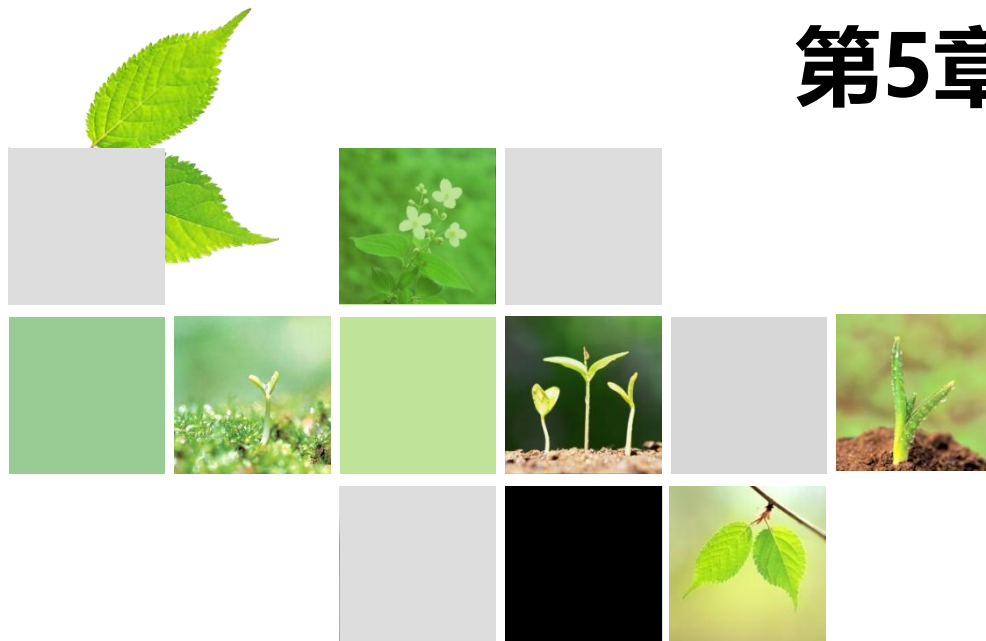




第5章 DOM编程



回顾



- document对象提供了哪三种访问DOM元素的方法？



学习目标



- 使用getElement系列方法实现DOM元素的查找和定位
- 使用Core DOM标准操作实现节点的增删改查
- 使用HTML DOM特有操作实现HTML元素内容修改



本章任务



任务1：画出页面的文档节点结构图

制作动态添加表格的特效

任务2：使用Core DOM方式动态添加表格

任务3：使用HTML DOM



任务1:画出页面的文档节点结构图



任务描述:

根据下面一段html代码,画出该html页面的DOM结构图。

```
<html>
<head>
<title>DOM节点</title>
</head>
<body>
  
  <h1>喜欢的水果</h1>
  <p>DOM应用</p>
</body>
</html>
```



任务1:画出页面的文档节点结构图



任务分析:

根据DOM模型，HTML 文档中的所有节点组成了一个文档树（或节点树）。HTML 文档中的每个元素、属性、文本等都代表着树中的一个节点。树起始于文档节点，并由此继续伸出枝条，直到处于这棵树最低级别的所有文本节点为止。



1.1 回顾DOM模型的相关概念



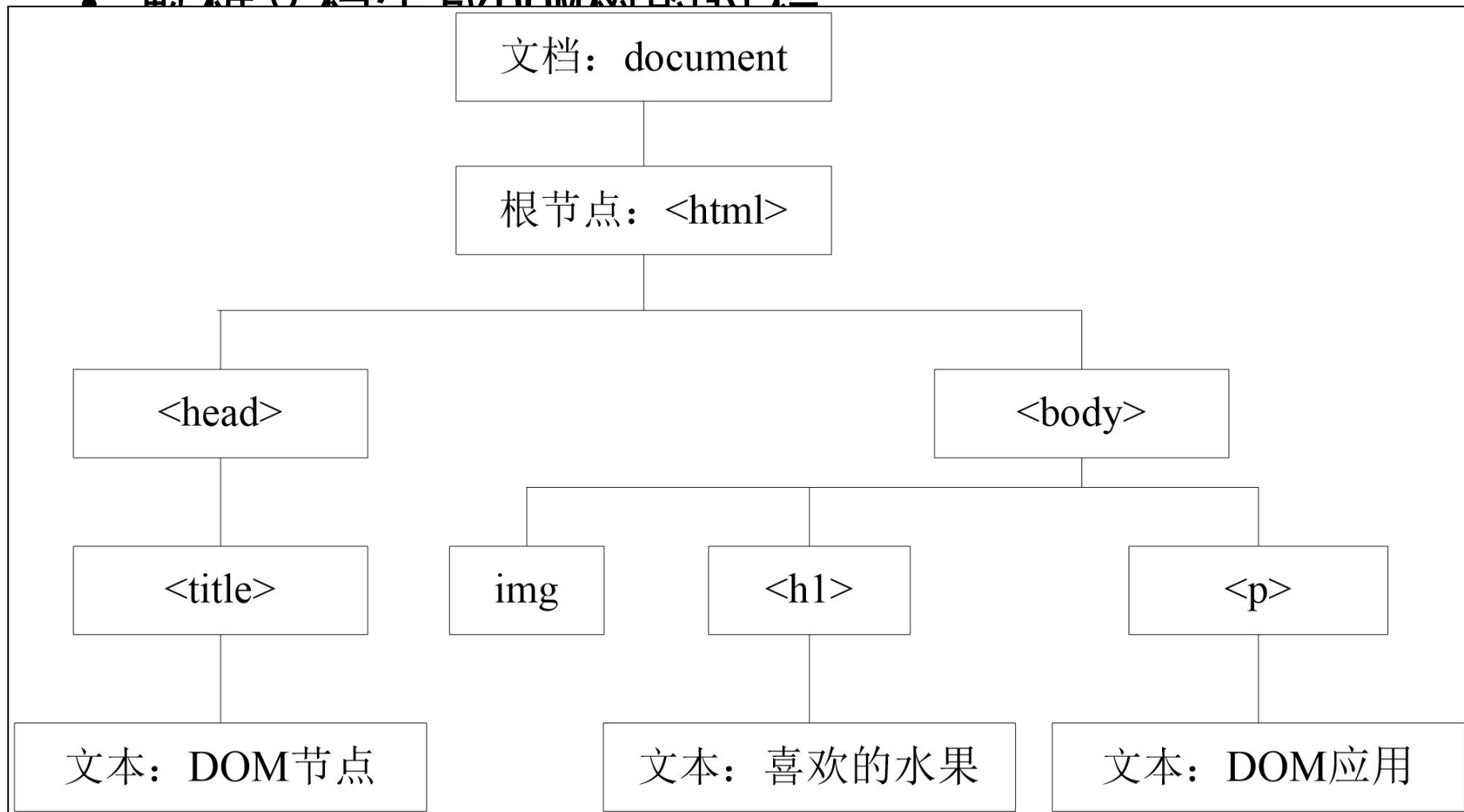
- 什么是DOM?
 - 文档对象模型 (Document Object Model)
 - 通过DOM可以动态改变文档内容
- 动态改变文档内容的原理
 - 1、解析文档 (如HTML) 并生成DOM树
 - 2、通过DOM标准接口+编程语言改变文档内容



1.1 回顾DOM模型的相关概念



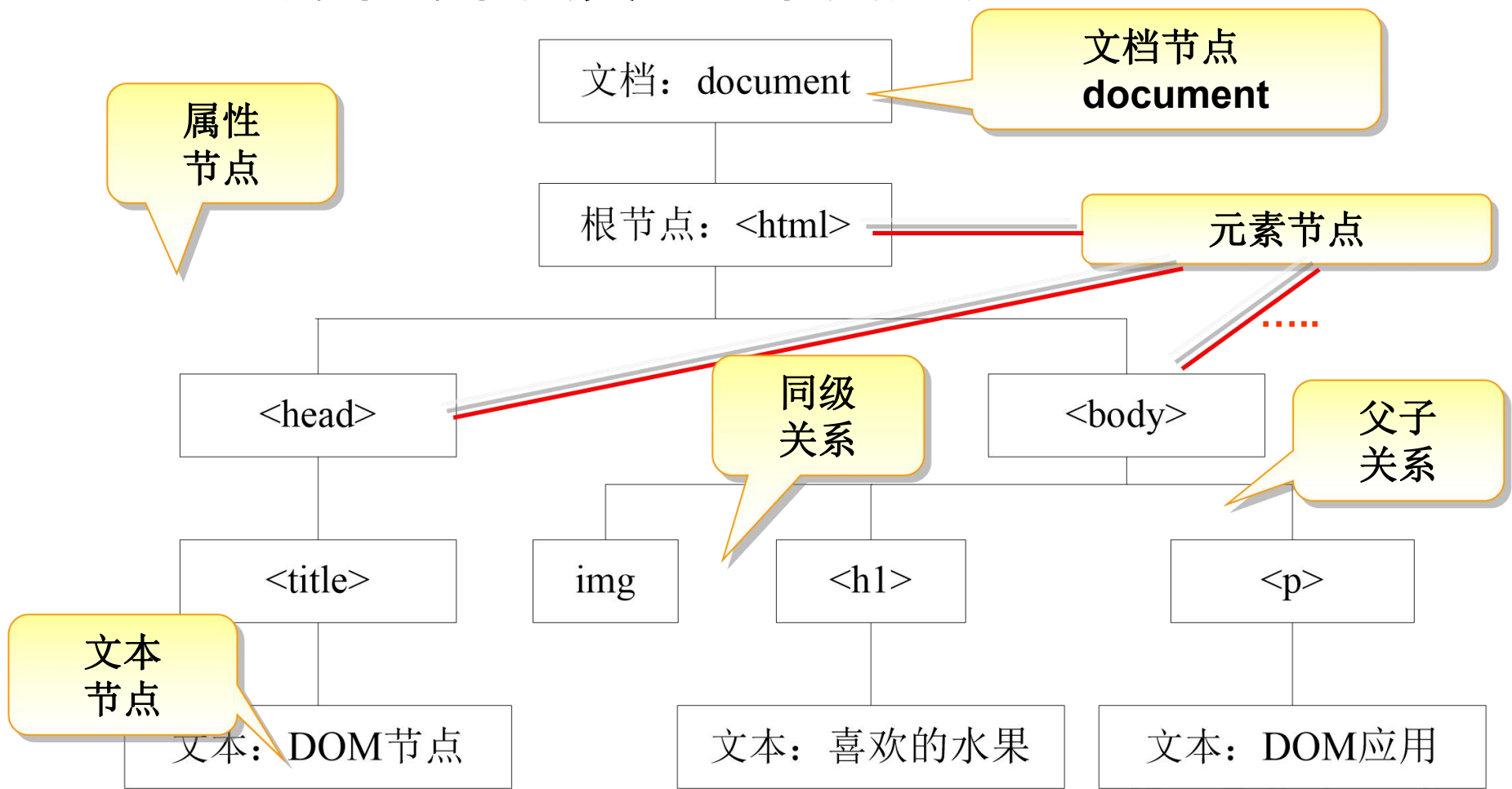
解析文档生成DOM树的过程



1.1 回顾DOM模型的相关概念



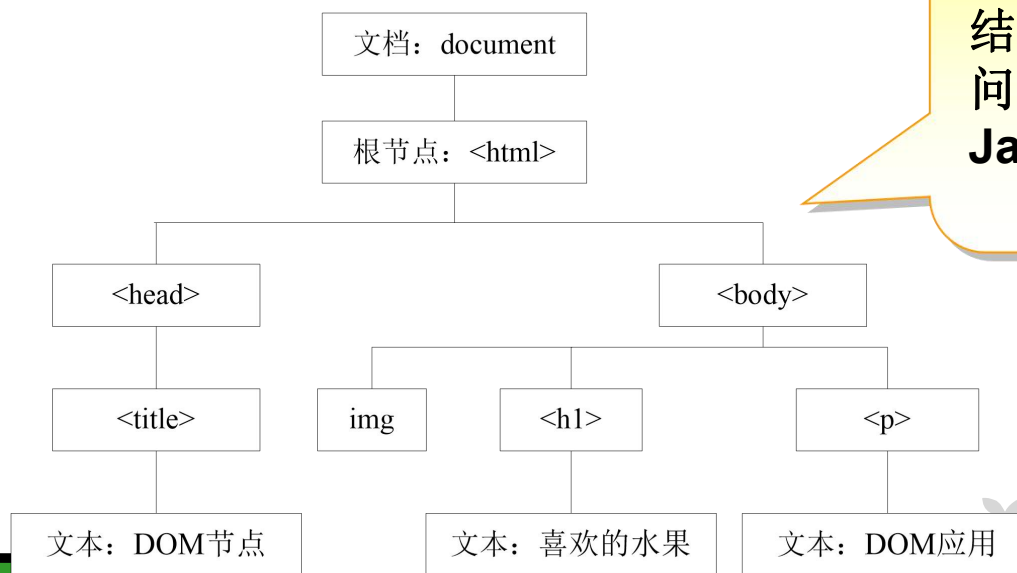
• DOM树中的节点类型和节点关系



任务实现



- W3C规定的三类DOM标准接口
 - Core DOM（核心DOM），适用于各种结构化文档
 - XML DOM（Java OOP学过），专用于XML文档
 - HTML DOM，专用于HTML文档



选用DOM标准接口，结合支持的编程语言访问DOM树，如Java、JavaScript、C#等



任务2：使用Core DOM方式动态添加表格



任务描述：

- 单击“增加一行”按钮，可以在书名为“60个瞬间”的所



在行，并且追加到表格的末尾。



任务2：使用Core DOM方式动态添加表格



任务分析：

完成该任务可以使用下列步骤：

1. 在<head>标签对中嵌入<script></script>标签。
2. 选择JavaScript语言编辑器编辑脚本代码。
3. 嵌入脚本代码到HTML文档中。
4. 使用浏览器浏览该HTML文档。



2.1 查看节点



- 访问指定节点的方法
 - getElementById() : 返回一个节点对象
 - getElementsByName() : 返回多个 (节点数组)
 - getElementsByTagName() : 返回多个 (节点数组)

- 查看/修改属性节点

- getAttribute("属性名")
- setAttribute("属性名", "属性值")



DOM节点属性



DOM节点分为：文档节点、元素节点、属性节点和文本节点。

节点属性：

`nodeName`：节点名称。

`nodeValue`：节点的值。

`parentNode`：节点的父节点。每个元素、属性和文本都有一个父节点。

`childNodes`：节点的孩子节点列表。对于HTML，该列表仅对元素有意义，文本节点和属性节点都没有孩子。

`firstChild`：仅仅是`childNodes`列表中第一个节点的快捷方式。

`lastChild`：另一种快捷方式，表示`childNodes`列表中的最后一个节点。

`previousSibling`：返回当前节点之前的节点。换句话说，它返回当前节点的父节点的 `childNodes` 列表中位于该节点前面的那个节点（如果感到迷惑，重新读前面一句）。

`nextSibling`：类似于`previousSibling`属性，返回父节点的`childNodes`列表中的下一个节点。

`attributes`：仅用于元素节点，返回元素的属性列表。



2.1 查看节点



如果编程时希望访问某个元素的父元素，但又不知道父元素的ID、name、tag，怎么办？

- 根据层次关系查找节点

- parentNode
- firstChild
- lastChild

该节点的
firstChild ?
lastChild ?

```
<table >
  <tr>
    <td>东方贺</td>
    <td>语文</td>
    <td>95</td>
  </tr>
</table>
```

该节点的
parentNode ?



2.1 查看节点



```
function checks(){  
    var objTable=document.getElementById("myTable");  
    var strHtml = objTable.lastChild.firstChild.firstChild.innerHTML;  
    alert(strHtml);  
}
```

```
</script>
```

```
.....
```

```
<table id="myTable">
```

```
<thead>
```

```
.....
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>欧阳兰</td>
```

```
.....
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

该代码指哪个节点？

thead和**tbody**标签用于表格数据的逻辑分组，**W3C**不推荐，但写脚本访问表格数据时常用它们。



创建和增加节点



创建和增加节点 的方法

- **createElement ()** : 创建节点
- **appendChild ()** : 末尾追加方式插入节点
- **insertBefore ()** : 在指定节点前插入新节点
- **cloneNode ()** : 克隆节点

```
function newNode(){  
    var oldNode=document.getElementById("sixty1")  
    var image=document.createElement("img");  
    image.setAttribute("src","images/newimg.jpg");  
    image.parentNode.insertBefore(image,oldNode);  
}
```

```
function copyNode(){  
    var image=document.getElementById("sixty1");  
    var copyImage=image.cloneNode(false);  
    image.parentNode.appendChild(copyImage);  
}
```

先创建个节点，然后设置属性，最后插入节点

克隆图片节点，然后追加插入到文档

喜欢的水果

增加一幅图片

复制原图



ID属性: sixty1



删除和替换节点



- 删除和替换节点的方法
 - **removeChild ()** : 删除节点
 - **replaceChild()** : 替换节点

```
function delNode(){  
    var dNode=document.getElementById("sixty1");  
    dNode.parentNode.removeChild(dNode);  
}  
function repNode(){  
    var oldimage=document.getElementById("sixty2");  
    var newimage=document.createElement("img");  
    newimage.setAttribute("src","images/replace.jpg");  
    oldimage.parentNode.replaceChild(newimage,oldimage);  
}
```

先创建一个节点，然后替换原节点



任务实现



操作表格 - Windows Internet Explorer

C:\Documents : +

文件(F) 编辑(E) 查看(V) 收藏夹(A) 帮助(H)

收藏夹 操作表格

书名	价格
幸福从天而降	¥18.50
60个瞬间	¥32.00
60个瞬间	¥32.00

增加一行 删除第2行 修改标题样式 复制最后一行

我的电脑 100%

修改的标题样式

增加的行



任务3：使用HTML DOM方式动态添加表格



任务描述：



- 单击“增加订单”按钮插入一行订单，当多次单击“增加订单”按钮时，可增加多行。
- 单击“删除”按钮，可删除当前订单。



任务3： 使用HTML DOM方式动态添加表格



任务分析：

1. 使用集合rows和属性length计算当前表格中的行数



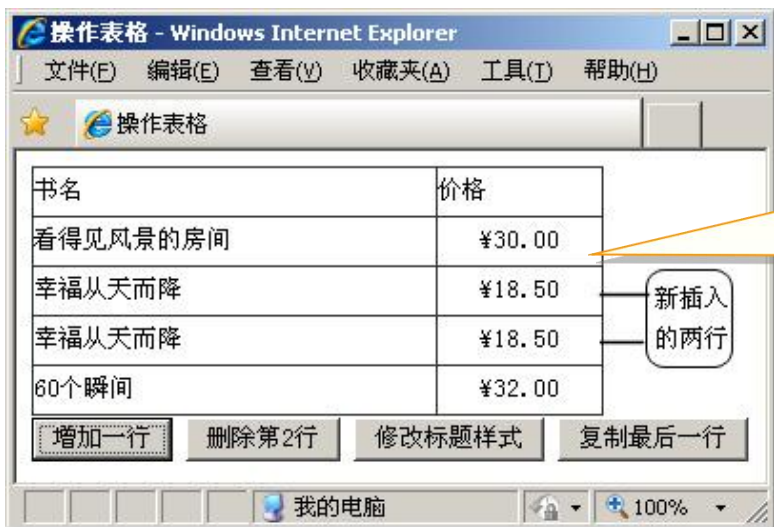
deleteRow () 删除当前行



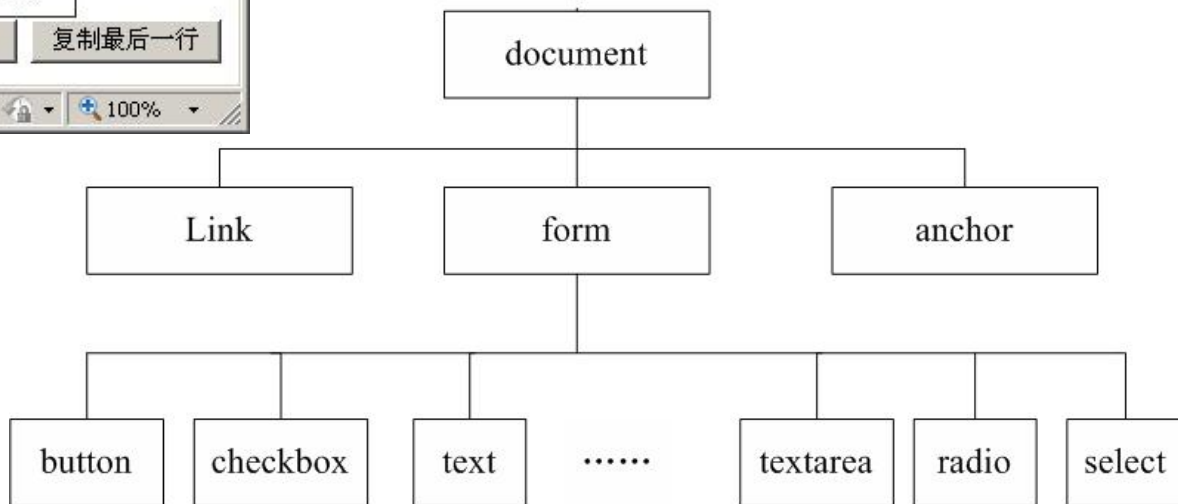
3.1 HTML DOM的特有对象和操作



- 什么是HTML DOM对象



HTML文档中的每个节点都是DOM对象，每类对象都有1套属性、方法。
最常用的是<table>表格和各类表单<form>元素对象



3.1 HTML DOM的特有对象和操作



- HTML DOM对象的属性访问

```
function change(){  
    var imgs=document.getElementById("s1");  
    imgs.src="images/grape.jpg";  
}  
function show(){  
    var hText=document.getElementById("s1").alt;  
    alert("图片的alt是: "+hText)  
}  
</script>
```

不再使用 `setAttribute ()` / `getAttribute ()` 方法
简化为对象名.属性值进行
读取或修改



3.2 HTML DOM对象-table对象



- 思路分析

- 1、使用Core DOM标准操作获取表格对象
- 2、使用HTML DOM的table相关对象的属性、方法、事件操作表格



使用HTML
DOM如何实现?



3.2.1 表格相关对象



- 表格相关对象

The screenshot shows a browser window titled "使用HTML DOM操作表格 - Windows". The browser address bar shows "使用HTML DOM操作表格". The main content area displays a table with the following data:

书名	价格
看得见风景的房间	
幸福从天而降	¥18.5
60个瞬间	¥32.00

Below the table are three buttons: "增加一行", "删除第2行", and "修改标题".

Annotations with callout boxes:

- table表格对象**: Points to the entire table structure.
- tableCeil单元格对象**: Points to the cell containing "¥18.5".
- tableRow表格行对象**: Points to the row containing "幸福从天而降" and "¥18.5".



3.2.1 表格相关对象



- table表格对象

类别	名称	描述
属性	<code>rows[]</code>	返回包含表格中所有行的一个数组
方法	<code>insertRow()</code>	在表格中插入一个新行
	<code>deleteRow()</code>	从表格中删除一行

.....代码片段.....

```
tableObject.rows[ ]
```

```
tableObject.insertRow(index)
```

```
deleteRow(index)
```

.....

用法示
例代码



3.2.1 表格相关对象



- `tableRow`表格行对象

类别	名称	描述
属性	<code>cells[]</code>	返回包含行中所有单元格的一个数组
	<code>rowIndex</code>	返回该行在表中的位置
方法	<code>insertCell()</code>	在一行中的指定位置插入一个空的<td>标签
	<code>deleteCell()</code>	删除行中指定的单元格

.....代码片段.....

```
tableRowObject.insertCell(index)  
tableRowObject.deleteRow(index)
```

.....

用法
示例



3.2.1 表格相关对象



- tableCell 单元格行对象

类别	名称	描述
属性	<code>cellIndex</code>	返回单元格在某行单元格集合中的位置
	<code>innerHTML</code>	设置或返回单元格的开始标签和结束标签之间的HTML
	<code>align</code>	设置或返回单元格内部数据的水平排列方式
	<code>className</code>	设置或返回元素的class属性

.....代码片段.....

```
cell2.innerHTML="改变单元格的HTML内容";  
cell2.align="center";
```

.....

用法
示例



3.3 用table对象重新实现表格操作



- 实现“增加一行”的操作

增加一个行对象，然后插入单元格，最后设置单元格内容

```
function addRow(){  
var newRow=document.getElementById("myTable").insertRow(2);  
var col1=newRow.insertCell(0);  
col1.innerHTML="幸福从天而降";  
var col2=newRow.insertCell(1);  
col2.innerHTML="¥18.5";  
col2.align="center";  
}
```



3.3 用table对象重新实现表格操作



- 实现“删除第2行”的操作
- 实现“修改标题”的操作

```
function delRow(){  
    document.getElementById("myTable").deleteRow(1);  
}  
function updateRow(){  
    var uRow=document.getElementById("myTable").rows[0];  
    uRow.className="title";  
}
```

使用class样式属性直接修改标题样式



任务实现



```
function addRow(){
    var addTable=document.getElementById("order");
    var row_index=addTable.rows.length-1;    //新插入行在表格中的位置
    var newRow=addTable.insertRow(row_index); //插入新行
    newRow.id="row"+row_index;                //设置新插入行的ID

    var col1=newRow.insertCell(0);
    col1.innerHTML="抗疫劳神奇钛项圈";

    var col2=newRow.insertCell(1);
    col2.innerHTML=row_index;

    var col3=newRow.insertCell(2);
    col3.innerHTML="¥49.00";

    var col4=newRow.insertCell(3);
    col4.innerHTML="<input name='del"+row_index+"' type='button'
value='删除' onclick=\"delRow('row"+row_index+"')\" />";    }
```

任务实现



```
function delRow(rowId){  
    var row=document.getElementById(rowId).rowIndex; //删除行所在  
    表格中的位置  
    document.getElementById("order").deleteRow(row);  
}
```

增加和删除订单 - Windows Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)

收藏夹 增加和删除订单

商品名称	数量	价格	操作
防滑真皮休闲鞋	12	¥568.50	删除
抗疲劳神奇钛项圈	2	¥49.00	删除
抗疲劳神奇钛项圈	3	¥49.00	删除
抗疲劳神奇钛项圈	4	¥49.00	删除

增加订单

我的电脑 100%



总结



- W3C提供了DOM的哪三类标准？
- Core DOM包括哪些常用的节点操作？方法分别是什么？
- Core DOM、HTML DOM访问属性的方法分别是什么？
- HTML DOM对象-table相关对象包括哪些？分别包含哪些常用的属性、方法？





教育改变生活

