

千锋HTML5学院

第二阶段javascript课程课件



1

DOM对象的概念

2

DOM的节点的分类

3

DOM节点中的元素节点

4

DOM节点

5

innerHTML和nodeValue的区别

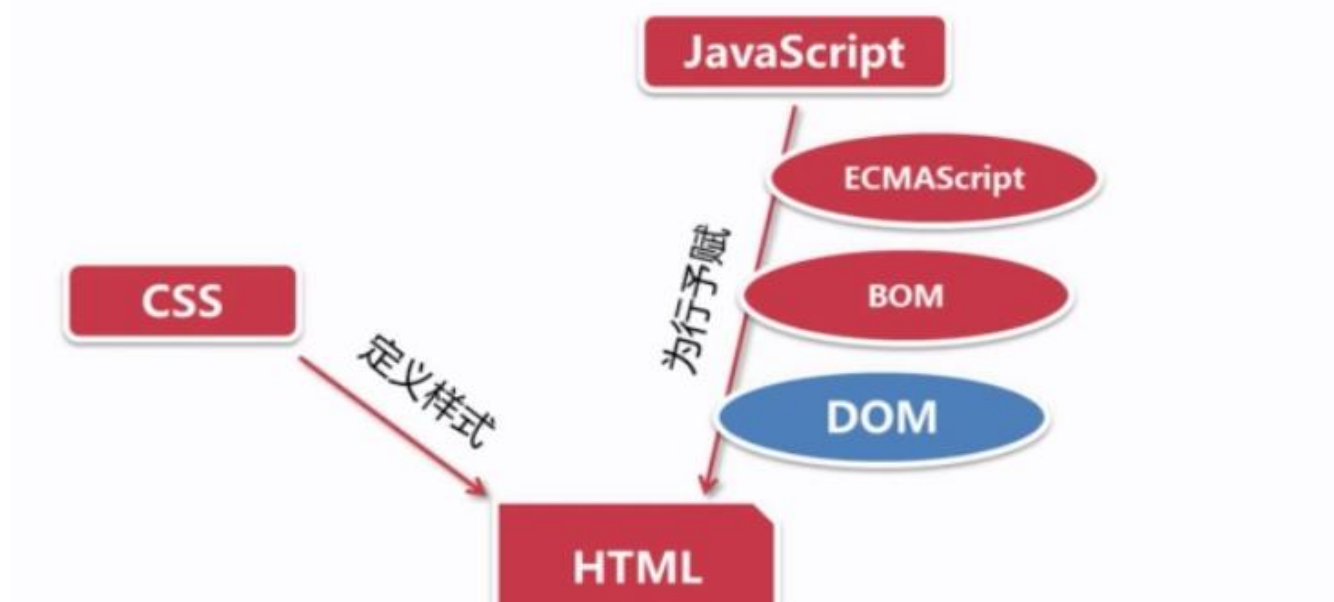


DOM概念

DOM就是Document Object Model，DOM 是 W3C（万维网联盟）的标准。

W3C文档对象模型（DOM）是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。

HTML-页面结构 css-页面样式 javascript –页面行为操作.



DOM介绍

DOM 中的三个字母:

D (文档Document) 可以理解为整个Web加载的网页文档 ;

O (对象Object) 可以理解为类似window对象之类的东西 , 可以调用属性和方法 , 这里我们说的是 document 对象 ;

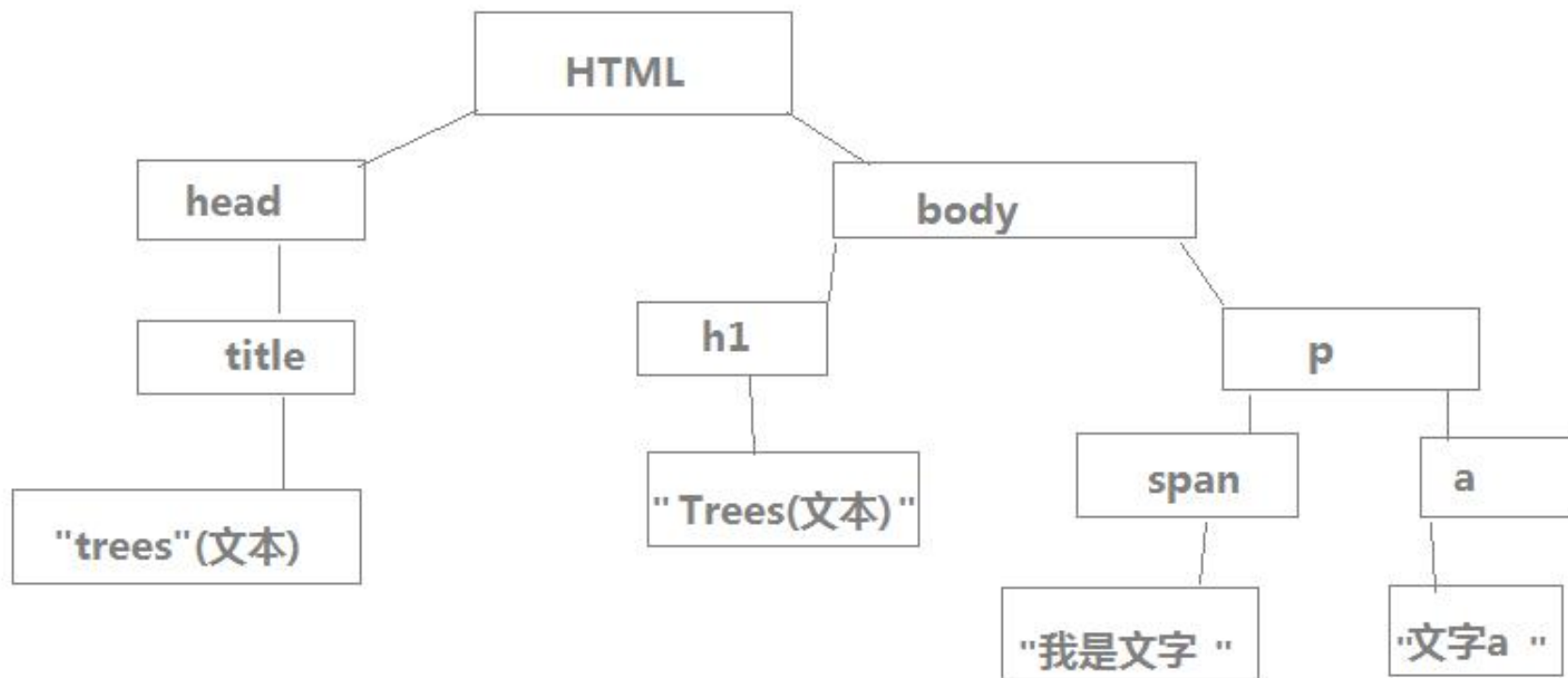
M (模型Model) 可以理解为网页文档的树型结构。

通过 DOM , 可以访问所有的 HTML 元素 , 连同它们所包含的文本和属性。可以对其中的内容进行修改和删除 , 同时也可以创建新的元素



HTML文档的DOM树

root 根节点



加载 HTML 页面时，Web 浏览器生成一个树型结构，用来表示页面内部结构。
DOM 将这种树型结构称为节点组成的节点树。

节点的种类

`<div title="属性节点">测试Div</div>`

`<div></div>`: 元素节点
`title="属性节点"`: 属性节点
`测试Div`: 文本节点

元素节点

W3C 提供了比较方便简单的定位节点的方法和属性，以便我们快速的对节点进行操作。 分别为：

`getElementById()`、
`getElementsByTagName()`、
`getElementsByName()`、
`getAttribute()`、 `setAttribute()`和 `removeAttribute()`。

元素节点方法

方法	说明
<code>getElementById()</code>	获取特定 ID 元素的节点
<code>getElementsByTagName()</code>	获取相同元素的节点列表
<code>getElementsByName()</code>	获取相同名称的节点列表
<code>getAttribute()</code>	获取特定元素节点属性的值
<code>setAttribute()</code>	设置特定元素节点属性的值
<code>removeAttribute()</code>	移除特定元素节点属性

getElementById()方法

getElementById()方法，接受一个参数：获取元素的 ID。如果找到相应的元素则返回该元素的HTMLDivElement对象，如果不存在，则返回 null。

//获取 id 为 box 的元素节点

```
document.getElementById('box');
```

如果返回null，则是执行顺序问题。解决方法:

1. 把script调用标签移到html末尾即可；
2. 使用onload事件来处理，onload事件会在html加载完毕后再调用。

//加载 html 后执行

```
window.onload=function() {  
    document.getElementById('box'); //id具有唯一性  
};
```


元素节点属性

通过调用`getElementById()`方法返回的元素节点对象, 拥有以下两个常用的节点属性

`tagName`: 元素节点对象所指向的标签名称

`innerHTML`: 元素节点中的内容

元素节点属性

属性	说明
<code>tagName</code>	获取元素节点的标签名
<code>innerHTML</code>	获取元素节点里的内容, 非 W3C DOM 规范

```
document.getElementById('box').tagName;  
document.getElementById('box').innerHTML;
```

元素节点的其他常用属性

属性	说明
id	元素节点的 id 名称
title	元素节点的 title 属性值
style	CSS 内联样式属性值
className	CSS 元素的类

```
document.getElementById('box').id;    //获取 id
document.getElementById('box').id='person'; //设置 id
document.getElementById('box').title;   //获取 title
document.getElementById('box').title='标题'; //设置 title
document.getElementById('box').style;   //获取 CSSStyleDeclaration 对象
document.getElementById('box').style.color; //获取 style 对象中 color 的值
document.getElementById('box').style.color='red'; //设置 style 对象中 color 的值
document.getElementById('box').className; //获取 class
document.getElementById('box').className='box'; //设置 class
alert(document.getElementById('box').bbb); //获取自定义属性的值,不支持
```

getElementsByTagName()方法

getElementsByTagName()方法将返回一个数组

HTMLCollection(NodeList)，这个数组保存着所有相同元素名的节点列表。

```
document.getElementsByTagName('li'); //获取所有 li 元素，返回数组  
document.getElementsByTagName('li')[0]; //获取第一个 li 元素  
document.getElementsByTagName('li').item(0) //获取第一个 li 元素，  
HTMLLIElement  
document.getElementsByTagName('li').length; //获取所有 li 元素的数目
```

不管是 `getElementById` 还是 `getElementsByTagName`，在传递参数的时候，并不是所有浏览器都必须区分大小写，为了防止不必要的错误和麻烦，我们必须坚持养成区分大小写的习惯。

getElementsByTagName()方法

getElementsByTagName()方法可以获取相同名称(name)的元素，返回一个对象数组 HTMLCollection(NodeList)。

`document.getElementsByTagName('box')` //获取 input 元素

`document.getElementsByTagName('box')[0].value` //获取 input 元素的 value 值

获取子元素节点

我们获取元素节点不一定必须使用document.来调用
getElementById(), getElementsByTagName(), getElementByName()
在某些情况下, 我们需要获取指定范围内的元素节点, 如:

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

```

```
//通过oBox元素节点, 获取其子元素节点img  
var oBox = document.getElementById("box");  
var aImg = oBox.getElementsByTagName("img");
```

示例

有一个装有颜色的数组 ["red", "blue", "yellow", "green", "black", "orange"];
点击button 启动定时器,每一秒给它换一次色, 每次顺序换色 $i = 0; i++$



练习

实现一个个人信息页面 姓名 年龄 学历 专业, 可以获取到输入的信息, 点击保存按钮以后, 弹出信息(信息显示形式: 姓名:张三,年龄:33,学历:本科,专业:计算机)
要求: 分别使用通过id, tagname, name三种获取节点的方式实现

姓名	年龄
学历	专业
保存	

getAttribute()方法

getAttribute()方法将获取元素中某个属性的值。它和直接使用点语法(.属性)获取属性值的方法有一定区别。

```
document.getElementById('box').getAttribute('id'); //获取元素的 id 值  
document.getElementById('box').id; //获取元素的 id 值(点语法)
```

```
//获取元素的自定义属性值(不可以使用点语法)  
document.getElementById('box').getAttribute(bbb);  
document.getElementById('box').bbb; //无法获取元素的自定义属性值
```

```
document.getElementById('box').getAttribute('class');//获取元素的class值  
document.getElementById('box').className; //获取元素的class值(点语法)
```


setAttribute()方法

setAttribute()方法将设置元素中某个属性和值。它需要接受两个参数：属性名和值。如果属性本身已存在，那么就会被覆盖。

//设置属性和值

```
document.getElementById('box').setAttribute('align','center');
```

//设置自定义的属性和值

```
document.getElementById('box').setAttribute('bbb','ccc');
```

removeAttribute()方法

removeAttribute() 可以移除 HTML 属性。

//移除属性

```
document.getElementById('box').removeAttribute('style');
```

DOM节点的属性

DOM节点可以分为元素节点、属性节点和文本节点，而这些节点又有三个非常有用的属性，分别为：`nodeName`、`nodeType`和 `nodeValue`。

`document.getElementById('box').nodeType;` //1，元素节点

节点类型	nodeName	nodeType	nodeValue
元素	元素名称	1	null
属性	属性名称	2	属性值
文本	#text	3	文本内容(不包含 html)

DOM节点的属性

元素节点的这三大属性值：nodeName、nodeType
和 nodeValue。

```
var oDiv = document.getElementById("div");  
console.log(oDiv.nodeName); //DIV  
console.log(oDiv.nodeType); //1  
console.log(oDiv.nodeValue); //null
```

childNodes属性

childNodes: 某一个元素节点的所有子节点，这些子节点包含元素子节点 和文本子节点。

```
var box = document.getElementById( 'box'); //获取一个元素节点
```

```
alert(box.childNodes.length); //获取这个元素节点的所有子节点
```

```
alert(box.childNodes[0]); //获取第一个子节点对象
```

childNodes属性

使用 `childNodes[n]` 返回子节点对象的时候，有可能返回的是元素子节点，比如 `HTMLElement`；也有可能返回的是文本子节点，比如 `Text`。元素子节点可以使用 `nodeName` 或者 `tagName` 获取标签名称，而文本子节点可以使用 `nodeValue` 获取值。

```
for (var i=0; i<oDiv.childNodes.length; i++) {  
    if (oDiv.childNodes[i].nodeType == 1) {  
        //元素节点  
        console.log("元素节点: " + oDiv.childNodes[i].nodeName);  
    }  
    else if (oDiv.childNodes[i].nodeType == 3) {  
        //文本节点  
        console.log("文本节点: " + oDiv.childNodes[i].nodeValue);  
    }  
}
```

innerHTML和nodeValue的区别

1, innerHTML 和 nodeValue 第一个区别，就是取值；
nodeValue可以获取文本节点的内容和属性节点的值。
innerHTML只能获取元素节点中的内容。

2, innerHTML 和 nodeValue 第二个区别: 赋值;
nodeValue 会把包含在文本里的HTML标签按原样输出。
innerHTML 可以解析内部的HTML标签

```
box.childNodes[0].nodeValue='<strong>abc</strong>';
```

```
//结果为: <strong>abc</strong>
```

```
box.innerHTML='<strong>abc</strong>';
```

```
//结果为: abc 加粗
```

innerHTML, innerText, outerHTML

```
<div id="box">  
  <span style="color:red">test1</span> test2  
</div>
```

box.innerHTML:

也就是从对象的起始位置到终止位置的全部内容,包括Html标签。

上例中的box.innerHTML的值也就是:

"test1 test2 "。

box.innerText:

从起始位置到终止位置的内容,但它去除Html标签

上例中的text.innerText的值也就是: "test1 test2", 其中span标签去除了

box.outerHTML:

除了包含innerHTML的全部内容外, 还包含对象标签本身。

上例中的text.outerHTML的值也就是:

```
<div id="box"> <span style="color:red">test1</span> test2</div>
```

➤ outerHTML/innerText, 非 W3C DOM 规范, 而innerText不适用火狐浏览器

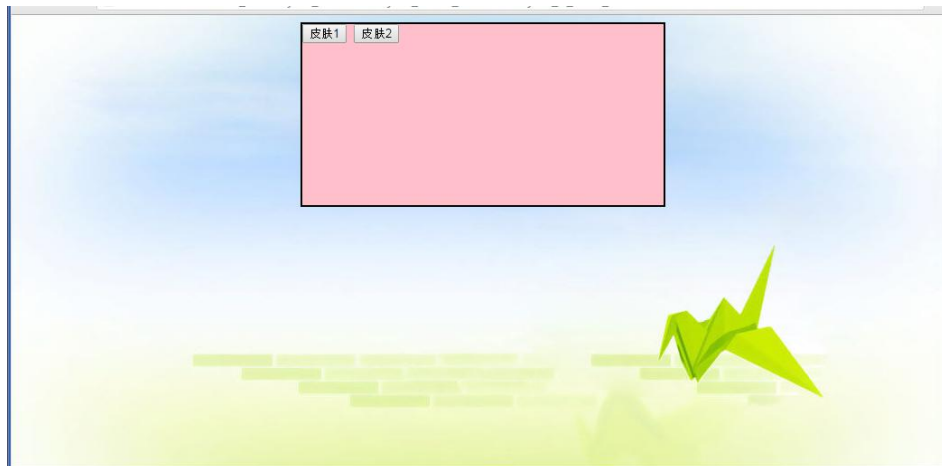
示例

1, 写一个定时器, 每隔0.1秒修改一次div内文字颜色和文字大小.
最开始这个文字是默认大小, 开启定时器后文字大小开始增大,
当增大了6次以后, 文字大小开始缩小, 缩小6次, 文字再开始增大...
效果如下图:



示例

2, 网页换肤, 点击皮肤1, 切换到第一张图, 点击皮肤2切换到第二张图



练习

1, 隔行变色

•	1
•	2
•	3
•	4
•	5
•	6
•	7
•	8
•	9

练习

2, 点击表格中的三张图片, 切换下面的图



作业

1, 简易年历, 如右图, 可以点击对应的月份, 改变下面的文字内容

1 Jan	2 Jan	3 Jan
4 Jan	5 Jan	6 Jan
7 Jan	8 Jan	9 Jan
10 Jan	11 Jan	12 Jan

1月份好

