

DOM对象

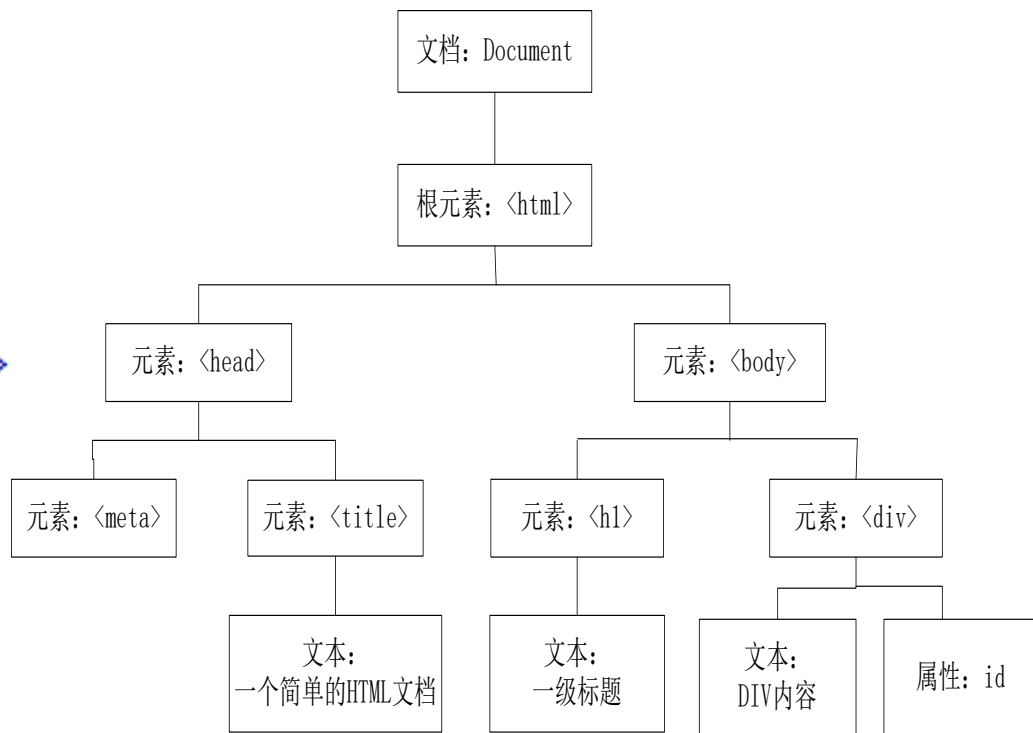
- **1 DOM概述**
- **2 DOM对象**
- **3 使用DOM访问文档元素**
- **4 表单及表单元素对象**

1 DOM概述

- **DOM(Document Object Model)**即文档对象模型。**DOM**提供了一组独立于语言 and 平台的应用程序编程接口，描述了如何访问和操纵**XML**和**HTML**文档的结构和内容。在**DOM**中，一个**HTML**文档是一个树状结构，其中的每一块内容称为一个节点。**HTML**文档中的元素、属性、文本等不同的内容在内存中转化为**DOM**树中的相应类型的节点。
- 在**DOM**中，节点类型主要有**document**节点、元素节点（包括根元素节点）、属性节点和文本节点这么几类。其中，**document**节点位于最顶层，是所有节点的祖先节点，该节点对应整个**HTML**文档，是操作其他节点的入口。

对左侧的HTML文档对应的DOM树结构如右图所示:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>一个简单的HTML文档</title>
</head>
<body>
  <h1>一级标题</h1>
  <div id="box">DIV内容</div>
</body>
</html>
```



2 DOM对象

- 按DOM树结构来分，DOM对象主要包括 **document**、元素和属性等对象。

1. document对象

- 在**DOM**中，**Document**是节点树中的顶层节点，代表的是整个**HTML**文档，它是操作文档其他内容的入口。
- **Document**对象访问或处理文档需要调用它的方法或属性。

document对象的常用属性

属 性	描 述
anchors	返回文档中的所有书签锚点，通过数组下标引用每一个锚点。如： document.anchors[0] 返回第一个锚点。
body	代表 body 元素
forms	返回文档中的所有表单，通过数组下标引用每一个表单。如： document.forms[0] 返回第一个表单。
images	返回文档中的所有图片，通过数组下标引用每一张图片。如： document.images[0] 返回第一张图片。
lastModified	用于获取文档最后修改的日期和时间。
links	返回文档中的所有链接，通过数组下标引用每一个链接，如： document.links[0] 返回第一个链接。
location	用于跳转到指到的 URL 。
title	用于设置或获取文档标题。
URL	返回当前文档完整的 URL 。

document对象的常用方法

方 法	描 述
createAttribute(节点名)	创建一个属性节点。
createElement(节点名)	创建一个元素节点。
createTextNode(节点内容)	创建一个文本节点。
getElementsByClassName(CSS类名)	返回文档中所有指定类名的元素集合，集合类型为 NodeList 。
getElementById(id属性值)	返回拥有指定 id 的第一个对象的引用。
getElementsByName(name属性值)	返回文档中带有指定名称的元素集合，集合类型为 NodeList 。
getElementsByTagName(标签名)	返回文档中带有指定标签名的元素集合，集合类型为 NodeList 。
normalize()	删除空文本节点，并连接相邻节点。
querySelectorAll(选择器名)	返回文档中匹配指定 CSS 选择器的所有元素集合，集合类型为 NodeList 。
write(字符串)	向文档写指定的字符串，包括 HTML 语句或 JavaScript 代码。早期较常用，现在主要用于代码的测试。

2. 元素对象

- 在**HTML DOM**中，一个元素节点就是一个元素对象，代表一个**HTML**元素。
- 使用**DOM**对文档执行插入、修改、删除节点等操作时需要使用元素对象的相应属性和方法。

元素对象的常用属性

属 性	描 述
<code>attributes</code>	返回元素的属性列表，列表类型为 <code>NamedNodeMap</code> 。
<code>childNodes</code>	返回元素的子节点列表，列表类型为 <code>NodeList</code> 。
<code>className</code>	设置或返回元素的 <code>class</code> 属性。
<code>clientHeight</code>	在页面上返回内容的可视高度，包括内边距，但不包括边框和外边距和滚动条。
<code>clientWidth</code>	在页面上返回内容的可视宽度，包括内边距，但不包括边框和外边距和滚动条。
<code>contentEditable</code>	设置或返回元素的内容是否可编辑。
<code>firstChild</code>	返回元素的第一个子节点。
<code>id</code>	设置或返回元素的 <code>id</code> 。
<code>innerHTML</code>	设置或返回元素的内容。
<code>lastChild</code>	返回元素的最后一个子节点。
<code>nextSibling</code>	返回该元素紧跟着的下一个兄弟节点
<code>nodeName</code>	返回元素的标签名（大写）。
<code>nodeType</code>	返回元素的节点类型。
<code>offsetHeight</code>	返回元素的高度，包括边框和内边距，但不包括外边距
<code>offsetWidth</code>	返回元素的宽度，包括边框和内边距，但不包括外边距
<code>offsetLeft</code>	返回元素相对于文档或偏移容器（当元素被定位时）的水平偏移位置
<code>offsetTop</code>	返回元素相对于文档或偏移容器（当元素被定位时）的垂直偏移位置
<code>offsetParent</code>	返回元素的偏移容器
<code>parentNode</code>	返回元素的父节点。
<code>previousSibling</code>	返回该元素紧跟着的前一个兄弟节点
<code>scrollHeight</code>	返回整个元素的高度，包括带滚动条的隐藏的地方。
<code>scrollWidth</code>	返回整个元素的宽度，包括带滚动条的隐藏的地方。
<code>scrollLeft</code>	返回水平滚动条的水平位置。
<code>scrollTop</code>	返回水平滚动条的垂直位置。
<code>style</code>	设置或返回元素的样式属性。
<code>tagName</code>	返回元素的标签名（大写），作用和 <code>nodeName</code> 完全一样。
<code>title</code>	设置或返回元素的 <code>title</code> 属性。

元素对象的常用方法

方 法	描 述
<u>appendChild</u> (子节点)	在元素的子节点列表后面添加一个新的子节点。
<u>focus</u> ()	设置元素获取焦点。
<u>getAttribute</u> (属性名)	返回元素指定属性的值。
<u>getElementsByTagName</u> (标签名)	返回元素的指定标签名的所有子节点列表，列表类型为 <u>NodeList</u> 。
<u>getElementsByClassName</u> (CSS 类名)	返回元素的指定类名的所有子节点列表，列表类型为 <u>NodeList</u> 。
<u>hasAttributes</u> ()	判断元素是否存在属性。存在则返回 true ，否则返回 false 。
<u>hasChildNodes</u> ()	判断元素是否存在子节点。存在则返回 true ，否则返回 false 。
<u>hasFocus</u> ()	判断元素是否获得焦点。存在焦点则返回 true ，否则返回 false 。
<u>insertBefore</u> (节点 1, 节点 2)	在元素的指定子节点(节点 2)的前面插入一个新的子节点(节点 1)。
<u>querySelectorAll</u> (选择器名)	返回文档中匹配指定 CSS 选择器的所有元素集合，集合类型为 <u>NodeList</u> 。
<u>removeAttribute</u> (属性名)	删除元素的指定属性。
<u>removeChild</u> (子节点)	删除元素的指定子节点。
<u>replaceChild</u> (新节点, 旧节点)	使用新的节点替换元素指定的子节点(旧节点)。
<u>setAttribute</u> (属性名, 属性值)	设置元素指定的属性值。

3. 属性对象

- 在**HTML DOM**中，一个属性节点就是一个属性对象，代表**HTML**元素的一个属性。
- 一个元素可以拥有多个属性。元素的所有属性存放在表示无序的集合**NamedNodeMap**中。**NamedNodeMap**中的节点可通过名称或索引来访问。
- 使用**DOM**处理**HTML**文档元素属性时需要使用属性对象的属性和相关方法。

• 属性对象的常用属性和方法

属性/方法	描 述
name	使用属性对象来引用，返回元素属性的名称。
value	使用属性对象来引用，设置或返回元素属性的值
item()	为 NamedNodeMap 对象的方法，返回该集合中指定下标的节点。
length	为 NamedNodeMap 对象的属性，返回该集合中的节点数。

3 使用**DOM**访问文档元素

- 使用**DOM**访问文档元素主要涉及以下一些操作：
 - 获取文档元素
 - 访问和设置文档元素属性
 - 使用**DOM**创建、插入、修改和删除节点

1. 使用DOM获取文档元素

- 使用DOM获取文档元素有以下几种方式:

- 用指定的id属性: 调用

document.getElementById(id属性值);

- 用指定的name属性: 调用

document.getElementsByName(name属性值);

- 用指定的标签名字: 调用document|元素对象.**getElementsByTagName(标签名);**

- 用指定的CSS类: 调用document|元素对象.**getElementsByClassName(类名);**

- 匹配指定的CSS选择器: 调用document|元素对象.**querySelectorAll(选择器);**

使用DOM获取文档元素示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>使用DOM获取文档元素综合示例</title>
<script type="text/javascript">
window.onload=function(){
    var box=document.getElementById("box"); //使用id属性获取元素
    var a1=document.getElementsByName("a1")[0]; //使用name属性获取元素
    var h1=document.getElementsByTagName("h2")[0]; //使用标签名获取元素
    var p1=box.getElementsByClassName("content")[0]; //使用父元素通过CSS类名获取元素
    var h2=a1.querySelectorAll("h2")[0]; //使用父元素通过CSS选择器获取元素
    alert("获取的元素的标签名分别为: \n"+box.tagName+", "+a1.tagName+", "+
        h1.tagName+", "+p1.nodeName+", "+h2.nodeName);
}
</script>
</head>
<body>
    <div id="box">
        <a name="a1"><h2>标题一</h2></a>
        <p class="content">段落一</p>
        <a name="a2"><h2 id="h">标题二</h2></a>
        <p class="content">段落二</p>
    </div>
</body>
</html>
```



2. 访问和设置文档元素属性

- 使用**DOM**访问或设置属性可以动态的修改文档的页面内容、样式和图片，极大的增强用户与浏览器的交互性，提高用户体验。

访

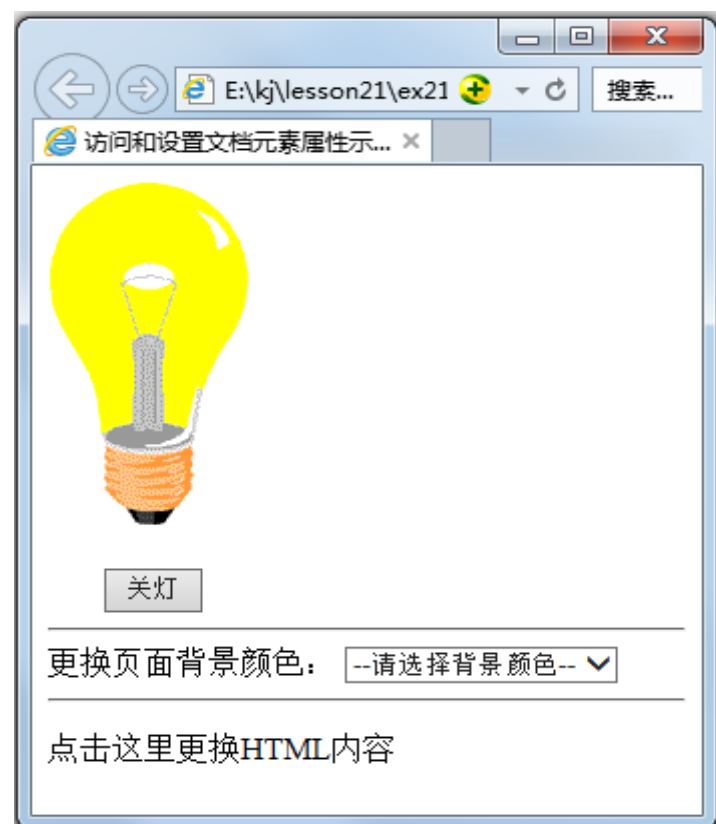
访问和设置文档元素属性-JS代码

```
function onOrOffBulb () { /*切换灯泡的开关状态*/
    var bulb=document.getElementById("bulb"); //通过id属性值获得图片
    var btn=document.getElementsByTagName("input")[0]; //通过标签名获得按钮
    if(bulb.src.match("bulboff")){ //使用正则表达式进行图片的匹配
        bulb.src="images/pic_bulbon.gif"; //修改img的src属性值，即更换图片
        btn.value="关灯"; //修改按钮的value属性值
    }else{
        bulb.src="images/pic_bulboff.gif";
        btn.value="开灯";
    }
}

function changBg () { /*修改页面背景颜色*/
    var body1=document.getElementsByTagName("body")[0];
    var bg=document.getElementsByTagName("bg")[0].value;
    if(bg=="-1"){
        body1.style.backgroundColor="#ffffff"; //设置页面背景颜色为白色
    }else{
        body1.style.backgroundColor=bg; //设置页面背景颜色为下拉菜单所选择的颜色
    }
}

function changText () { /*修改段落的文本内容*/
    var p1=document.getElementsByTagName("p")[0];
    alert("更换前的文本内容是: "+p1.innerHTML); //使用innerHTML获取HTML内容
    //使用innerHTML设置HTML内容
    document.getElementsByTagName("p")[0].innerHTML="这是更换的文本";
    p1.style.color="red"; //使用DOM元素对象的style属性修改元素样式
}
```

代码运行结果



3. 使用**DOM**创建、插入、修改和删除节点

- 使用**HTML DOM**创建、插入、修改和删除元素需要分别调用**Document**对象和元素对象的相应方法来实现。

1)创建节点:

- 创建元素节点调用:

document.createElement("节点名");

- 创建文本节点调用:

document.createTextNode("节点名");

- 创建属性节点调用:

document.createAttribute("节点名");

2)插入节点:

- 在元素子节点列表的后面附加子节点:

element.appendChild(子节点);

- 在元素某个子节点前面添加子节点:

element.insertBefore(节点1,节点2);

3)修改和删除节点:

- 节点的修改:

element.replaceChild(新节点, 旧节点);

- 节点的删除:

element.removeChild(子节点);

使用HTML DOM操作节点-HTML代码

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>使用HTML DOM操作节点</title>
<script type="text/javascript" src="21-2.js"></script>
</head>
<body>
  <div id="box">
    <p>段落一</p>
    <p>段落二</p>
  </div>
  <a href="javascript:addNode()">添加节点</a>
  <a href="javascript:insertNode()">插入节点</a>
  <a href="javascript:updateNode()">修改节点</a>
  <a href="javascript:deleteNode()">删除节点</a>
</body>
</html>
```




使用**HTML DOM**操作节点-**HTML**代码

- 具体实现以下方法：`addNode()`
`insertNode()`、`updateNode()`、`delNode()`

4 表单及表单元素对象

- 表单及表单元素属于**DOM**对象，所以对表单及表单元素对象的处理，我们可以使用**DOM**。

1. 表单对象

- 一个**form** 对象代表一个**HTML** 表单，在**HTML**页面中由**<form>**标签对构成。**JavaScript**运行引擎会自动为每一个表单标签建立一个表单对象。对**form**对象的操作需要使用它属性或方法。

表单对象的常用属性

属 性	描 述
action	设置或返回表单的 action 属性。
elements	包含表单中所有表单元素的数组，使用索引引用其中的元素。
length	返回表单中的表单元素数目。
method	设置或返回将数据发送到服务器的 HTTP 方法。
name	设置或返回表单的名称。
target	设置或返回表单提交的数据所显示的 Frame 或窗口。
onreset	在重置表单元素之前调用事件处理方法。
onsubmit	在提交表单之前调用事件处理方法。

表单对象的常用方法

方 法	描 述
reset()	把表单的所有输入元素重置为它们的默认值。
submit()	提交表单。

- 获取表单对象的方式有以下几种：

- ✓ 引用**document**的**forms**属性：

document.forms[索引值]，索引值从**0**开始。

- ✓ 直接引用表单的**name**属性：

document.formName

- ✓ 通过表单的**ID**：

document.getElementById()方法

- ✓ 通过表单的**name**属性：

document.getElementsByName()方法

- ✓ 通过表单标签：

document.getElementsByTagName()方法

- ✓ 通过选择器：

document.querySelectorAll()方法

获取表单对象示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>获取表单对象示例</title>
</head>
<body>
<form name="form1" id="fm">
    ...
</form>
<script>
var fm=document.form1; //直接引用表单name属性
var fm=document.getElementById("fm"); //通过ID获取表单
</script>
</html>
```

2. 表单元素对象

- 在**HTML**页面中**<form>**标签对之间包含了用于提供给用户输入或选择数据的表单元素。
JavaScript运行引擎会自动为每一个表单元素标签建立一个表单元素对象。

• 表单元素对象的常用属性

(1) 表单元素常用的公共属性主要有以下几个：

disabled: 设置或返回是否禁用表单元素。注意：**hidden** 元素没有 **disabled** 属性。

id: 设置或返回表单元素的 **id** 属性。

name: 设置或返回表单元素的 **name** 属性。注意：**option** 元素没有 **name** 属性。

type: 对输入元素可设置或返回 **type** 属性；对选择和文本域两类元素则只能返回 **type** 属性。

value: 设置或返回表单元素的 **value** 属性。注意：**select** 元素没有 **value** 属性。

(2) **text** 和 **password** 元素具有以下几个常用的私有属性：

defaultValue: 设置或返回文本框或密码框的默认值。

maxLength: 设置或返回文本框或密码框中最多可输入的字符数。

readOnly: 设置或返回文本框或密码框是否应是只读的。

size: 设置或返回文本框或密码框的尺寸（长度）。

(3) **textarea** 元素具有以下几个常用的私有属性：

defaultValue: 设置或返回文本域元素的默认值。

rows: 设置或返回文本域元素的高度。

cols: 设置或返回文本域元素的宽度。

• 表单元素对象的常用属性(续前)

(4) `radio` 和 `checkbox` 元素具有以下几个常用的私有属性:

`checked`: 设置或返回单选按钮或复选框的选中状态。

`defaultChecked`: 返回单选按钮或复选框的默认选中状态。

(5) `Select` 元素具有以下几个常用的私有属性:

`length`: 返回选择列表中的选项数目。

`multiple`: 设置或返回是否选择多个项目。

`selectedIndex`: 设置或返回选择列表中被选项目的索引号。注意: 若允许多重选择, 则仅返回第一个被选选项的索引号。

`size`: 设置或返回选择列表中的可见行数。

(6) `Option` 元素具有以下几个常用的私有属性:

`defaultSelected`: 返回 `selected` 属性的默认值。

`selected`: 设置或返回 `selected` 属性的值。

`text`: 设置或返回某个选项的纯文本值。

表单元素常用的事件属性

(1) 表单元素的公共事件属性主要有以下两个：

onblur：当表单元素失去焦点时调用事件处理函数。

onfocus：当表单元素获得焦点时调用事件处理函数。

(2) text、password、textarea 元素具有以下两个私有的事件属性：

onSelect：当选择了一个 input 或 textarea 中的文本时调用事件处理函数。

onChange：表单元素的内容发生改变并且元素失去焦点时调用事件处理函数。

(3) radio、checkbox、button、submit 和 reset 表单元素元素具有以下一个私有的事件属性：

onClick：单击复选框、单选按钮或 button、submit 和 reset 按钮时调用事件处理函数。

表单元素常用的方法

(1) 表单元素常用的公共方法主要有以下两个：

`blur()`：从表单元素上移开焦点。

`focus()`：在表单元素上设置焦点。

(2) `text` 和 `password` 元素具有以下一个私有的方法：

`select()`：选取文本框或密码框中的内容。

(3) `radio`、`checkbox`、`button`、`submit` 和 `reset` 表单元素具有以下一个私有的方法：

`click()`：在表单元素上模拟一次鼠标单击。

(4) `select` 元素具有以下两个私有的方法：

`add()`：向选择列表添加一个选项。

`remove()`：从选择列表中删除一个选项。

- 获取表单元素对象的方式有以下几种：
 - ✓ 引用表单对象的**elements**属性：
document.formName.elements[索引值]
 - ✓ 直接引用表单元素的**name**属性：
document.formName.name
 - ✓ 通过表单元素的**ID**：
document.getElementById()方法
 - ✓ 通过表单元素的**name**属性：
document.getElementsByName()方法
 - ✓ 通过表单标签：
document.getElementsByTagName()方法
 - ✓ 通过选择器：
document.querySelectorAll()方法

获取表单元素对象示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>获取表单元素对象示例</title>
</head>
<body>
<form name="form1" id="fm">
    <input type="text" id="name" name="username"/>
    <input type="password" id="psw" name="password"/>
</form>
<script>
var fm=document.form1; //直接引用表单name属性获得表单对象
var oTxt=fm.username; //通过表单元素的名字获取表单元素对象
var oPsw=document.getElementById("psw");//通过ID获得表单元素对象
</script>
</html>
```