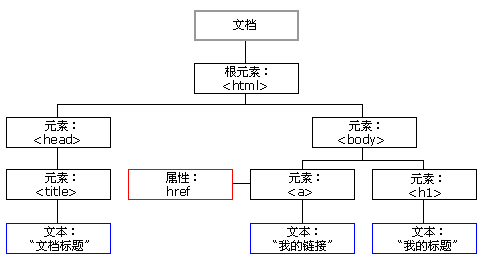
**JavaScript DOM编程**

**讲师: 张晓飞**



# DOM概述

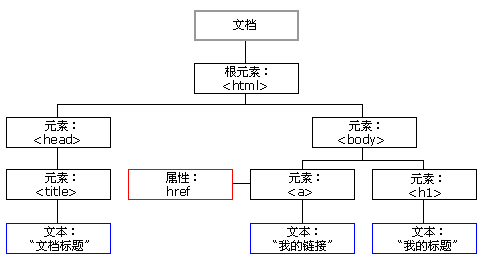
## **什么是DOM?**

* DOM= Document Object Model，[文档对象模型](http://baike.baidu.com/view/758570.htm)， DOM 是 W3C（万维网联盟）的标准。DOM 定义了访问 HTML 和 XML 文档的标准：独立于平台和语言的接口. W3C DOM 标准被分为 3 个不同的部分：
  + 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

## **什么是HTML DOM?**

* **HTML DOM 定义了访问和操作HTML文档的标准方法。**
* **HTML DOM 把 HTML 文档呈现为带有元素、属性和文本的树结构（节点树）**
* **当我们用浏览打开一个html文件时, 浏览器内置的HTML解析引擎就会将整个文档加载到内存中, 并根据HTML DOM定义的接口和实现类生成一个包含文件内所有数据的DOM对象树.**

|  |
| --- |
| <html>  <head>  <title>文档标题</title>  </head>  <body>  <h1>我的标题</h1>  <a href=*"http://www.atguigu.com"*>我的链接</a>  </body>  </html> |



# HTML DOM核心: 节点

## HTML节点(Node)

* HTML文档中的每个成分都是一个节点。
  + 整个文档是一个**文档节点**(Document)
  + 每个 HTML 标签是一个**元素节点**(Element)
  + 每一个 HTML 属性是一个**属性节点**(Attribute)
  + 包含在 HTML 元素中的文本是**文本节点**(Text

## Node层次

* DOM中的节点彼此都有等级关系。
* 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。

## HTML DOM节点树

* **一棵节点树中的所有节点彼此都是有关系的**

|  |
| --- |
| <html>  <head>  <title>DOM Tutorial</title>  </head>  <body>  <h1>DOM Lesson one</h1>  <p>Hello world!</p>  </body>  </html> |

|  |  |
| --- | --- |
| 父节点 | 除文档节点之外的每个节点都有父节点 |
| 子节点 | 大部分元素节点都有子节点 |
| 同辈节点 | 当节点分享同一个父节点时，它们就是同辈 |
| 后代节点 | 后代指某个节点的所有子节点 |
| 先辈节点 | 先辈是某个节点的父节点，或者父节点的父节点，以此类推 |

# DOM操作

## HTML DOM访问节点(查询)

|  |  |  |
| --- | --- | --- |
| **对象类型** | **属性/方法** | **说明** |
| 文档/元素节点 | **getElementById(id)** | 根据标签的id得到对应的标签对象 |
| 文档/元素节点 | **getElementsByTagName(tag)** | 根据标签名得到对应的所有子标签对象的集合(数组) |
|  |  |  |
| 节点 | **nodeName** | 得到节点名 |
| 节点 | **nodeValue** | 得到节点的值 |
| 节点 | **nodeType** | 节点类型值 |
|  |  |  |
| 元素节点 | **childNodes** | 得到所有子节点的集合(数组) |
| 元素/文本节点 | **parentNode** | 得到父节点对象(标签) |
| 元素节点 | **firstChild** | 得到第一个子节点(标签/文本) |
| 元素节点 | **lastChild** | 得到最后一个子节点(标签/文本) |
|  |  |  |
| 元素节点 | **getAttributeNode(attrName)** | 根据属性名标签的属性节点 |

## HTML DOM的增删改

|  |  |  |
| --- | --- | --- |
| **对象类型** | **属性/方法** | **说明** |
| 文档节点 | **createElement(tagName)** | 创建一个新的元素节点对象 |
| 文档节点 | **createTextNode(text)** | 创建一个文本节点对象 |
|  |  |  |
| 元素节点 | **appendChild(node)** | 将指定的节点添加为子节点 |
| 元素节点 | **insertBefore(new,target)** | 在指定子节点的前面插入新节点 |
|  |  |  |
| 元素节点 | **replaceChild(new, old)** | 用新节点替换原有的旧子节点 |
| 元素节点 | **removeChild(childNode)** | 删除指定的子节点 |
| 元素节点 | **setAttribute(name, value)** | 为标签添加一个属性 |
| 元素节点 | **removeAttribute(name)** | 删除指定的属性 |
|  |  |  |
| 元素节点 | **innerHTML** | 向标签中添加一个标签 |

|  |
| --- |
| <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>HTML DOM编程测试</title>  <style type=*"text/css"*>  **ul** {  list-style-type: *none*;  }  **li** {  border-style: *solid*;  border-width: *1px*;  padding: *5px*;  margin: *5px*;  background-color: *#99ff99*;  float: *left*;  }  *.out* {  width: *400px*;  border-style: *solid*;  border-width: *1px*;  margin: *10px*;  padding: *10px*;  float: *left*;  }  </style>  </head>  <body>  <div class=*"out"*>  <p>你喜欢哪个城市?</p>  <ul id=*"city"*>  <li id=*"bj"* name=*"BeiJing"*>北京</li>  <li>上海</li>  <li id=*"dj"*>东京</li>  <li>首尔</li>  </ul>  <br>  <div id=*"inner"*></div>  </div>  </body>  </html> |

# 事件监听

## 添加事件监听

* 页面的很多交互操作都是通过在一些页面视图上添加事件监听的方式来实现的
* 给标签添加事件属性

|  |
| --- |
| <button onclick="showMsg()">Click Me</button> |

* 给标签对象添加事件属性

|  |
| --- |
| **var** btnEle = document.getElementById("btn");  btnEle.onclick = **function**(){  alert("Thank you, baby!!!!!");  }; |

## 事件回调函数之this

* 在事件函数的回调函数中, this代表发生事件所有的视图对象, 经常利用this来获取数据

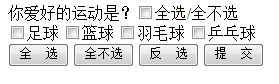
# 练习

## 练习1: 爱好选择

### 技术点:

* + DOM节点的查找
  + 添加事件监听
  + checkbox的操作
  + 逻辑能力
  + 重构能力

### 效果图



### 页面

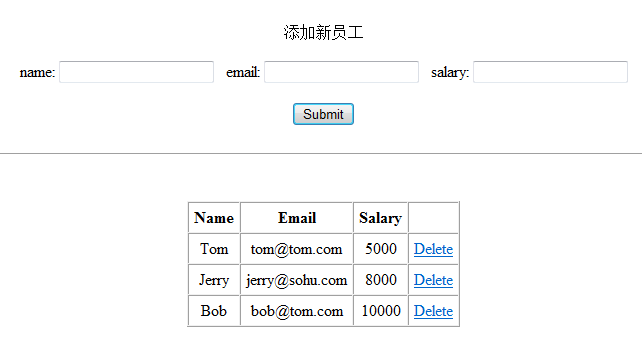
|  |
| --- |
| <body>  <form method=*"post"* action=*""*>  你爱好的运动是？  <input type=*"checkbox"* id=*"checkedAll\_2"*/>全选/全不选  <br/>  <input type=*"checkbox"* name=*"items"* value=*"足球"*/>足球  <input type=*"checkbox"* name=*"items"* value=*"篮球"*/>篮球  <input type=*"checkbox"* name=*"items"* value=*"羽毛球"*/>羽毛球  <input type=*"checkbox"* name=*"items"* value=*"乒乓球"*/>乒乓球  <br/>  <input type=*"button"* id=*"CheckedAll"* value=*"全　选"*/>  <input type=*"button"* id=*"CheckedNo"* value=*"全不选"*/>  <input type=*"button"* id=*"CheckedRev"* value=*"反　选"*/>    <input type=*"submit"* id=*"send"* value=*"提　交"*/>  </form>  </body> |

## 练习2: 员工管理

### 技术点:

* + DOM节点的CRUD操作
  + 添加事件监听
  + 逻辑能力
  + 重构能力

### 效果图



### 页面

|  |
| --- |
| <center>  <br> <br> 添加新员工 <br> <br> name: <input type=*"text"*  name=*"name"* id=*"name"* />&nbsp;&nbsp; email: <input type=*"text"*  name=*"email"* id=*"email"* />&nbsp;&nbsp; salary: <input type=*"text"*  name=*"salary"* id=*"salary"* /> <br> <br>  <button id=*"addEmpButton"*>Submit</button>  <br> <br>  <hr>  <br> <br>  <table border=*"1"* cellpadding=*"5"* cellspacing=*0*>  <tbody id=*"employeetable"*>  <tr>  <th>Name</th>  <th>Email</th>  <th>Salary</th>  <th>&nbsp;</th>  </tr>  <tr>  <td>Tom</td>  <td>tom@tom.com</td>  <td>5000</td>  <td><a href=*"deleteEmp?id=001"*>Delete</a></td>  </tr>  <tr>  <td>Jerry</td>  <td>jerry@sohu.com</td>  <td>8000</td>  <td><a href=*"deleteEmp?id=002"*>Delete</a></td>  </tr>  <tr>  <td>Bob</td>  <td>bob@tom.com</td>  <td>10000</td>  <td><a href=*"deleteEmp?id=003"*>Delete</a></td>  </tr>  </tbody>  </table>  </center> |