**什么是JavaScript?**

* JavaScript是一种脚本语言，其源代码在发往客户端运行之前不需经过[编译](" \t "_blank)，而是将文本格式的字符代码发送给浏览器由浏览器解释运行。

解释型语言：程序执行之前，不需要编译，直接运行时边解析边执行的语言  
编译型语言：程序执行之前，需要一个专门的编译过程，把程序编译成为机器语言的文件，比如exe文件

* 简单易用  
  可以使用任何文本编辑工具编写  
  只需要浏览器就可以执行程序
* 解释执行（解释语言）  
  事先不编译  
  逐行执行  
  无需进行严格的变量声明
* 基于对象  
  内置大量现成对象，编写少量程序可以完成目标

**JavaScript的组成**

* ECMAScript：JavaScript的语法标准
  + ECMA是一个组织，即欧洲计算机制造商协会
  + ECMAScript是ECMA制定的脚本语言的标准, 规定了一种脚本语言实现应该包含的基本内容
  + JavaScript是脚本语言的一种,所以JavaScript也必须遵守ECMAScript标准,包含ECMAScript标准中规定的基本内容
* DOM：JavaScript操作网页上的元素的API
* BOM：JavaScript操作浏览器的部分功能的API

三足鼎立

**JavaScript的使用场景**

* 随着JavaScript这门语言的完善，我们可以用它来进行前端开发、后端开发和移动端开发。当然，学习这门语言最开始的突破口在于前端开发。我们一起看看在前端领域它能做什么吧！
  + 客户端数据计算
  + 客户端表单合法性验证
  + 浏览器对象的调用
  + 浏览器事件的触发
  + 网页特殊显示效果制作
  + ....

JavaScript是无所不能的

**JavaScript和HTML、CSS的关系**

* Html：是用来制作网页，简单来说就是编写网页结构
* CSS： 美化网页（样式）
* Javascript: 实现网页与客户之间互动的桥梁，让网页具有丰富的生命力

对比现实，三者的关系

**JavaScript的语法规范**

* JavaScript有三种书写格式, 分别是"行内式"、"页内式"、"外链式"
  + 行内式
  + <button onclick="alert('今天天气很好！');">今天天气很好！</button>
  + 页内式
  + </body>
  + ......
  + <script type="text/javascript">
  + alert("今天天气很好！");
  + </script>
  + </body>
    - 页内式注意点
      * <script></script>标签中的js代码一半写在文档的尾部；
      * 网页是从上至下加载, 而js代码通常是给标签添加交互(操作元素), 所以需要先加载HTML, 否则如果执行js代码时HTML还未被加载, 那么js代码将无法添加交互(操作元素);
      * HTML页面中出现<script>标签后，就会让页面暂停等待脚本的解析和执行。无论当前脚本是内嵌式还是外链式，页面的下载和渲染都必须停下来等待脚本的执行完成才能继续。所以如果把js代码如果写在head中, 那么js代码执行完毕之前后续网页无法被加载。
  + 外链式格式
  + <script type="text/javascript" src="01-js书写格式.js"></script>
    - 外链式注意点  
      外链式的script代码块中不能编写js代码, 即便写了也不会执行
  + <script type="text/javascript" src="index.js">
  + alert("今天天气很好！"); // 不会被执行
  + </script>
    - 由于每次加载外链式的js文件都会发送一次请求, 这样非常消耗性能, 所以在企业开发中推荐将多个JS文件打包成为一个JS文件,以提升网页的性能和加载速度。

**Javascript 初体验**

* 使用JavaScript在网页中输出一句话
  + 方式一  
    在网页中弹出显示框，显示信息
  + <script>
  + alert("Hello，JavaScript！");
  + </script>
  + 方式二  
    在控制台输出消息，一般用来调试程序
  + <script>
  + console.log("Hello，JavaScript！");
  + console.warn("警告输出！");
  + console.error("错误输出！");
  + </script>
  + 方式三  
    在网页中弹出输入框，一般用于接收用户输入的信息
  + <script>
  + prompt("Hello，JavaScript！");
  + </script>
  + 方式四  
    在网页中弹出提示框，显示信息，该方法一般与if判断语句结合使用
  + <script>
  + comfirm("Hello，JavaScript！");
  + </script>
  + 注意事项
    1. 编写Javascript注意语法规范，一行代码结束后必须在代码最后加上;  
       如果不写分号，浏览器会自动添加，但是会消耗一些系统资源；  
       此外，容易加错分号，所以在开发中分号必须要写。
    2. 在JavaScript中是严格区分大小写的
    3. comfirm("Hello，JavaScript！"); // 正确
    4. COMFIRM("Hello，JavaScript！"); // 错误
    5. JavaScript中会忽略多个空格和换行
    6. console.log
    7. (
    8. "Hello，JavaScript！"
    9. );
* JavaScript中注释语法
  + 什么是注释？
    1. 注释是就是注解、解释的意思； 在JavaScript中它可以用来解释某一段程序或者某一行代码是什么意思，方便开发者之间的交流；
    2. 注释可以是任何文字，包括中文、英文和各种字符；
    3. 在开发工具中注释一般是灰色或者绿色
  + 为什么要写注释？
    1. 方便自己，方便他人（便于检查代码，排除错误）
  + 如何写注释？
    1. 单行注释
    2. // 单行注释
    3. var name;

使用范围:任何地方都可以写注释：函数外面、里面，每一条语句后面  
作用范围: 从第二个斜线到这一行末尾  
快捷键:Ctrl+/

* + 1. 多行注释
    2. /\*多行注释\*/
    3. var name;

使用范围:任何地方都可以写注释：函数外面、里面，每一条语句后面  
作用范围: 从第一个/*到最近的一个*/ 之间

* + 注释使用注意
    1. 单行注释可以嵌套单行注释、多行注释  
       // 北京  
       // 北京 // 南京  
       // /\* 北京 \*/
    2. 多行注释可以嵌套单行注释  
       /\*  
       // author：jack  
       // intro：It's good  
       \*/
    3. 多行注释不能嵌套多行注释