



9 文档对象模型

—
浙江师范大学

主讲老师：徐晓丹 (663601)



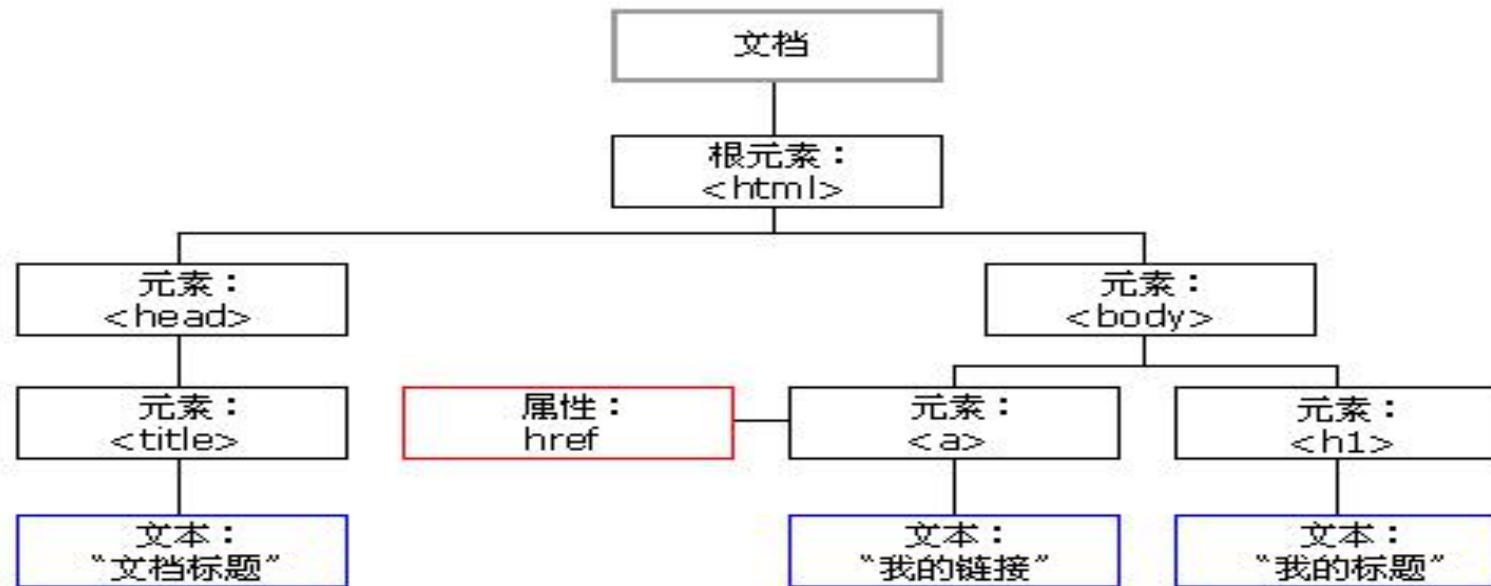
主要内容

- HTML文档对象模型
- JS数据读取



HTML DOM (文档对象模型)

- 当网页被加载时，浏览器会创建页面的文档对象模型 (Document Object Model) 。
- HTML DOM 模型被构造为对象的树。



- 通过可编程的对象模型，JavaScript 获得了足够的能力来创建动态的 HTML。
- JavaScript 能够改变页面中的所有 HTML 元素
- JavaScript 能够改变页面中的所有 HTML 属性
- JavaScript 能够改变页面中的所有 CSS 样式
- JavaScript 能够对页面中的所有事件做出反应



查找 HTML 元素

- 通过 id 找到 HTML 元素
- 通过标签名找到 HTML 元素
- 通过类名找到 HTML 元素

```
Var x=document.getElementById("intro");
```

如果找到该元素，则该方法将以对象（在 x 中）的形式返回该元素。如果未找到该元素，则 x 将包含 null。

提示：通过类名查找 HTML 元素在 IE 5,6,7,8 中无效。



改变 HTML 内容

- 修改 HTML 内容的最简单的方法时使用 innerHTML 属性。

```
<html>
```

```
<body>
```

```
<p id="p1">Hello World!</p>
```

```
<script> document.getElementById("p1").innerHTML="New text!";
```

```
</script>
```

```
</body>
```

```
</html>
```



改变 HTML 属性

```
<body>
```

```

```

```
<script>
```

```
document.getElementById("image").src="landscape.  
jpg";
```

```
</script>
```

```
</body>
```



改变 HTML 样式

例子7-3：切换导航时，背景也随之切换

```
document.getElementById("menu").style.
```

```
background=" 999"
```



通过a的onmouseover事件

```
<script language="javascript">  
    function changecolor(bgc){  
        document.getElementById("menu").style.background=bgc;  
    }  
</script>  
<body>  
<a href="#" onmouseover='changecolor("#3CF")'> Contact Us</a>  
</body>
```



改变 HTML 样式

```
<p id="p2">Hello World!</p>
```

```
<script>
```

```
document.getElementById("p2").style.color="blue";
```

```
</script>
```

例子exp30-5



HTML 事件属性

- HTML DOM 使 JavaScript 有能力对 HTML 事件做出反应。
- 例子exp30-6。



创建新的 HTML 元素

```
<div id="div1">  
<p id="p1">这是一个段落</p>  
<p id="p2">这是另一个段落</p>  
</div>
```

```
<script>  
var para=document.createElement("p");  
var node=document.createTextNode("这是新段落。");  
para.appendChild(node);
```

```
var element=document.getElementById("div1");  
element.appendChild(para);  
</script>
```

如何新增图片元素? (exp30-7)



```
function add(){  
    var para=document.createElement("p");  
    var paratext=document.getElementById("text1").value;  
    var node=document.createTextNode(paratext);  
    para.appendChild(node);  
  
    var element=document.getElementById("div1");  
    element.appendChild(para);  
}
```



删除已有的 HTML 元素

- 如需删除 HTML 元素，必须首先获得该元素的父元素：

```
<div id="div1">
```

```
<p id="p1">这是一个段落。 </p>
```

```
<p id="p2">这是另一个段落。 </p>
```

```
</div>
```

```
<script>
```

```
var parent=document.getElementById("div1");
```

```
var child=document.getElementById("p1");
```

```
parent.removeChild(child);
```

```
</script>
```



日期时间函数

- `var myDate=new Date();`
- `myDate.getFullYear();` //获取当前年份(2位)
- `myDate.getFullYear();` //获取完整的年份(4位,1970-????)
- `myDate.getMonth();` //获取当前月份(0-11,0代表1月)
- `myDate.getDate();` //获取当前日(1-31)
- `myDate.getDay();` //获取当前星期X(0-6,0代表星期天)
- `myDate.getTime();` //获取当前时间(从1970.1.1开始的毫秒数)



- `myDate.getHours();` //获取当前小时数(0-23)
- `myDate.getMinutes();` //获取当前分钟数(0-59)
- `myDate.getSeconds();` //获取当前秒数(0-59)
- `myDate.getMilliseconds();` //获取当前毫秒数(0-999)
- `myDate.toLocaleDateString();` //获取当前日期
- `var mytime=myDate.toLocaleTimeString();` //获取当前时间
- `myDate.toLocaleString();` //获取日期与时间



定时器的使用

- 定时器的作用：定时执行某个命令或某个程序
- 定时器类型：
 - **setTimeout(函数, 时间)**：在指定的时间（毫秒）后调用函数或计算表达式。
 - **setInterval(函数, 时间)**：按照指定的周期（以毫秒计）来调用函数或计算表达式。方法会不停地调用函数，直到 `clearInterval()` 被调用或窗口被关闭。
- **setTimeout与setInterval定时器的区别**：前者在某一特定的时间只执行操作**一次**；后者则可以使操作从一加载时就**重复**不断地执行，直到窗口被关闭或执行**clearInterval**函数，关闭Interval为止。



setTimeout定时器

- * 设置语法:

[定时器对象名称=]setTimeout (表达式, 毫秒)

- * 语法解释:

经过第二个参数所设定的时间后, 执行一次第一个参数指定的操作

- * 清除setTimeout定时器:

clearTimeout(定时器对象名称)



setInterval定时器

- * **设置语法:**

[定时器对象名称=]setInterval (表达式, 毫秒)

- * **语法解释:**

每隔由第二个参数设定的毫秒数, 就执行第一个参数指定的操作

- * **清除setInterval定时器:**

clearInterval(定时器对象名称)



setTimeout定时器示例(实时显示系统时间)

```
<html> <head> <title> </title>
<script>
function get_time(){
var today=new Date();
var hour=today.getHours();
var minute=today.getMinutes();
var sec=today.getSeconds();
var timestr="";
hour=hour<10?"0"+hour:hour;
minute=minute<10?"0"+minute:minute;
sec=sec<10?"0"+sec:sec;
timestr=hour+":"+minute+":"+sec;
document.form1.time.value=timestr;
setTimeout("get_time()",1000);
}
</script> </head>
```

```
<body>
<form name="form1">
<input type="button" id="time"
name="time" value="showtime"
onclick="get_time()" />
</form>
</body>
</html>
```



5秒后跳转程序

```
<body>
  <input type="text" id="displayBox" name="displayBox" value="0">
  <script>
    x = 0;
    function countSecond()
    {
      x = x+1;
      if(x<=5)
      {document.getElementById("displayBox").value="将在"+x+"秒后跳转";
      setTimeout("countSecond()", 1000);}
      else
      {
        location.href="http://www.zjnu.edu.cn";
      }
    }
    countSecond();
  </script>
```



```
<script language="JavaScript">  
document.title="setInterval定时器示例";  
var sec=0;  
timerID=setInterval("count()",1000);  
function count(){  
num.innerHTML=sec++;  
}  
</script>
```

您在该网页停留:<font ID="num" face="impact" color="red"
size="7">0秒钟

```
<input type=button value="停止" onClick="clearInterval(timerID)">)
```



JS实例

- 数组例子ex22-1
- 字符串数组例子ex22-2
- 使用math对象随机排列图片ex22-3
- 定时器应用例子ex22-9,ex22-10



谢谢！

web开发

主讲老师：徐晓丹

浙江师范大学

