**正课:**

**1. 什么是jQuery**

**2. 如何使用jQuery**

**3. 查找**

**1. 什么是jQuery**

**jQuery是第三方提供的执行DOM操作的极简化的函数库**

**第三方: 先下载，才能用**

**执行DOM操作: 学jQuery还是在学DOM的增删改查，事件绑定**

**极简化: jQuery是对DOM API的终极简化**

**函数库: jQuery是用函数解决一切问题！**

**为什么: 2个主要原因:**

**1. API极简化: 4个方面: 增删改查，事件绑定，动画，ajax**

**2. 解决了大部分兼容性问题: 凡是jQuery让用的都没有兼容性问题**

**何时: 几乎所有项目或框架都用jQuery开发。**

**2. 如何使用jQuery:**

**下载:**

**版本: 1.x 兼容旧浏览器**

**压缩版: 去掉空字符，注释，极简化变量名**

**体积小，便于下载**

**可读性查**

**何时: 生产环境中才用**

**未压缩版: 包含完备的空字符，注释，以及见名知意的变量名**

**可读性好**

**体积大，下载慢**

**何时: 学习和开发时使用**

**2.x 不再兼容旧浏览器**

**3.x 不兼容旧浏览器**

**鄙视: jQuery 3.x的新特性:**

**1. 所有代码运行在严格模式下**

**2. 支持for...of循环遍历（以前用.each()函数）**

**3. 新的动画API: requestAnimationFrame()**

**4. 支持Promise**

**引入: 2种方式:**

**1. 引入项目本地文件夹中的jquery.js文件**

**2. 引入CDN网络中共享的jquery.js文件**

**原理:**

**引入jquery.js其实是在window中添加一个新类型jQuery**

**包括: 1. 构造函数: 创建该类型的子对象**

**2. 原型对象: 保存该类型的子对象共用的API**

**要使用jQuery简化版API，必须先创建jQuery类型的子对象。**

**因为jQuery是执行DOM操作的API，所以jQuery对象都要封装一个或多个DOM元素。**

**如何创建jQuery对象: 2种:**

**1. 用选择器查找DOM元素，并封装进jQuery对象中:**

**var $jq=new jQuery("选择器")**

**用选择器查询DOM元素，并封装进新创建的jQuery对象中**

**因为: function jQuery(){**

**return new xxxx()**

**}**

**所以: 创建jQuery对象时，不用写new**

**var $jq= jQuery("选择器")**

**因为: window.$=jQuery**

**所以: 可进一步简化为: var $jq=$("选择器")**

**2. 直接将获得的DOM元素封装为jQuery对象**

**var $jq=$(DOM元素)**

**什么是jQuery对象: jQuery对象是封装一组DOM元素对象，并提供操作DOM对象的简化版API的 类数组对象**

**jQuery API三大特性:**

**1. 一个函数两用: 没给新值，默认读取**

**给了新值，变为修改**

**2. 自带遍历效果: 对jQuery对象调用一次API，等效于自动对内部每个DOM元素都调用一次API**

**3. 几乎每个API都返回正在使用的jQuery对象:**

**可使用链式操作，精简代码**

**3. 查找: 按选择器查找:**

**如何: var $jq=$("选择器")**

**包括: jQuery支持所有CSS选择器，并扩展了部分新选择器**

**1. 基本选择器: 同CSS**

**5个: #id, element, .class, \*, select1,select2,...**

**2. 层级选择器: 同CSS**

**4个: 父 后代**

**父>子代**

**兄+弟**

**兄~弟**

**3. 子元素过滤选择器: 同CSS**

**4个: :first-child :last-child :nth-child(n) :only-child**

**特点: 1. 在每个父元素内分别编号**

**2. 编号从1开始**

**4. 基本过滤选择器: jQuery新增:**

**:first :last :gt(i) :lt(i) :eq(i) :even :odd**

**特点: 1. 先取出所有符合条件的元素放入一个集合，在集合中同一编号**

**2. 编号从0开始**

**5. 属性过滤选择器: 同CSS**

**[属性名] [属性名=值] [属性名^=值] [属性名$=值]**

**[属性名\*=值] [属性名!=值]**

**"而且"：多个选择器不加分割的连续使用**

**6. 内容过滤选择器: jQuery中新增**

**用元素的内容作为条件进行查找或判断**

**1. :contains(文本): 内容中包含指定"文本"的元素**

**2. :has(selector): 包含符合selector要求的子元素 的父元素**

**3. :parent : 选择不为空的元素**

**4. :empty : 选择空元素**

**7. 可见性过滤: jQuery中新增**

**:visible 选择可见的元素**

**:hidden 选择隐藏的元素，只识别display:none和input type="hidden"的元素**

**8. 表单元素过滤和状态过滤:**

**表单元素过滤: 2大类:**

**:input 选择所有表单元素(input button select textarea)**

**每类input都有一个对应的选择器:**

**:text :password :submit :radio :checkbox ...**

**状态过滤: :enabled :disabled :selected :checked**

**按节点间关系查找:**

**1. 父子关系:**

**$elem.parent() //.parentNode**

**$elem.children() //.children**

**$elem.children().last() //.lastElementChild**

**2. 兄弟关系:**

**$elem.next() //.nextElementSibling**

**$elem.nextAll() //之后所有兄弟**

**修改:**

**内容: $elem.html() //.innerHTML**

**属性: $elem.prop() //.属性名**

**样式: $elem.css() //.style.css属性**

**$elem.addClass() //.className**

**.removeClass()**

**事件绑定: $elem.on("事件名",function(){**

**var $elem=$(this)**

**})**

**判断元素特征: var bool=$elem.is("选择器")**