**正课：**

**1. 什么是jQuery**

**2. 如何使用jQuery**

**3. 查询**

**4. 修改**

**1. 什么是jQuery:**

**什么是: 第三方开发的 执行DOM操作的 极简化的函数库**

**第三方: 下载**

**执行DOM操作: 学jQuery还是在学DOM,增删改查,事件**

**极简化: 是DOM操作的终极简化**

**函数库: jQuery一切都用函数解决**

**为什么: 2个原因:**

**1. 终极简化**

**2. 屏蔽了大量兼容性问题: 只要jQuery让用的，都没有兼容性问题**

**何时: 今后几乎所有大型项目和框架的开发，都是用jQuery完成的。**

**2. 如何:**

**鄙视: 如何学习一项新技术**

**1. 找官网: 是什么**

**2. 看DEMO: 快速入门**

**3. 下手册: 备查**

**下载: 2种版本:**

**1.x 支持旧浏览器**

**2.x 3.x 不再支持旧浏览器**

**压缩和未压缩版:**

**jquery-xxx.js 未压缩版**

**包括完整的注释，代码格式，见名知意的变量名**

**体积大，可读性好——学习或开发时**

**jquery-xxx.min.js 压缩版——专门工具生成**

**去掉所有注释，空字符，极简化变量名**

**体积小，可读性差——生产环境中**

**部署: 2种:**

**1. 将js部署在服务器本地:**

**2. 利用CDN网络:**

**什么是: 网络中的多态服务器，智能判断客户端到某台服务器的网络状况，选择最优服务器下载资源**

**为什么: 避免距离带来的网络延迟**

**何时: 在项目中用到了公共的库或框架**

**如何: 官网->下载->CDN地址**

**引入网页:**

**先引入jquery.js，再引入自定义脚本**

**<script src="本地相对路径/CDN网址">**

**\*\*\*jQuery类型:**

**引入jquery.js时: 其实是向全局作用域中，添加了一个新的类型——jQuery:**

**构造函数: 负责创建jQuery类型的对象**

**必须先创建jQuery类型的对象，才能用jQuery**

**原型对象: 保存jQuery对象可用的所有简化版API**

**只有jQuery类型的子对象，才能用简化版API**

**创建jQuery对象:**

**什么是jQuery对象: 封装DOM元素，并提供操作DOM元素的简化版API的类数组对象**

**为什么: DOM元素不是jQuery类型子对象，无法直接使用jQuery的简化版API**

**何时: 只要使用jQuery，必须先创建jQuery对象，封装DOM对象。才能使用jQuery简化版API**

**如何:**

**1. 通过选择器，从无到有的查找DOM元素，创建jQuery对象**

**jQuery("selector") 可进一步简写为: $("selector")**

**先查找符合"selector"的DOM元素**

**将DOM元素装进新建的jQuery对象中**

**2. 将到手的DOM元素封装进jQuery对象:**

**$(DOM元素) 创建一个jQuery对象，保存DOM元素**

**jQuery API的三大特性:**

**1. 一个函数两用:**

**没给新值，默认执行获取操作**

**给了新值，变为执行修改操作**

**比如: .html() 获取内容**

**.html(xxx) 修改内容为xxx**

**2. 自带遍历:**

**对jQuery对象调用一次简化版API，等效于对类数组对象中每个元素都调用一次API.**

**3. 几乎每个jQueryAPI都返回正在使用的jQuery对象本身**

**链式操作**

**3. 查找:**

**1. 按选择器:**

**$(...)支持所有css选择器！**

**css中没有的，jquery添加的:**

**基本过滤:**

**先将所有元素收集到一个结果集合中统一编号**

**再选择集合中指定位置的元素**

**和元素在其父元素下的位置无关**

**见first vs first-child.png**

**内容过滤: 3种:**

**1. 按包含的文本内容过滤:**

**:contains(text) 查找内容中包含指定text文本的元素**

**2. 按子元素特征过滤:**

**:has(selector) 查找内容中包含符合selector要求的子元素的父元素**

**3. 按是否包含内容来过滤:**

**:empty 空元素**

**:parent 非空的元素**

**可见性过滤:**

**:hidden: 只能选择两种情况的隐藏:**

**display:none input type=hidden**

**[:visible](mk:@MSITStore:C:\\xampp\\htdocs\\11_JQUERY\\jQuery1.11.3_20150618.chm::/visible.html): 就是:hidden的反义**

**状态过滤: 三大状态: :disabled :checked :selected**

**表单元素过滤:**

**:input 选择所有四大类表单元素:**

**input button textarea selected**

**:[type] 每种input，都对应一种专门的选择器**

**比如: :text :password :checkbox :radio ...**

**css选择器 vs jquery选择器:**

**优先选css，因为效率高**

**jquery选择器都是用js程序模拟出来**

**2. 按节点间关系:**

**父元素: .parent() =>.parentNode**

**直接子元素: .children() => .children**

**查找父元素下符合条件的子元素:**

**.children(["selector"]) 只能找直接子元素**

**.find("selector") 在所有后代中查找**

**第一个子元素: .children().first() => .firstElementChild**

**最后一个子元素: .children().last() => .lastElementChild**

**前一个兄弟: .prev() => .previousElementSibling**

**.prevAll()**

**后一个兄弟: .next() => .nextElementSibling**

**.nextAll()**

**除我之外，所有兄弟: .siblings();**

**4. 修改: 一个函数两用！**

**内容: .html() => .innerHTML**

**.text() => .textContent**

**.val() => .value**

**属性:**

**标准属性: .attr("属性名"[,值])**

**状态属性: .prop("状态属性"[,bool])**

**自定义扩展属性: .data("属性名"[,值])**

**样式:**

**修改内联样式:**

**1. 仅修改一个css属性:**

**.css("css属性名",值) => .style.css属性=值**

**2. 批量修改多个css属性:**

**.css({**

**css属性名:值,**

**... : ...**

**})**

**获取值:**

**.css("css属性名") => getComputedStyle()获得计算后的样式**