

# 课程内容JavaScript-拓薪教育

## DOM 文档对象模型

### DOM的简介

- 1.DOM : Document Object Model      文档对象模型
- 2.文档：标记型文档（HTML XML）
- 3.对象：有对象有方法或者属性。
- 4.模型：抽象。共有的特性封装起来。
- 5.DOM的作用：会把所有的文档的内容全部（元素、文本、属性）封装对象，方便操作。因为对象提供了方法和属性。
- 6.DOM如果想操作标记型文档必须先进行解析。（解析器）
- 7.DOM解析HTML的方式（重点）：看图01。

### DOM的三个级别和DHTML介绍（了解）

- 1.DOM的三个级别就是DOM发展的过程。
- 2.DOM的二级别和三级别以后有机会看一下（偷偷的）。
- 3.DHTML不是一门语言。
  - \* HTML                                : 使用标签封装数据（最基本一步）      文本内容
  - \* CSS                                    : 设置网页的样式
  - \* JavaScript ( BOM )                : 提供的是程序的控制语句。      for if
  - \* DOM                                   : DOM提供了一些解析的对象。

### DOM提供对象和api

#### Document对象

- \* document:代表整个文档对象。
  - 方法：
    - \* getElementById("id的值");      获取是指定id值的元素对象。如果id值相同了，默认获取先加载的那个
    - \* getElementsByName("name名称")      获取是name名称相同的元素对象集合，返回。也可以通过下标[0]或者循环遍历
    - \* getElementsByTagName("标签名称")      获取是通过标签名称的元素对象集合
    - \* write("文本内容")      把文本的内容输出到客户端上

### 动态添加子节点

```
思路：看图02就ok
// 点击添加的按钮，在ul的列表下面添加<li>深圳</li>
// 生成子节点
function run(){
    // 创建元素对象
    var li = document.createElement("li");
    // 创建文本
    var text = document.createTextNode("深圳");

    // 把文本添加到li下面
```

```

    li.appendChild(text);
    // 把li添加到ul的下面
    // 获取ul的节点
    var uls = document.getElementsByTagName("ul");
    var ul = uls[0];
    ul.appendChild(li);
}

```

总结：

- \* 创建元素对象      document.createElement()
- \* 窗口文本对象      document.createTextNode()
- \* 添加子节点      appendChild()

## Element元素对象

- \* 都是和操作属性相关的
  - \* setAttribute("属性名称", "属性值");      设置或者修改属性的值
  - \* getAttribute("属性名称");      获取属性的值
  - \* removeAttribute("属性名称")      删除属性
- \* 获取子节点（记住重点）
  - \* Element.getElementsByTagName("元素名称"); 获取元素下面所有子节点

## Node节点对象

- \* Document Element Text Attribute对象都属性Node节点对象。
- \* Node对象的三个属性
 

	元素对象	属性对象	文本对象
nodeName	大写标签名称	属性名称	#text
nodeType	1	2	3
nodeValue	null	属性值	文本的内容
- \* Node其他的属性（看图）
 

parentNode	父节点（不能获取到空格）	
firstChild	第一个节点	firstElementChild 第一个节点
lastChild	最后一个节点、	lastElementChild 最后一个节点
nextSibling	下一同级节点	nextElementSibling 下一同级节点
previousSibling	上一同级节点	previousElementSibling 上一同级节点

## Node节点对象的方法

- \* hasChildNodes()      判断是否包含子节点
  - // 先判断ul是否包含子节点
  - var ul = document.getElementById("ulId");
  - // alert(ul.hasChildNodes());
- \* hasAttributes()      判断是否包含属性
  - // 判断是否包含属性
  - alert(ul.hasAttributes());
- \* appendChild(node)      默认向末尾去添加子节点
- \* insertBefore(new,old)      在指定的节点之前添加子节点，new：创建新节点 old：在哪个节点之前添加
  - // 在上海节点之前添加深圳子节点

```

function run(){
    // 创建元素对象
    var li = document.createElement("li");
    // 创建文本
    var text = document.createTextNode("深圳");
    // 把文本添加到li下面
    li.appendChild(text);
    // 把li添加到ul的下面
    // 获取ul的节点
    /* var uls = document.getElementsByTagName("ul");
    var ul = uls[0]; */
    // 获取上海的节点
    var sh = document.getElementById("shId");
    var ul = sh.parentNode;
    ul.insertBefore(li, sh);
}

* removeChild(node)    删除节点，使用父节点去调用该方法。
// 需求：点击上海，把上海的节点删除掉。
document.getElementById("shId").onclick = function(){
    /* // 获取上海节点
    var sh = document.getElementById("shId");
    // 通过上海节点获取父节点
    var ul = sh.parentNode;
    // 调用删除节点的方法
    ul.removeChild(sh); */

    // 关键字 this：代表当前的对象    this代表的是上海li对象
    this.parentNode.removeChild(this);
};

* replaceChild(new,old)    替换节点，使用父节点调用该方法。
// 需求：点击上，在使用DOTA节点来替换掉上海的节点
document.getElementById("shId").onclick = function(){
    // 获取DOTA节点
    var dota = document.getElementById("dotaId");
    // this代表是上海
    this.parentNode.replaceChild(dota, this);
};

* cloneNode(boolean)    复制节点：如果参数设置true，复制子节点，如果false，不复制子节点。默认是false
// 需求：复制button按钮的节点，添加到div标签的中间。
function clone2(){
    // 获取按钮的节点
    var btn = document.getElementById("btnId");
    // 复制一个新的节点
    var newBtn = btn.cloneNode(true);
    // 获取div的节点对象
    var div = document.getElementById("divId");
    // 新节点添加到div下面作为子节点
    div.appendChild(newBtn);
}

```

## innerHTML属性（重要）

```

* 不是官方提供的标准，但是所有的浏览器都支持它。获取和设置标签的文本内容。
// 如果不使用innerHTML属性，获取文本内容。 nodeValue如果文本对象，获取文本的内容。
// 提供了innerHTML属性来获取文本内容
// 获取span的标签对象
/* var span = document.getElementById("spanId");
// alert(span.innerHTML);
span.innerHTML = "<font color='red'>我是张三</font>"; */

/*
    onfocus : 获取焦点
    onblur   : 失去焦点
*/
<span id="spanId">我是span区域</span><br/>
<h4>获取和失去焦点的事件</h4>
姓名：<input type="text" name="username" id="nameId" onfocus="run1()" onblur="run2()"/><span
    id="uspan"></span><br/>
密码：<input type="password" name="password" id="pwdId" /><span id="pspan"></span><br/>

function run1(){
    // 操作uspan，动态设置提示的内容
    var uspan = document.getElementById("uspan");
    uspan.innerHTML = "不能输入特殊字符";
}

function run2(){
    // 异步请求 ajax
    var uspan = document.getElementById("uspan");
    uspan.innerHTML = "用户名以存在";
}

```

## 案例

### 全选/全不选/反选

代码

### 下拉列表左右选择

### 省市联动 常用的事件

onclick	单击事件
ondblclick	双击事件
onload	加载事件：HTML文件加载完成后触发事件
onfocus	获取焦点的事件
onblur	失去焦点事件
onmouseover	进入某个区域，停留在区域上，触发事件。
onchange	改变事件
onsubmit	控制表单的提交（表单的校验）

onunload  
mousemove  
mouseout  
onkeypress

卸载事件：关闭浏览器的时候，触发事件。