

## 第十二课 DOM

### 学习目录

- 什么是 DOM
- DOM 操作
- DOM 节点

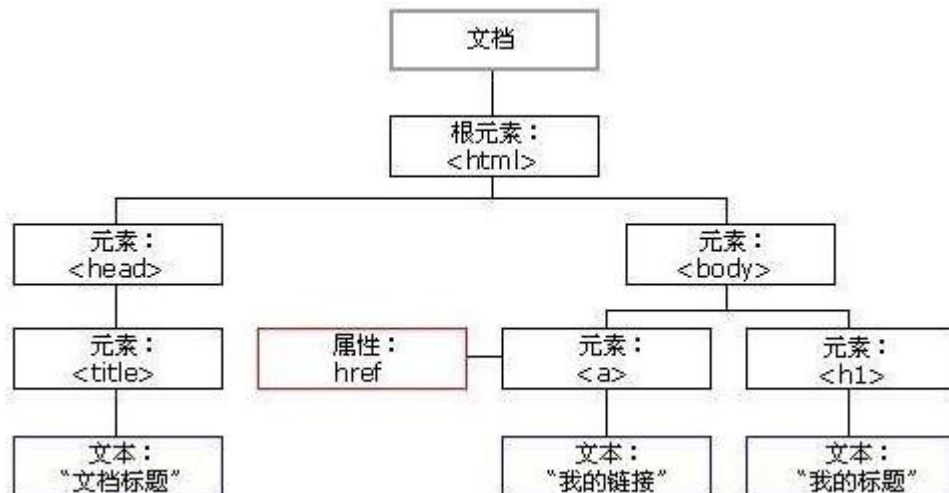
### ①什么是 DOM

DOM 是 Document Object Model( 文档对象模型 )的简写。DOM 文档对象模型 是独立于使用平台和语言之外的接口，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。

DOM 标准分为核心 DOM - 针对任何结构化文档的标准模型，XML DOM - 针对 XML 文档的标准模型，**HTML DOM - 针对 HTML 文档的标准模型**。HTML DOM 定义了所有 HTML 元素的对象和属性，以及访问它们的方法。

#### 1.DOM 结构

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>第十二课 DOM (什么是DOM) </title>
</head>
<body>
  <div>
    <ul>
      <li>文本1</li>
      <li>文本2</li>
      <li>文本3</li>
    </ul>
    <h5>
      <a href="###" title="超链接1">超链接1</a>
      <a href="###">超链接2</a>
      <a href="###">超链接3</a>
    </h5>
  </div>
</body>
</html>
```



HTML 文档中的所有内容都是节点，整个文档是一个文档节点，每个 HTML 元素是元素节点，HTML 元素内的文本是文本节点，每个 HTML 属性是属性节点，包块注释也是注释节点。

DOM 结构方便解析和查询，可以理解成 DOM 树形结构，浏览器可以按照这棵树上的层次顺序解析每个节点，从而显示出网页。

## ②DOM 操作

### 1.查找元素

文档对象表示我们的网页，要使用 HTML 页面中的任何元素，就必须从访问 document 对象开始。

#### 获取元素

document.getElementById(id) 通过元素 id 来查找元素

document.getElementsByTagName(name) 通过元素标签名来查找元素集合

document.getElementsByClassName(name) 通过元素 class 名来查找元素

document.getElementsByName() 通过 name 查找相同名称的元素列表

## 元素属性

`getAttribute(attr)` 获取元素节点指定属性 `attr` 的值

`setAttribute(attr,value)` 设置特定元素节点属性 `attr` 的值为 `value`

`removeAttribute(attr)` 移除特定元素节点属性 `attr`

`element.innerHTML` 获取或修改元素的 inner HTML

`element.style.property` 获取或修改 HTML 元素的样式

## 2.元素关系

`parentNode` 获取当前元素节点的父节点

`childNodes\children` 获取当前元素节点的所有子节点

`firstElementChild \ firstChild` 获取当前元素节点的第一个子节点

`lastElementChild \ lastChild` 获取当前元素节点的最后一个子节点

`previousElementSibling \ previousSibling` 获取当前元素节点的前一个同级节点

`nextElementSibling \ nextSibling` 获取当前元素节点的后一个同级节点

`childElementCount` 获取当前元素节点的子节点数量

`Element Traversal` 规范定义了 `ElementTraversal` 接口，它可以遍历 DOM 树中的元素节点，而不会包含元素节点之外的其他节点，比如标签换行带来的空白文字节点、注释节点等，不需要在借助节点类型来判断节点类型，因此可以方便和快速的操作查找到的节点。

## 3.操作元素

`document.write()` 写入 HTML 输出流。可以把任意字符串写入到文档中

`document.createElement()` 创建 HTML 元素节点

`document.appendChild()` 添加 HTML 元素节点，新元素节点会追加到子元素节点列表的末尾

`document.insertBefore()` 将新节点插入在前面

`document.replaceChild()` 替换 HTML 元素节点

`document.cloneNode()` 复制元素节点

`document.removeChild()` 删除 HTML 元素节点

### ③DOM 节点

#### 1.document 对象

每个浏览器中的 HTML 文档都会成为 Document 对象。Document 对象是 HTML 文档的根节点。操作方法则是该对象方法和对象属性。

#### Document 对象属性

`document.doctype` 返回与文档相关的文档类型声明 (DTD)。

`document.body` 返回文档的 body 元素

`document.domain` 返回当前文档的域名。

`document.title` 返回当前文档的标题。

`document.cookie` 设置或返回与当前文档有关的所有 cookie。

`document.URL;` 获取 URL 路径

#### Document 对象集合

`forms[]` 返回对文档中所有 Form 对象引用。

`images[]` 返回对文档中所有 Image 对象引用。

## 2.DOM 类型

Document 文档类型，整个文档，DOM 树的根节点

Element 元素节点类型

Text 文本节点类型，元素或属性中的文本内容，

Comment 文档中的注释类型

DocumentType 文档声明类型

Attr 属性节点类型

文档是文档节点，页面中 HTML 元素是元素节点，HTML 元素的属性是属性节点，HTML 元素内的文本是文本节点，注释是注释节点。

### 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，文本节点是 #text，文档节点的 nodeName 始终是 #document。

### 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。

### nodeType 属性

nodeType 属性规定节点的类型。

元素 1

属性 2

文本 3

注释 8

文档 9

**ps:元素节点的 attributes 属性可以返回这个元素节点的属性节点的集合。**

### 3.NodeList 与 HTMLCollection

HTMLCollection 是 HTML 元素节点的集合。

NodeList 是所有类型节点的集合。

NodeList 与 HTMLCollection 都是类数组的结构，可以使用索引下标来获取节点。

HTMLCollection 是比较早的 DOM 模型，只包含 HTML 元素，我们用的获取元素的集中方法接口返回的就是 HTMLCollection。

NodeList 是相对比较新的 DOM 模型，除了返回 HTML 元素节点，还会返回 text 文本节点。之后在 js 进阶课程中会跟大家讲解到的其他的获取节点的接口返回的节点全部是 NodeList，比如说 document.querySelectorAll，这里不做过多介绍，避免混淆干扰大家学习。



学习前端，最快的进步是持续！

**谢谢观看！**

我是星星课堂老师：周小周