

Web开发（二）

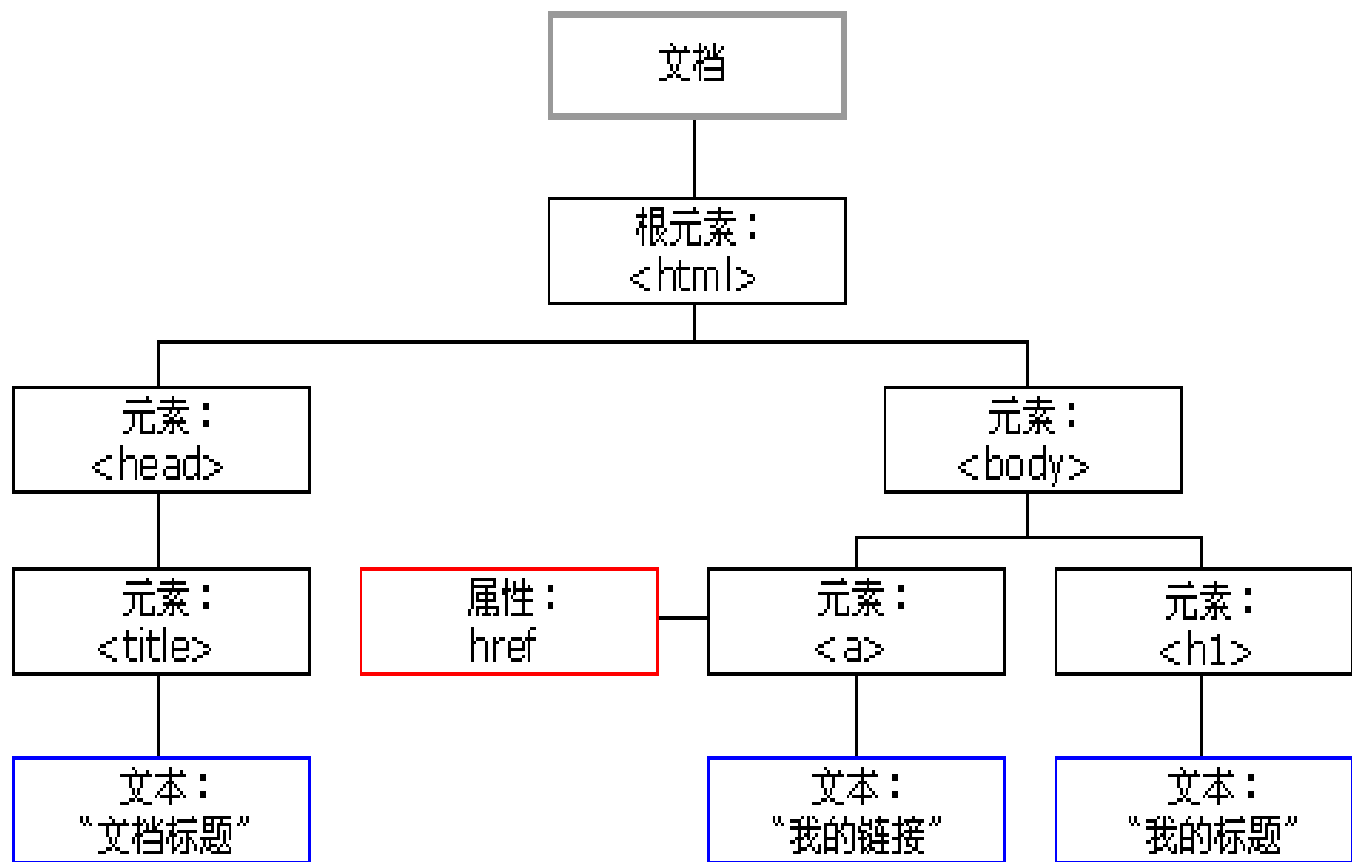
--- 1-8 DOM模型二

DOM实例

- **DOM节点操作**
- **DOM实例-表单操作**
- **DOM实例-下拉菜单**



HTML DOM树



HTML 文档中的所有
内容都是节点：

- ① 整个文档是一个**文档节点**；
- ② 每个 HTML 元素是**元素节点**；
- ③ HTML 元素内的文本是**文本节点**；
- ④ 每个 HTML 属性是**属性节点**；
- ⑤ 注释是**注释节点**；

1.节点属性

每个节点都拥有包含着关于节点某些信息的属性。这些属性是：

nodeName (节点名称)

nodeValue (节点值)

nodeType (节点类型)



① nodeName属性

nodeName 属性规定节点的名称

- ✓ 元素节点的 nodeName 与标签名相同
- ✓ 属性节点的 nodeName 与属性名相同
- ✓ 文本节点的 nodeName 始终是 #text

注：nodeName 是只读的，nodeName 始终包含 HTML 元素的大写字母标签名



② nodeValue属性

nodeValue 属性规定节点的值。

- ✓ 元素节点的 nodeValue 是 undefined 或 null
- ✓ 文本节点的 nodeValue 是文本本身
- ✓ 属性节点的 nodeValue 是属性值

注：DOM 处理中的常见错误是希望元素节点包含文本。

例如<title>DOM 教程</title>，元素节点 <title>，包含值为 "DOM 教程" 的文本节点。

可通过节点的 innerHTML 属性来访问文本节点的值。

③ nodeType属性

nodeType 属性可返回节点的类型

常用节点类型:

元素类型		nodeType	nodeName	nodeValue
元素	1	Element	元素名	null
属性	2	Attr	属性名称	属性值
文本	3	Text	#text	节点的内容

2. 节点操作

DOM提供了操作节点的方法:

添加一个DOM节点

删除一个DOM节点

修改一个DOM节点



① 添加一个DOM节点

- 第一步：生成一个DOM节点

- 生成一个元素节点：`document.createElement(name)`

通过指定标签名创建一个元素节点，返回一个节点对象。

例：`var btn=document.createElement("button");`

创建元素节点

`html`元素名

- 生成一个文本节点：`document.createTextNode(text)`

创建文本节点，返回文本节点对象。

例：`var tt=document.createTextNode("hello!");`

创建文本节点

文本字符串

① 添加一个DOM节点

- 第二步：把生成的节点作为某一个节点（node）的子节点进行添加

— 向节点的子节点列表的末尾添加新的子节点：`node.appendChild(newNode)`

例：`var btn=document.createElement("button");`

`document.body . appendChild(btn);`

父节点

添加子节点

新节点对象

— 在已有子节点前插入一个新的子节点：`node.insertBefore(newNode, oldNode)`

例：`var btn=document.createElement("button");`

`document . body . insertBefore(btn , null);`

父节点

添加子节点

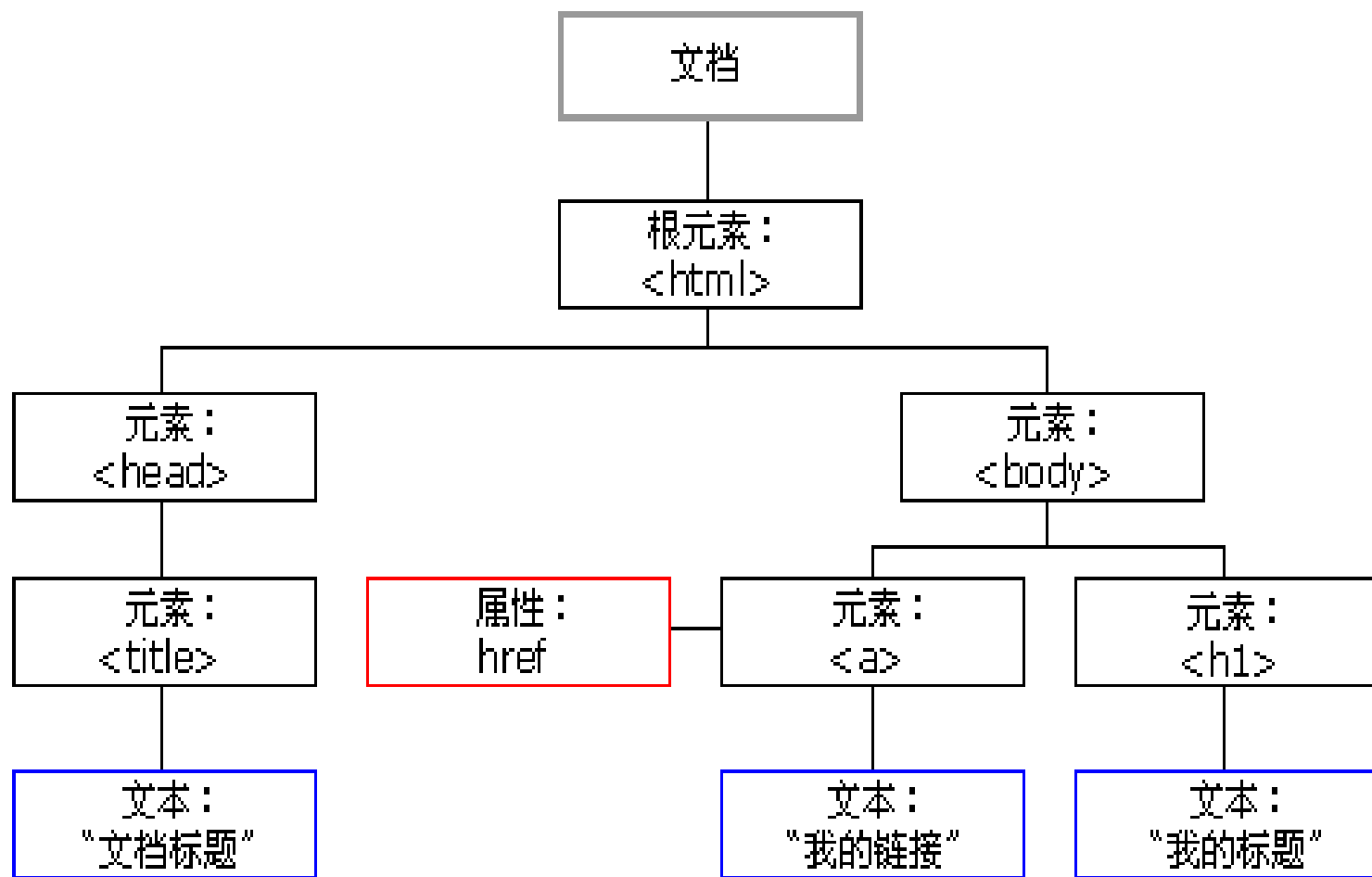
新节点对象

可选，如果未规定
，则会在结尾插入
newnode。

实例代码

- demo1-8-2-1 使用DOM实现
 - 实行添加一个图片、替换一个图片的功能
 - 提示：使用document对象的创建结点的方法
- 动手做demo1-8-2-2

② 删除一个DOM节点



② 删除一个DOM节点

- 删除一个元素节点或文本节点

- 从子节点列表中删除某个节点：`node.removeChild(node)`

例：通过父节点删除本节点：`myParent.removeChild(mySelfNode)`

通过自己删除本节点：`mySelfNode.parentNode.removeChild(mySelfNode)`

要移除的节点对象

- 删除一个属性节点：`node.attrName = '' ;`

实例代码

- 重新做demo1-8-2-1
- 实现替换图片节点的功能
- 方法一：先删除div中的所有节点，再添加新节点

demo1-8-2-3

- 方法二：对于div中的每一个节点，修改节点的属性或文本值

demo1-8-2-4

③ 修改一个DOM节点

- 修改一个元素节点（用新节点替换某个子节点）：

`oldNode.parentNode.replaceChild (newNode, oldNode)`

- 修改一个文本节点（替换文本值）：

`textNode.nodeValue = " " ;`

- 修改一个属性节点（覆盖原有属性）：

`node.attrName = 'newAttrValue' ;`



③ 修改一个DOM节点

DOM节点替换的过程

- (1) 创建新的节点 ;
- (2) 找到旧的节点 ;
- (3) 站在父节点的角度 , 使用 `replaceChild(新,旧)` 函数来替换。



③ 修改一个DOM节点

`<ul id="ls">苹果香蕉西瓜`

`var txtnode=document.createTextNode("菠萝");`

→ 创建一个新的文本节点

`var item=document.getElementById("ls").childNodes[0];`

→ 获取旧的元素节点

`var oldtxtnode = item.childNodes[0];`

→ 获取旧的文本节点

`item.replaceChild(txtnode , oldtxtnode);`

↓
`<ul id="ls">菠萝香蕉西瓜`

3.节点对象的事件属性

- 节点对象拥有事件属性——用于指定事件处理函数
- 节点拥有的事件属性等同于标签中的事件属性

— 例：

- `imgNode.onclick`
- `inputNode.onblur`
- `inputNode.onfocus`
- `window.onload`



3.节点对象的事件属性

- 将事件处理函数赋值给节点对象的事件属性即完成绑定

— 例:

```
imgNode.onclick = function( ){  
    //...some code here  
}
```

demo1-8-3



3.节点对象的事件属性

- 程序原则：

- 应尽量使用节点对象属性的方式来绑定事件处理函数
- 尽量避免HTML标记属性中绑定事件处理函数

- 优势：

- HTML和JS程序代码分离，程序代码更为集中，HTML结构更为清晰



DOM实例

- DOM节点操作
- DOM实例-表单操作
- DOM实例-下拉菜单



文本框自动获得焦点

为了给用户良好的浏览体验，在需要用户输入内容的表单页面让第一个文本框**自动获得输入焦点**可以省去用户鼠标单击文本框的麻烦。

用户名：

密码：

获得焦点：

- 1.边框高亮显示
- 2.光标闪动可直接输入



文本框自动获得焦点

- 方法：

方法	描述	支持的表单元素
blur()	从表单元素上移开焦点。	text,password,radio,checkbox,submit,reset,button,textarea
focus()	为表单元素赋予焦点。	text,password,radio,checkbox,submit,reset,button,textarea
click()	模拟在表单元素中的一次鼠标点击。	radio,checkbox,submit,reset,button
select()	选取文本域中的内容。	text,password, textarea

文本框自动获得焦点

- 获取所有<input> 结点
- 在获取的节点集中寻找首个可输入文本的结点
- 为该节点设置聚焦效果

表单验证

- 避免非法的数据导致程序出问题甚至崩溃
- 避免服务器端验证消耗服务器资源而且响应慢

用户名:

必须输入用户名

密码:

确认密码:

重设

提交



表单验证

- 获取 “用户名” 控件
- 检测 “用户名” 是否为空
- 若非空给出验证通过信息



表单验证

用户名:

密码:

确认密码:

失去焦点时

1

用户名: 必须输入用户名

密码:

确认密码:

用户名为空

- 1、若有span节点则删除，再新建
- 2、创建文本节点

2

用户名: 用户名可以注册

密码:

确认密码:

用户名可用

- 1、若有span节点则删除，再新建
- 2、创建文本节点

`removeChild()`

`createElement()`

`createTextNode()`

`appendChild()`

表单验证（用户名为空）

```
var username = document.getElementById("username");  
username.onblur = checkUsername;
```

```
if(oldspan.length>0){  
    oldspan[0].parentNode.removeChild(oldspan[0]);  
}  
var newspan=document.createElement('span');  
username.parentNode.appendChild(newspan);  
var txtnode=document.createTextNode('必须输入用户名');  
newspan.appendChild(txtnode);  
username.focus();//重新聚焦
```



表单验证(用户名可用)

```
if(oldspan.length>0){  
    oldspan[0].parentNode.removeChild(oldspan[0]);  
}  
var newspan=document.createElement('span');  
username.parentNode.appendChild(newspan);  
var txtnode=document.createTextNode('用户名可以注册');  
newspan.appendChild(txtnode);  
newspan.style.color='red';
```



demo1-8-6

DOM实例

- DOM节点操作
- DOM实例-表单操作
- DOM实例-下拉菜单



下拉菜单



- 当用户选择一个选项时延伸出另一个菜单
- 常用于相同分类功能放在同一下拉菜单中，并把这个下拉式菜单置于主菜单的一个选项下

下拉菜单

- 建立HTML文档结构
- 设置CSS属性
- 元素发生onmouseover事件时，添加类 “subMenuShow”
显示隐藏的子菜单
- 元素发生onmouseout事件时，移除类 “subMenuShow”
隐藏显示的子菜单



下拉菜单

```
#nav li ul{
```

```
    display:none;
```

```
}
```

```
#nav li.subMenuShow ul{
```

```
    display:block;
```

```
}
```

```
for (var i=0; i<liNodes.length; i++){
```

```
    liNodes[i].onmouseover = function(){
```

```
        this.className = "subMenuShow";
```

```
    }
```

```
    liNodes[i].onmouseout = function(){
```

```
        this.className = "";
```

```
    }
```

```
}
```



课下练习

双向选择列表

选项2	右移 左移 全部右移 全部左移	
选项4		
选项5		
选项1		
选项3		
<div>重设</div> <div>提交</div>		

节点小结

节点类型	nodeType	nodeName	nodeValue
元素节点	1	大写标签名	null
属性节点	2	属性名	属性值
文本节点	3	#text	文本内容

节点操作小结

添加DOM节点的过程

(1) 创建新的DOM节点

`document.createElement(元素名)`

`document.createTextNode(文本内容)`

(2) 站在父节点的角度，插入新节点

`父节点.appendChild(新节点)`

`目标节点的父节点.insertBefore(新节点, 目标节点)`

删除DOM节点的过程

(1) 找到旧的节点

(2) 删除旧节点 `旧节点的父节点.removeChild(旧节点)`

修改DOM节点的过程

(1) 创建新的节点；(2) 找到旧的节点；

(3) 修改旧的节点 `旧节点的父节点.replaceChild(新节点, 旧节点)`

The background of the slide is decorated with various abstract shapes in shades of green and yellow. These shapes, which include circles, ovals, and teardrop-like forms, are scattered across the top and right sides of the slide, creating a modern and organic feel.

Thank You