Web开发（二）

--- 第七章 DOM模型（一）

河北师范大学软件学院
Software College of Hebei Normal University

内容提纲

- DOM 简介
- DOM 树和 DOM 节点
- 访问 DOM 节点

DOM 简介

- DOM（Document Object Model）：文档对象模型
  - 对文档的结构化的表述
  - 定义了在程序中对该结构进行访问的方式
- DOM 分类
  - 核心 DOM：用于任何结构化文档的标准模型
    - 定义了与系统平台和编程语言无关的接口，DOM 是一种 API
  - HTML DOM：用于 HTML 文档的标准模型
  - XML DOM：用于 XML 文档的标准模型

DOM 简介

核心 DOM

```
<html>
<head>
    <title>文档标题</title>
</head>
<body>
    <a href="">我的链接</a>
    <h1>我的标题</h1>
</body>
</html>
```

HTML DOM

DOM 简介

- DOM 作用
  - 访问文档内容，包括元素、属性、文本
  - 增加文档内容，包括元素、属性、文本
  - 删除文档内容，包括元素、属性、文本
  - 修改文档内容，包括元素、属性、文本
- DOM 的应用十分广泛，各种网页特效均有 DOM 的踪影

内容提纲

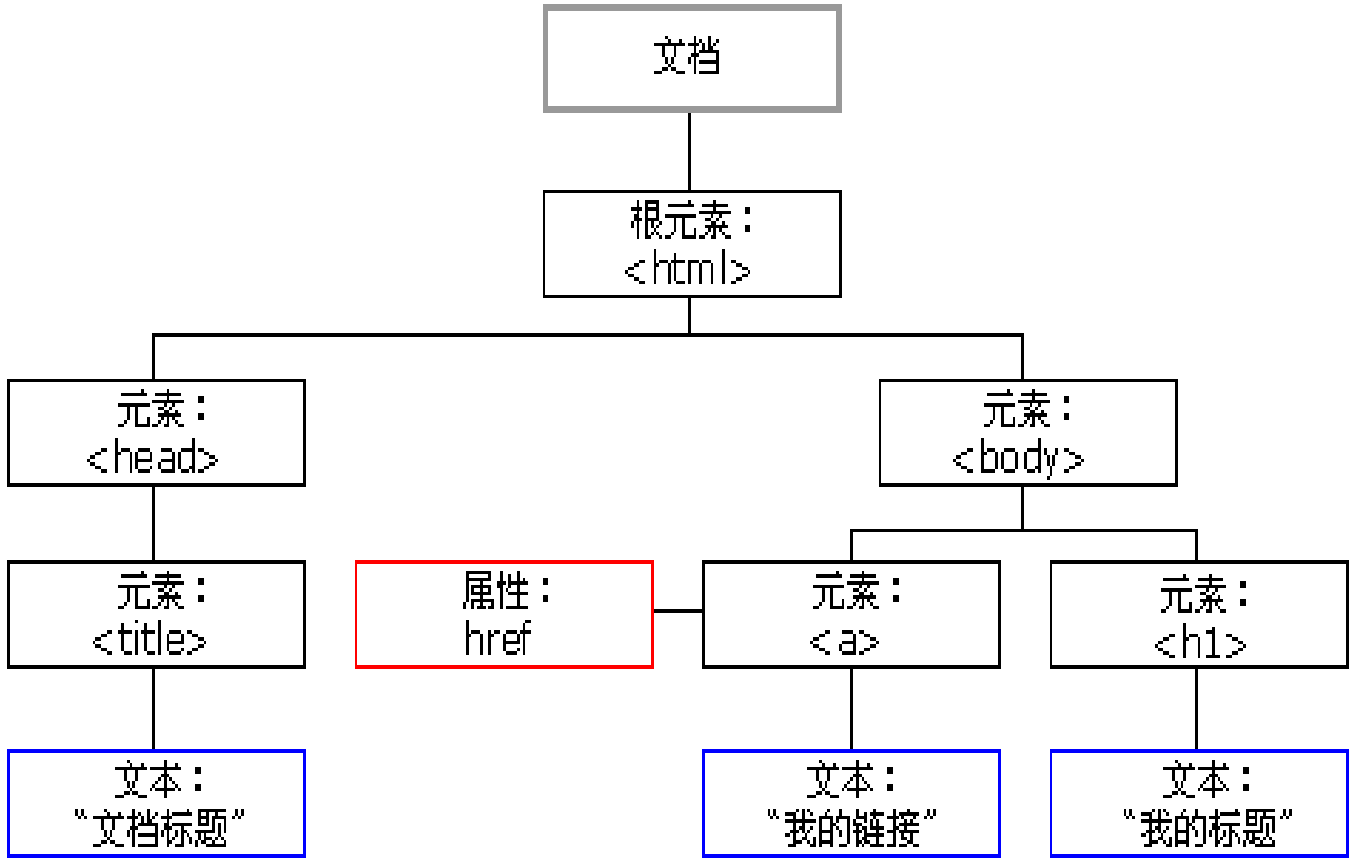
- DOM 简介
- DOM 树和 DOM 节点
- 访问 DOM 节点

DOM 树

- DOM 将 HTML 文档抽象为树形结构，称这棵树为 DOM 树
- HTML 中的每一项内容都可以在 DOM 树中找到
- 对文档内容的操作即对 DOM 树的操作

```
<html>
<head>
    <title>文档标题</title>
</head>
<body>
    <a href="1.html">我的链接</a>
    <h1>我的标题</h1>
</body>
</html>
```

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；

DOM 节点

- DOM 节点是一个对象，拥有属性和方法
  - 元素节点
    - HTML 开始标签中的属性为元素节点对象的属性
    - HTML 开始标签中的事件属性为元素节点对象的方法
    - 元素节点对象中的方法函数中的 this 指向当前触发事件的元素
  - 属性节点
  - 文本节点
    - 空格、换行空格属于文本节点

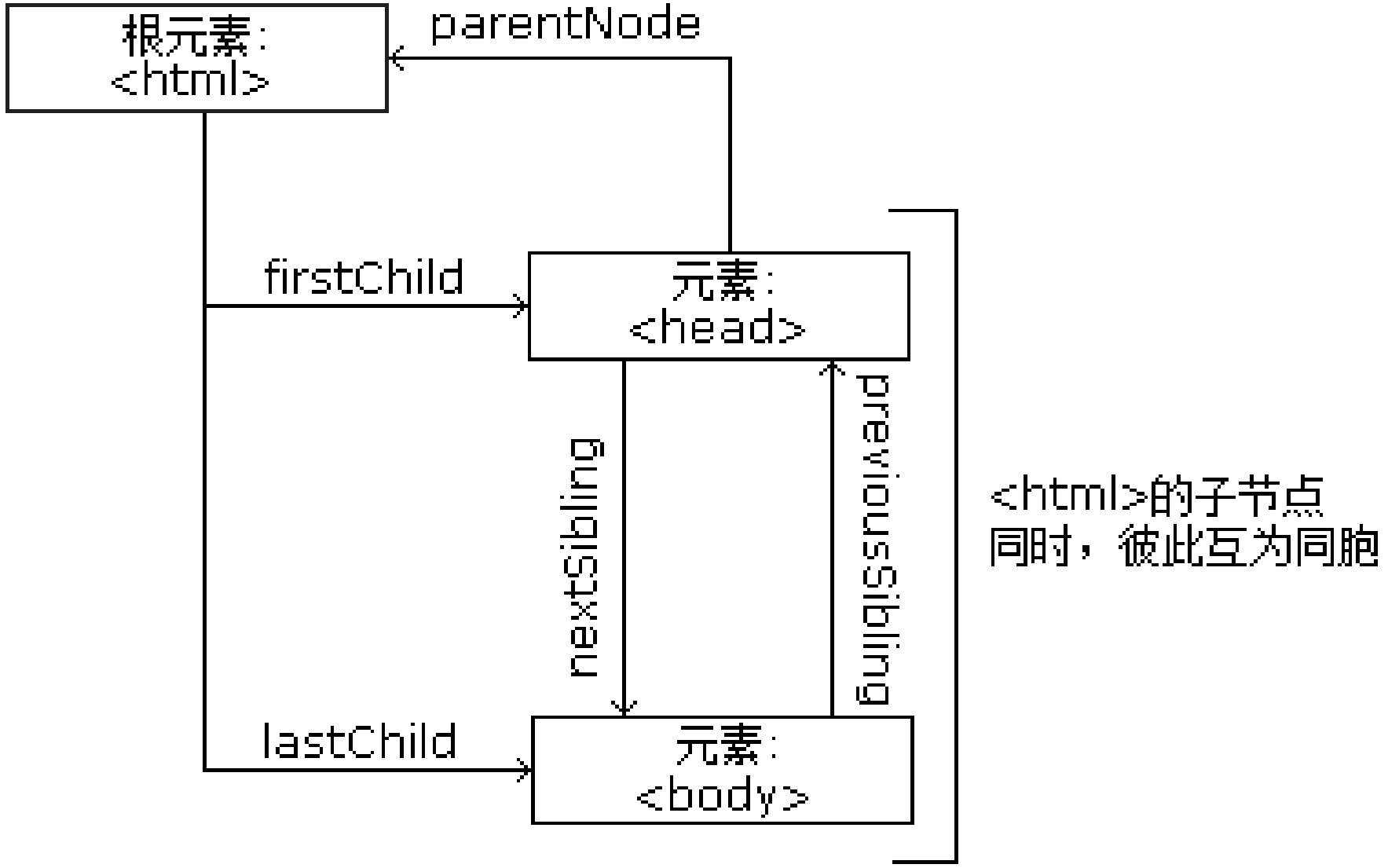
```
<h1>我是标题一</h1>
<h2>我是标题二</h2>
```

换行空格也属于文本节点

元素节点关系

- 在节点树中，顶端节点被称为根（root）
- 父节点
  - 每个节点都有父节点、除非该元素是文档的根节点
- 子节点
  - 每个元素节点可以有 0 个、1 个或多个子节点
- 同胞节点
  - 指拥有相同父节点的节点

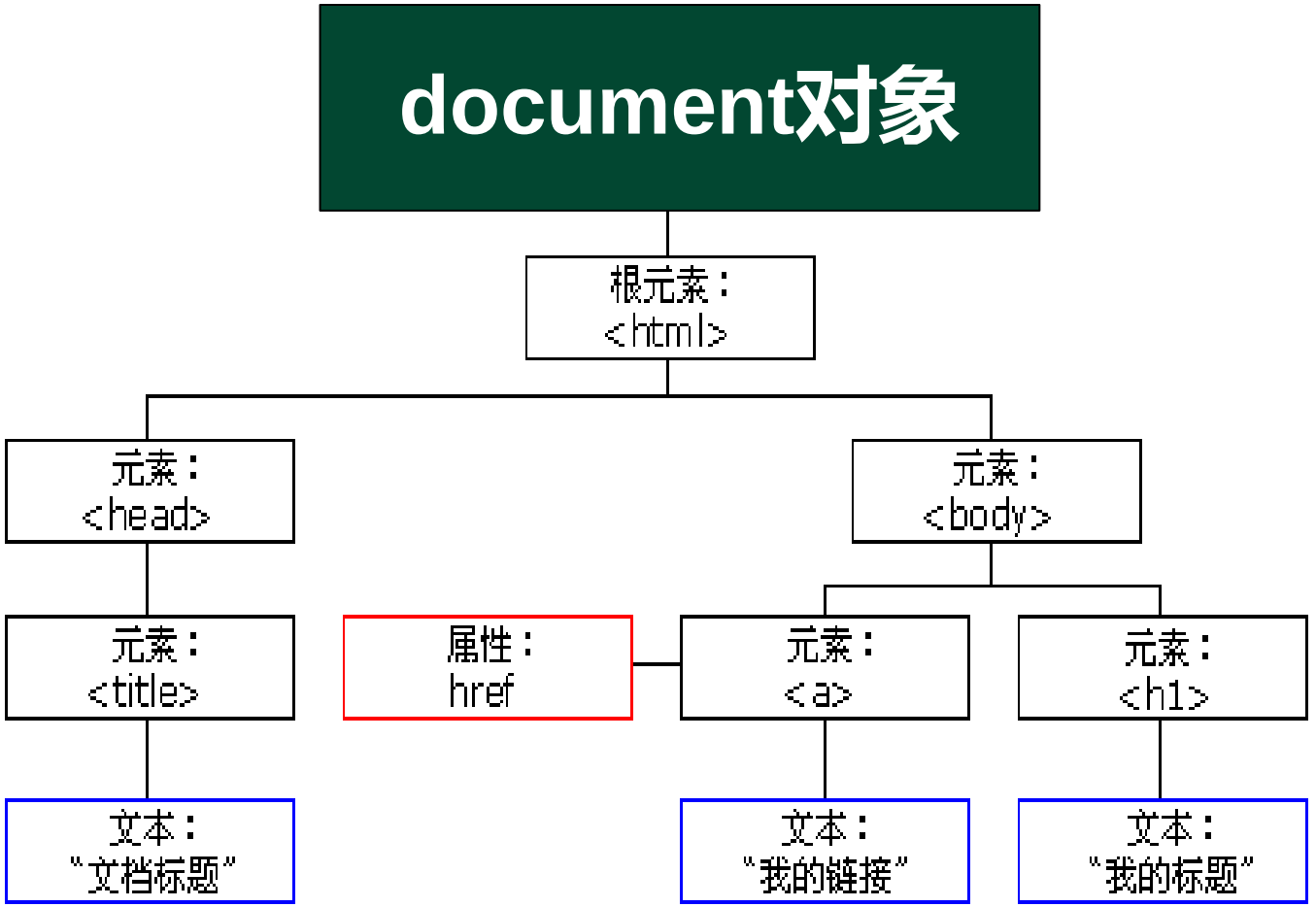
元素节点关系


根元素:
<html>
parentNode
firstChild
元素:
<head>
nextsibling
previousSibling
lastChild
元素:
<body>
<html>的子节点
同时，彼此互为同胞

内容提纲

- DOM 简介
- DOM 树和 DOM 节点
- 访问 DOM 节点

访问 DOM 节点



- JavaScript 解释器会为载入的每个 HTML 文档创建一个对应的 document 对象
- 通过使用 document 对象，可以从脚本中对 HTML 页面中的所有节点进行访问

demo 7-1.html

访问元素节点

元素节点

- 直接获取节点
  - 通过 id 属性获得节点
  - 通过标签名获得所有同名标签
  - 通过类名获得所有类名相同的标签
- 通过节点关系获取
  - 通过父节点获得子节点
  - 通过子节点获得父节点
  - 获得前后兄弟节点

直接获取元素节点

- 通过 id 属性获得节点
  - document.getElementById( ) → ID 选择器
- 通过标签名获得**所有**同名标签
  - document.getElement**s**ByTagName( ) → 标签选择器
- 通过类名获得**所有**类名相同的标签
  - document.getElement**s**ByClassName( ) → 类选择器

小驼峰命名法

demo 7-2.html

直接获取元素节点

```
var p = document.getElementsByTagName("p");
for (i = 0; i < p.length; i++) {
    document.write(p[i].innerHTML);
    document.write("<br/>");
}
```

长度属性

类似 数组

访问某一个具体值

demo 7-3.html

特殊元素节点

- document.d 文档中
  <html>元素
- document.h 元素
- document.b 元素

```
<html>
<head>
    <meta charset="UTF-8">
    <title>文档标题</title>
</head>
<body>
    <h1>我的标题</h1>
</body>
</html>
```


demo 7-4.html

元素树

- 父、子、兄弟关系

```
<li>  ①
      <a href="#"> — </a>  ② ③
</li>
```

li 节点的子节点有哪些?

① 文本节点
② 元素节点
③ 文本节点

- JavaScript 引入了元素树的概念
- 元素树：仅仅包含元素节点的树结构，不是一颗新树，只是节点树的子集

```
<li>
      <a href="">—</a>  ①
</li>
```

节点树&元素树

	节点树	元素树（没有文本、注释）
获取父节点	parentNode	parentElement
获取子节点	childNodes	children
获取第一个子节点	firstChild	firstElementChild
获取最后一个子节点	lastChild	lastElementChild
获取前一个兄弟节点	previousSibling	previousElementSibling
获取后一个兄弟节点	nextSibling	nextElementSibling

①通过父节点获得子节点

- childNodes 属性——返回节点的子节点集合。之后可以通过循环或者索引找到需要的元素节点。

  例：document.getElementById("myList").childNodes;

  document.getElementById("myList").childNodes[0];

- children 属性——返回节点的所有元素子节点集合。

  例：document.getElementById("myList").children;

  document.getElementById("myList").children[0];

demo 7-5.html

② 通过父节点获得首个子节点

- firstChild 属性——返回指定节点的首个子节点。可递归使用，即支持 parentObj.firstChild.firstChild... 的形式，如此可获得更深层次的节点。

  例：document.getElementById("myList").firstChild;

- firstElementChild 属性——返回指定节点的首个元素子节点。

  例：document.getElementById("myList").firstElementChild;

demo 7-6.html

③ 通过父节点获得最后一个子节点

- lastChild 属性——返回指定节点的最后一个子节点。可递归使用，即支持 parentObj.lastChild.lastChild... 的形式，如此可获得更深层次的节点。

  例：document.getElementById("myList").lastChild;

- lastElementChild 属性——返回指定节点的最后一个元素子节点

  例：document.getElementById("myList").lastElementChild;

demo 7-7.html

④ 通过子节点获得父节点

- parentNode 属性——返回指定节点的父节点
- parentElement 属性——返回指定节点的父节点

html代码:

```
<h1>新闻动态<span id="time">2018-10-7</span></h1>
```

js代码:

```
var parent= document.getElementById("time").parentNode;
console.log(parent.innerHTML);
```

输出结果:

```
新闻动态<span id="time">2018-10-7</span>
```


demo 7-8.html

⑤ 获得前后兄弟节点

- previous(Element)Sibling 属性——返回指定节点紧跟的前一个（元素）兄弟节点。

  例：document.getElementById("item1").previousSibling;

- next(Element)Sibling 属性——返回指定节点之后紧跟的一个（元素）兄弟节点

  例：document.getElementById("item2").nextSibling;

demo 7-9.html

元素节点内容

- innerHTML 属性
  - innerHTML 是 DOM 中元素节点的属性，相当于一个容器
  - 用于获取或改变任意元素节点的内容，该属性可读可写
  - 操作简单，几乎所有浏览器均支持
- innerText 属性

  https://developer.mozilla.org/zh-CN/docs/Web/API/Node/innerText
- textContent 属性

  https://developer.mozilla.org/zh-CN/docs/Web/API/Node/textContent

元素节点内容

- 读取元素节点内容

```
var txt = document.getElementById("intro").innerHTML;
document.write(txt);
```

- 修改元素节点内容

```
document.getElementById("intro").innerHTML = "hello";
```


demo 7-10.html

元素节点属性

- 获取某一元素节点的属性
  - node.getAttribute(attrName)
- 设置某一元素节点的属性
  - node.setAttribute(attrName，value)
  - 当属性存在时，为修改相应属性值
  - 当属性不存在时，为添加相应属性

demo 7-11.html

元素节点属性

- 删除某一元素节点的属性
  - node.removeAttribute(attrName)
- 判断某一元素节点是否含有某属性
  - node.hasAttribute(attrName)
  - 返回值为布尔值 true/false

demo 7-12.html

综合练习

删除本行0						
删除本行1						
删除本行2						
删除本行3						
删除本行4						
删除本行5						
删除本行6						
删除本行7						

demo 7-12.html

DOM 操作小结

- DOM 操作注意事项：
  - 获取 DOM 节点的操作要在被浏览器加载之后进行
  - <script>标签放在 HTML 内容后面，即<body>结束标签前面
- 通用性强，几乎所有浏览器均支持
- 操作比较复杂，书写代码量过大

DOM操作小结

- DOM 节点、DOM 树与元素树
- 访问 DOM 元素节点的方式
  - 直接访问
  - 间接访问
  - 特殊节点访问
- 元素节点的属性操作
  - 增、删、改、查、判断方法
- 元素节点的文本内容操作 —— innerHTML

Thank You!