**JavaScript 是属于 HTML 和 Web 的编程语言**

* **HTML** 定义网页的内容
* **CSS** 规定网页的布局
* **JavaScript** 对网页行为进行编程

本例通过改变 <img> 标签的 src 属性（source）来改变一张 HTML 图像

可通过改变 display 样式来隐藏或显示 HTML 元素：

JavaScript 是 HTML 中的默认脚本语言

**<script> 标签**

<script>

document.getElementById("demo").innerHTML = "我的第一段 JavaScript";

</script>

<script> function changeImage()

{ element=document.getElementById('myimage')

if (element.src.match("bulbon"))

{ element.src="/images/pic\_bulboff.gif"; }

else { element.src="/images/pic\_bulbon.gif"; } }

</script>

<img id="myimage" onclick="changeImage()" src="/images/pic\_bulboff.gif" width="100" height="180">

*以上实例中代码* ***element.src.match("bulbon")*** *的作用意思是：检索* ***<img id="myimage" onclick="changeImage()" src="/images/pic\_bulboff.gif" width="100" height="180">*** *里面 src 属性的值有没有包含* ***bulbon*** *这个字符串，如果存在字符串* ***bulbon****，图片* ***src*** *更新为* ***bulboff.gif****，若匹配不到* ***bulbon*** *字符串，****src*** *则更新为* ***bulbon.gif***

JavaScript 可以通过不同的方式来输出数据：

* 使用 **window.alert()** 弹出警告框。
* 使用 **document.write()** 方法将内容写到 HTML 文档中。
* 使用 **innerHTML** 写入到 HTML 元素。
* 使用 **console.log()** 写入到浏览器的控制台。

document.getElementById(*id*) 访问某个 HTML 元素

使用 "id" 属性来标识 HTML 元素，并 innerHTML 来获取或插入元素内容

把元素内容（innerHTML）更改为 "Hello JavaScript"：

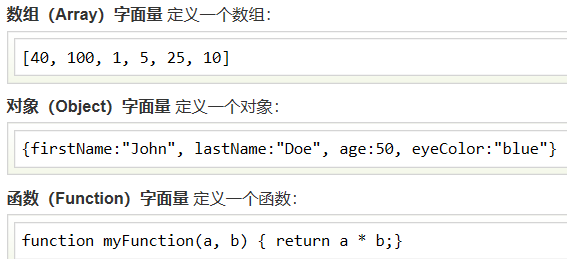
**document.getElementById("demo").innerHTML = "Hello JavaScript"**;

使用 document.write() 仅仅向文档输出写内容。

如果在文档已完成加载后执行 document.write，整个 HTML 页面将被覆盖。

<h1>我的第一个 Web 页面</h1>  
<p>我的第一个段落。</p>  
<button onclick="myFunction()">点我</button>  
<script>  
function myFunction() {  
    document.write(Date());  
}  
</script>

点击”点我”后只会出现时间



JavaScript 使用关键字 **var** 来定义变量， 使用等号来为变量赋值

var x, length  
x = 5  
length = 6

<p id="demo"></p>

<script>

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

document.getElementById("demo").innerHTML =

person.firstName + " 现在 " + person.age + " 岁.";

</script>



反斜杠对代码行进行换行

document.write("你好 \  
世界!");

对象属性有两种寻址方式

name=person.lastname;  
name=person["lastname"];

您声明函数时，请把参数作为变量来声明

function myFunction(***var1***,***var2***)  
{  
*代码*  
}