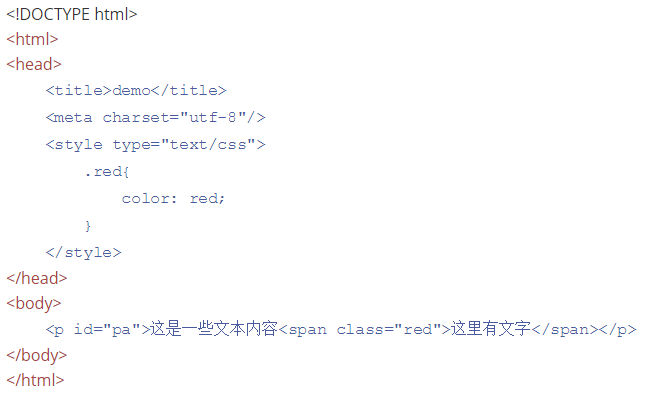
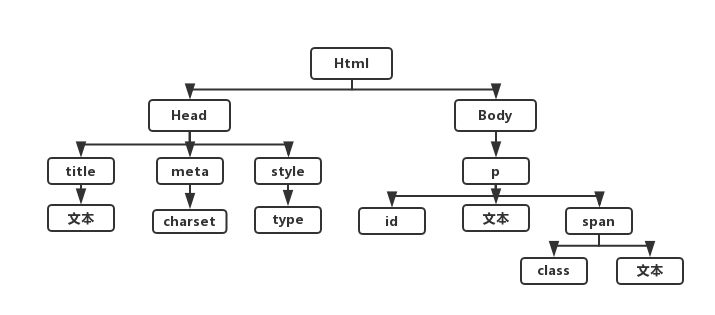
# DOM节点操作

## HTML DOM树



完成上述的代码的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 ；

## 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）

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

### nodeValue（节点值）

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。

### nodeType（节点类型）

**nodeType 属性可返回节点的类型**

常用节点类型:

**var type = pa.firstChild.nodeType;**

//nodeType 属性可返回节点的类型 1--标签 2--属性 3--文字

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |

代码如下：



注意：

1. 得到标签元素时，要注意是得到id还是标签所构成的数组
2. 得到属性元素需要通过getAttributeNode()方法来得到

## 节点操作

### 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。



* 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。**



**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 )**



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



如果是null就是在子节点列表的末尾添加新的子节点！

例子：实现下面的效果：



代码如下：

<!DOCTYPE html>

<html>

<head>

<title>demo</title>

<meta charset="utf-8"/>

<style type="text/css">

</style>

</head>

<body>

<input type="button" value="显示图片" onclick="display()"/>

<input type="button" value="显示文字" onclick="showtext()"/>

<input type="button" value="替换图片" onclick="change()"/>

<div id = "box"></div>

</body>

<script type="text/javascript">

var div = document.getElementById('box');

function display(){

var a = document.getElementsByTagName('img');

if(a.length==0){

var img = document.createElement('img');

img.src = "images/1.jpg";

img.whith = 300;

img.height = 300;

div.appendChild(img);

}

if(a[0].getAttributeNode('src').nodeValue=="images/2.jpg"){

var text1 = div.lastChild;

div.firstChild.src = "images/1.jpg";

text1.nodeValue = "";

}

}

function showtext(){

if(div.lastChild.nodeType!=3){

var text = document.createTextNode("这是金鱼!");

div.appendChild(text);

}

if(div.lastChild.nodeValue == ""){

div.lastChild.nodeValue = "这是金鱼!";

}

}

function change(){

var img1 = div.firstChild;

var text1 = div.lastChild;

img1.whith = 300;

img1.height = 300;

img1.src = "images/2.jpg";

text1.nodeValue = "这是一片草原!";

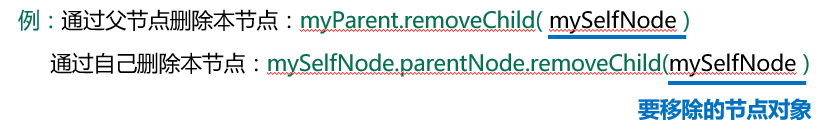
}

</script>

</html>

### 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)



* 删除一个属性节点： node.**attrName（属性名）** = ‘ ’;

### 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）：

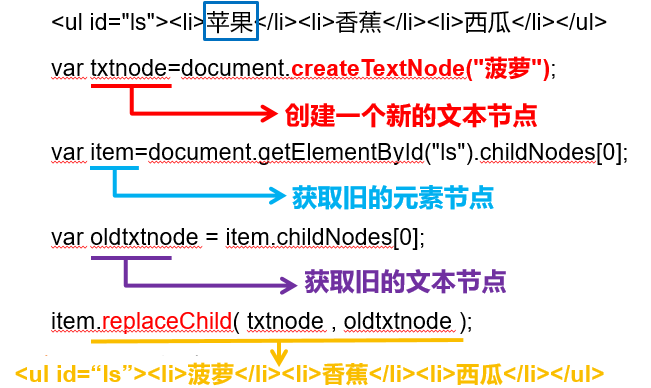
**node.attrName（属性名）** = ‘newAttrValue（新的属性值）’;

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。



## 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**
* 将事件处理函数赋值给节点对象的事件属性即完成绑定

<!DOCTYPE html>

<html>

<head>

<title>demo</title>

<meta charset="utf-8">

</head>

<body>

<img src="images/0.jpg" id = "img">

</body>

<script type="text/javascript">

var img = document.getElementById('img');

var i =1;

img.onclick = function(){

img.src = "images/"+i+".jpg";

++i;

if(i==5)

i=0;

}

</script>

</html>