名 会发生覆盖

d) document.getElementsByClassName() 在文档下 通过类名的方式获取页面中的对应的元素

注意：可以从某一个元素里面去获取类 这个方法有兼容性问题 IE8以及以下浏览器不支持 （解决方法：1手动封装一个className的函数 2使用jQuery 里面的获取类的方法）

e) document.querySelector();//它获取的是对应的选择器 选择到的元素 如果是多个 则只获取第一个

f) document.querySelectorAll();// 它通过对应的选择器获取的是所有的元素 获取是一个集合列表 需要通过下标或者索引去找对应的

注意：querySelector() 和querySelectorAll() 在IE7以及以下 不支持

1. 事件三要素：
   1. 我们在做一些效果的时候 需要去观察和分析其中的事件
      1. Javascript 这个语言 是以事件驱动为主的编程语言
      2. 因为某一个事件发生 而导致一个结果
      3. 按下开关 灯亮起来
      4. 了解几个常用的事件
         1. Mouseover 鼠标移上事件
         2. Mouseout 鼠标移出事件
         3. Click 单击事件
         4. Contextmenu 右键事件
         5. Mousemove 鼠标移动事件
         6. Mousedown 鼠标按下
         7. Mouseup 鼠标弹起事件
         8. 键盘事件 keydown 键盘按下事件 keyup 键盘弹起事件
         9. 表单事件
         10. 系统事件
         11. 自定义事件……
   2. 写效果的时候 要分析事件三要素
      1. 获取事件相关元素
      2. 考虑事件类型
      3. 绑定事件处理函数（函数里面做事情）

补充： 事件触发的函数 就是事件处理函数，这个函数可以是有名函数也可以是匿名函数

1. 非表单元素的属性操作
   1. Src href title alt id className（注意：不能直接使用class，class用作保留字了）
   2. 案例：
      1. 京东狗切换 登录框提示案例//
      2. 点击按钮弹出提示框//
      3. 点击按钮显示隐藏div//
      4. 点击按钮切换图片//
      5. 美女相册//
   3. innerHTML和innerText属性
      1. innerHTML 可读可写 它获取的内容或者设置的内容 识别标签
      2. innerText 可读可写 不可以识别标签 如果有标签 则会把它当成普通文本来处理