

# 《Web 前端技术 - 就业技能》

章节	Ch01 - 初识 HTML
题目 1)	<b>什么是 HTML?</b>
	HTML 是用来描述网页的一种语言。 HTML 指的是超文本标记语言 (Hyper Text Markup Language) HTML 不是一种编程语言, 而是一种标记语言 (markup language) 标记语言是一套标记标签 (markup tag) HTML 使用标记标签来描述网页
题目 2)	<b>XHTML 和 HTML 有什么区别</b>
	HTML 是一种基本的 WEB 网页设计语言, XHTML 是一个基于 XML 的置标语言最主要的不同 XHTML 元素必须被正确地嵌套。 XHTML 元素必须被关闭标签名必须用小写字母 XHTML 文档必须拥有根元素
题目 3)	<b>什么是 W3C?W3C 标准包括哪些内容?</b>
	(1)World Wide Web Consortium (万维网联盟) 成立于 1994 年, Web 技术领域最权威和具影响力的国际中立性技术标准机构 (2)包括:结构化标准语言 (XHTML 、 XML) 表现标准语言 (CSS) 行为标准 (DOM、ECMAScript )
题目 4)	<b>HTML5 有哪些新特性, 移除了哪些元素? 如何处理 HTML5 新标签的浏览器兼容问题? 如何区分 HTML 和 HTML5?</b>
	HTML5 现在已经不是 SGML 的子集, 主要是关于图像, 位置, 存储, 多任务等功能的增加。 绘画 canvas 用于媒介回放的 video 和 audio 元素 本地离线存储 localStorage 长期存储数据, 浏览器关闭后数据不丢失; sessionStorage 的数据在浏览器关闭后自动删除 语义化更好的内容元素, 比如 article,footer,header,nav,section 表单控件: calendar,date,time,email,url,search 新的技术 webworker,websocketGeolocation 移除的元素 纯表现的元素: basefont,big,center,font,s,strike,tt,u; 对可用性产生负面影响的元素: frame,frameset,noframes; 支持 HTML5 新标签: IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签, 可以利用这一特性让这些浏览器支持 HTML5 新标签, 浏览器支持新标签后, 还需要添加标签默认的风格。
题目 5)	<b>HTML5 为什么只需要写! DOCTYPE HTML?</b>
	HTML5 不基于 SGML,因此不需要对 DTD 进行引用, 但是需要 doctype 来规范浏览器的行为 (让浏览器按照它们应该的方式来运行); 而 HTML4.01 基于 SGMA, 所以需要对 DTD 进行引用, 才能告知浏览器文档所使用的文档类型。

<b>章节</b>	<b>Ch02 - 列表表格及表单美化</b>
<b>题目 1)</b>	<b>表格跨行和跨列及文本对齐的属性是什么?</b>
	跨行: rowspan="行数" 跨列: colspan="列数" 对齐: align="center/left/right";
<b>题目 2)</b>	<b>form 表单提交属性? 设置请求方式属性? 请求方式的区别?</b>
	表单提交属性: method 请求方式分为两种: post 和 get post 比较安全, 不会在浏览器上看到 form 表单中输入的参数 get 不安全, 会在浏览器上看到 form 表单中输入的参数
<b>题目 3)</b>	<b>请分别说出 form 表单中 input 标签文本输入框, 密码框, 单选按钮, 复选框, 下拉列表, 文件域, 隐藏域, 提交按钮, 重置按钮, 图片提交按钮的属性值?</b>
	文本框属性: text 密码框属性: password 单选按钮属性: radio 复选框属性: checkbox 下拉列表属性: <select> <pre> &lt;option value = "1"&gt;Volvo&lt;/option&gt; &lt;option value = "2"&gt;Saab&lt;/option&gt; &lt;option value = "3"&gt;Opel&lt;/option&gt; &lt;option value = "4"&gt;Audi&lt;/option&gt; &lt;/select&gt; </pre> 文件域属性: file 隐藏域属性: hidden 提交按钮属性: submit 重置按钮属性: reset 图片提交按钮属性: image
<b>题目 4)</b>	<b>请说出无序列表的、有序列表、自定义列表的标签?</b>
	无序列表: ul 有序列表: ol 自定义列表: dl
<b>题目 5)</b>	<b>请说出无序列表、有序列表、自定义列表的区别?</b>
	有序列表: 有序列表中默认的 type 类型是数字, 而且是从 1 开始的。它有两个属性: type、start。 无序列表: 无序列表默认的前缀样式是实心圆 (disc), ul 标签中的属性是 type, type 属性值有 disc (实心圆默认)、circle (空心圆)、square (实心正方形)、none (取消前缀) 自定义列表: 自定义列表与有序无序的区别是没有前缀, 而且是有缩进

<b>章节</b>	<b>Ch03 - CSS</b>
<b>题目 1)</b>	<b>什么是 CSS?</b>
	层叠样式表(英文全称: Cascading Style Sheets)是一种用来表现 HTML (标准通用标记语言的一个应用) 或 XML (标准通用标记语言的一个子集) 等文件样式的计算机语言。CSS 不仅

	可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。
<b>题目 2)</b>	<b>请问 CSS 样式有哪些分类？选择器有哪些分类？</b>
	<p>CSS 样式分三类：</p> <p>行内样式：</p> <pre>&lt;div style="width:100px;height:100px;background:red;"&gt;&lt;/div&gt;</pre> <p>内部：&lt;style type="text/css"&gt;&lt;/style&gt;&gt;</p> <p>外部式：就是引用外部 css 文件；</p> <pre>&lt;link href="css.css" type="text/css" rel="stylesheet" /&gt;</pre> <p>CSS 选择器分类</p> <p>基本选择器：</p> <ul style="list-style-type: none"> <li>(1 id 选择器 语法：#id 名{css 样式值; ...}</li> <li>(2 class 选择器 语法：.class 名{css 样式值;...}</li> <li>(3 标签选择器 语法:标签名{css 样式值;...}</li> <li>(4 交集选择器 语法:EF</li> <li>(5 并集选择器 语法: E,F,...</li> <li>(6 选择器的优先级 id 选择器&gt;class 类选择器&gt;标签选择器</li> </ul> <p>层次选择器、结构伪类选择器和属性选择器。</p>
<b>题目 3)</b>	<b>请说出基本选择器和曾次选择器的又细分为哪些选择器？</b>
	<p>基本选择器分为：ID 选择器、class 选择器、标签选择器、交集选择器、并集选择器。</p> <p>层级选择器分为：后代选择器、子选择器、相邻兄弟选择器、同辈选择器。</p>
<b>题目 4)</b>	<b>请说出 CSS 设置字体大小、颜色、粗细、字体行高的语法是哪些？</b>
	<p>字体大小语法：font-size:16px（数字代表像素大小、px 为像素单位）；</p> <p>字体颜色语法：color:16 进制颜色或英语；</p> <p>字体粗细语法：font-weight:bold 或像素；</p> <p>字体行高语法：line-height:像素大小；</p>
<b>题目 5)</b>	<b>请说出文本对齐方式、文本缩进、垂直方向设置、文本阴影的语法是哪些？</b>
	<p>文本对齐方式分三种对齐方式</p> <p>左对齐语法：text-align:left;</p> <p>右对齐语法：text-align:right;</p> <p>居中对齐语法：text-align:center;</p> <p>文本缩进语法分三种缩进：</p> <p>按字符缩进：text-indent: 2em;</p> <p>按长度缩进：text-indent: 2cm;</p> <p>按百分比缩进：text-indent: 50%;</p> <p>垂直方向设置分三种垂直：</p> <p>垂直居中：vertical-align:middle;</p> <p>垂直低端：vertical-align:bottom;</p> <p>垂直顶端：vertical-align:top;</p> <p>文本阴影的语法：text-shadow: 阴影颜色 x 轴 y 轴 阴影半径；</p>

章节	Ch04 - 盒子模型、浮动
题目 1)	什么是盒子模型？什么是浮动？
	<p>盒子模型：网页设计中常听的属性名：内容(content)、填充(padding)、边框(border)、边界(margin)，CSS 盒子模式都具备这些属性。</p> <p>这些属性我们可以用日常生活中的常见事物——盒子作一个比喻来理解，所以叫它盒子模式。CSS 盒子模型就是在网页设计中经常用到的 CSS 技术所使用的一种思维模型。</p> <p>浮动：float 是 css 样式中的定位属性，用于设置标签对象的浮动布局，浮动也就是我们所说标签对象浮动居左靠左 (float:left) 和浮动居右靠右(float:right)。</p>
题目 2)	盒子模型可以设置哪些属性？
	可以设置尺寸、边框、内填充或内边距、边界及外边距、圆角边框、盒子阴影。
题目 3)	浮动可以设置哪些属性？有哪些语法？
	<p>浮动可以设置的属性及语法如下：</p> <p>左浮动：float: left;</p> <p>右浮动：float: right;</p> <p>清楚左侧浮动：clear: left;</p> <p>清楚右侧浮动：clear: right;</p> <p>清除两侧浮动：clear: both;</p>
题目 4)	显示和隐藏元素的语法是什么？
	<p>隐藏：display: none;</p> <p>显示：display: block;</p>
题目 5)	盒子模型的边框有哪些元素？
	<p>上边框：border-top:颜色 大小 样式;</p> <p>下边框：border-bottom:颜色 大小 样式;</p> <p>左边框：border-left:颜色 大小 样式;</p> <p>右边框：border-right:颜色 大小 样式;</p> <p>所有边框：border:颜色 大小 样式;</p> <p>(属性值取值 颜色 (英语单词或者 16 进制的颜色) 大小 (例 1px) 样式 (实线: solid 虚线:dashed))</p>

章节	Ch05 - 定位
题目 1)	相对定位和绝对定位的区别？
	<p>相对定位：设置为相对定位的元素框会偏移某个距离。元素仍然保持其未定位前的形状，它原本所占的空间仍保留。</p> <p>绝对定位：绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。</p>
题目 2)	请说出相对定位和绝对定位的语法？
	<p>相对定位的语法：position: relative;</p> <p>绝对定位的语法：position: absolute;</p>
题目 3)	定位的方式有哪些？

	<p>position 属性值: static, relative, absolute, fixed</p> <p>1) static</p> <p>static 是默认值, 元素位于文档流中, 正常显示, 忽略元素的 top, bottom, left, right 属性。z-index 属性始终为 0.</p> <p>2) relative</p> <p>相对定位, 保留元素在文档流中占用的位置和尺寸, 由 left/right/top/bottom 几个属性确定相对移动的距离, 原来的位置保留</p> <p>3) absolute</p> <p>绝对定位, 将对象从文档流中脱离出来, 使用 left/right/top/bottom 等属性相对其最近的一个并有定位设置的父元素进行绝对定位; 如果没有, 则依据 body 对象</p> <p>4) fixed</p> <p>fixed 和 absolute 差不多, 但是他相对的是窗口的左上角, 不会跟随文档滚动。但是如果在多 frame 页面里面, 他相对的是所在 frame 的左上角, 而不是浏览器左上角。</p>
<b>题目 4)</b>	<b>定位的属性值都有哪些? 每个值得意思?</b>
	<p>static: 默认值。位置设置为 static 的元素会正常显示, 它始终会处于文档流给予的位置 (static 元素会忽略任何 top、bottom、left 或 right 声明)。</p> <p>absolute: 相对于父级元素的绝对定位, 元素脱离文档流, 它不占据空间, 就是我们所说的层, 其位置相对于最近的已定位父元素而言的位置, 可直接指定 "left"、"top"、"right" 以及 "bottom" 属性。若父级都没有定位, 则以 html (根元素)。(层叠的顺序 z-index:value)</p> <p>relative: 是相对于默认位置的偏移定位, 通过设置 left、top、right、bottom 值可将其移至相对于其正常位置的地方 (相对于自己的开始的位置发生的位置上的移动, 【不会破坏正常的布局流】)</p> <p>fixed: 相对浏览器的绝对定位, 是相对于浏览器窗口的指定坐标进行定位。此元素的位置可通过 "left"、"top"、"right" 以及 "bottom" 属性来规定。不论窗口滚动与否, 元素都会留在那个位置。</p>
<b>题目 5)</b>	<b>你对层的设置了解的有哪些</b>
	<p>z-index: 0;</p> <p>默认是 0;</p> <p>设置层的透明度:</p> <p>opacity: x x 值为 0-1, 值越小越透明</p> <p>filter: alpha (opacity=x) x 值为 0-100, 值越小越透明</p>

<b>章节</b>	<b>Ch06 - 1 号店网页制作一 (静态)</b>
<b>题目 1)</b>	<b>每个 HTML 文件里开头都有个很重要的东西, Doctype, 知道这是干什么的吗?</b>
	<!DOCTYPE> 声明位于文档中的最前面的位置, 处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。(重点: 告诉浏览器按照何种规范解析页面)
<b>题目 2)</b>	<b>div+css 的布局较 table 布局有什么优点?</b>
	<p>改版的时候更方便 只要改 css 文件。</p> <p>页面加载速度更快、结构化清晰、页面显示简洁。</p>

	<p>表现与结构相分离。</p> <p>易于优化 (seo) 搜索引擎更友好, 排名更容易靠前。</p>
<b>题目 3)</b>	<b>div+css 的布局较 table 布局有什么优点?</b>
	<p>src 用于替换当前元素, href 用于在当前文档和引用资源之间确立联系。</p> <p>src 是 source 的缩写, 指向外部资源的位置, 指向的内容将会嵌入到文档中当前标签所在位置; 在请求 src 资源时会将其指向的资源下载并应用到文档内, 例如 js 脚本, img 图片和 frame 等元素。</p> <p>&lt;script src = " js.js" &gt;&lt;/script&gt;</p> <p>当浏览器解析到该元素时, 会暂停其他资源的下载和处理, 直到将该资源加载、编译、执行完毕, 图片和框架等元素也如此, 类似于将所指向资源嵌入当前标签内。这也是为什么将 js 脚本放在底部而不是头部。</p> <p>href 是 Hypertext Reference 的缩写, 指向网络资源所在位置, 建立和当前元素 (锚点) 或当前文档 (链接) 之间的链接, 如果我们在文档中添加</p> <p>&lt;link href=" common.css" rel=" stylesheet" /&gt;</p> <p>那么浏览器会识别该文档为 css 文件, 就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用 link 方式来加载 css, 而不是使用 @import 方式。</p>
<b>题目 4)</b>	<b>知道的网页制作会用到的图片格式有哪些?</b>
	<p>png-8, png-24, jpeg, gif, svg。</p> <p>但是上面的那些都不是面试官想要的最后答案。面试官希望听到是 Webp, Apng。(是否有关注新技术, 新鲜事物)</p>
<b>题目 5)</b>	<b>行内元素和块级元素的具体区别是什么? 行内元素的 padding 和 margin 可设置吗?</b>
	<p>块级元素(block)特性:</p> <p>总是独占一行, 表现为另起一行开始, 而且其后的元素也必须另起一行显示;</p> <p>宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;</p> <p>内联元素(inline)特性:</p> <p>和相邻的内联元素在同一行;</p> <p>宽度(width)、高度(height)、内边距的 top/bottom(padding-top/padding-bottom)和外边距的 top/bottom(margin-top/margin-bottom)都不可改变 (也就是 padding 和 margin 的 left 和 right 是可以设置的), 就是里面文字或图片的大小。</p>

<b>章节</b>	<b>Ch07 - 1 号店网页制作二 (静态)</b>
<b>题目 1)</b>	<b>display:none 和 visiblity:hidden 的区别是什么?</b>
	<p>display:隐藏对应元素但不挤占该元素原来的空间。</p> <p>visiblity: 隐藏对应的元素并且挤占该元素原来的空间。</p> <p>即是, 使用 CSS display:none 属性后, HTML 元素 (对象) 的宽度、高度等各种属性都将 “丢失”; 而使用 visiblity:hidden 属性后, HTML 元素 (对象) 仅仅是在视觉上看不见 (完全透明), 而它所占据的空间位置任然存在。</p>
<b>题目 2)</b>	<b>常见的块级元素 内联元素有哪些?</b>
	<p>div -最常用的块级元素</p> <p>dl - 和 dt-dd 搭配使用的块级元素</p> <p>form - 交互表单</p> <p>h1 -h6- 大标题</p>

	<p>hr - 水平分隔线</p> <p>ol - 有序列表</p> <p>p - 段落</p> <p>ul - 无序列表</p> <p>fieldset - 表单字段集</p> <p>colgroup-col - 表单列分组元素</p> <p>table-tr-td 表格及行-单元格</p> <p>pre - 格式化文本</p> <p>a -超链接 (锚点)</p> <p>br - 换行</p> <p>i - 斜体</p> <p>em - 强调</p> <p>img - 图片</p> <p>input - 输入框</p> <p>label - 表单标签</p> <p>span - 常用内联容器, 定义文本内区块</p> <p>strong - 粗体强调</p> <p>sub - 下标</p> <p>sup - 上标</p> <p>textarea - 多行文本输入框</p> <p>u - 下划线</p> <p>select - 项目选择</p>
<b>题目 3)</b>	<b>垂直居中的几个条件?</b>
	<p>三个条件:</p> <p>1: 必须给容器 (父元素) 加上 text-align:center;</p> <p>2:必须给当前元素转成行内块元素 (display:inline-block;) 再给当前元素加上 vertical-align:middle;</p> <p>3: 在当前元素的后面 (没有回车) 加上同级元素 span;并对 span 进行 vertical-align:middle;width:0;height:100%;display:inline-block</p>
<b>题目 4)</b>	<b>写出 html5 新增的标签 15 个</b>
	<p>Header</p> <p>Nav</p> <p>Foot</p> <p>main</p> <p>aside</p> <p>Article</p> <p>section</p> <p>Figure</p> <p>datalist</p> <p>video</p> <p>audio</p> <p>fieldset</p> <p>legend</p> <p>Label</p>

	caption
<b>题目 5)</b>	<b>css 选择符有哪些?</b>
	1) 通配选择符 * 2) 类型选择符 /元素选择符 a 3) 属性选择符 input[type="button"] 4) 包含选择符 div .code a 5) 子对象选择符 ul.test>li 6) ID 选择符 # 7) 类选择符 . 8) 群组选择符 body,ul,li 9) 伪类及伪对象选择符 div:first-letter a:hover 10) 相邻选择符 li+li

<b>章节</b>	<b>Ch08 - JavaScript 基础</b>
<b>题目 1)</b>	<b>什么是 JavaScript?</b>
	JS 是一种页面脚本，通过执行程序脚本片段，我们可以对页面及页面上的元素进行操作，实现特定的功能与效果。 2、列举 Java 和 JavaScript 之间的区别？
<b>题目 2)</b>	<b>列举 Java 和 JavaScript 之间的区别?</b>
	1.js 主要用来向 HTML 页面中添加交互行为。 2.js 是一种脚本语言，语法和 Java 类似。 3.js 是一般用来编写客户端的脚本。下（相当于浏览器网页，运行在客户端环境，用来解释为机器码） 4.js 是一种解释性语言，边执行边解释。（解释性语言：（哪里有错哪里跳过。浏览器解释，js、vbscript）编辑性语言（编译通过才有结果）C++,C#,JAVA) 5.js 是弱语言（弱语言，编写出错，不会报错，需要自己找错误）
<b>题目 3)</b>	<b>JavaScript 组成有哪些?</b>
	由 ECMAScript、浏览器对象模型 BOM、文档对象模型 DOM 三个组成。
<b>题目 4)</b>	<b>JavaScript 的执行原理是什么?</b>
	1.浏览器客户端向服务器发送请求：一个用户在浏览器的地址输入要访问的页面包含 jsp 程序 2.数据处理：服务器端将某个包含 js 的页面进行处理。 3.发送响应：服务器端将某个含有 js 的 html 文件处理页面发送到浏览器客户端，然后由浏览器端从上而下逐条解析 html 的标签和 js，并将页面效果呈现给用户。
<b>题目 5)</b>	<b>页面三层架构是哪三层?</b>
	分别是 行为层 js、表示层 css、结构层 html。  行为层 js： 行为层就是定义网页中各种事件如何触发，甚至实现一些业务逻辑，所以叫行为层  表示层 css： 对结构层中描述的结点定义外观，由 CSS 负责创建。



	<p>结构层 html:</p> <p>由 HTML 或 XHTML 之类的标记语言负责创建。标签, 也就是那些出现在尖括号里的单词, 对网页内容的语义含义做出了描述, 但这些标签不包含任何关于如何显示有关内容的信息</p>
--	--

章节	Ch09 - JavaScript 操作 BOM、DOM 对象
题目 1)	<p><b>在 JavaScript 中如何使用 DOM?</b></p> <p>DOM 代表文档对象模型, 并且负责文档中各种对象的相互交互。DOM 是开发网页所必需的, 其中包括诸如段落, 链接等对象。</p> <p>可以操作这些对象以包括添加或删除等操作, DOM 还需要向网页添加额外的功能。除此之外, API 的使用比其他更有优势。</p>
题目 2)	<p><b>解释 window.onload 和 onDocumentReady?</b></p> <p>在载入页面的所有信息之前, 不运行 onload 函数。这导致在执行任何代码之前会出现延迟。</p> <p>onDocumentReady 在加载 DOM 之后加载代码。这允许早期的代码操纵。</p>
题目 3)	<p><b>JavaScript 中有哪些类型的弹出框?</b></p> <p>有三种类型; Alert、Confirm and、Prompt</p>
题目 4)	<p><b>什么是全局变量? 这些变量如何声明, 使用全局变量有哪些问题?</b></p> <p>全局变量是整个代码长度可用的变量, 也就是说这些变量没有任何作用域。var 关键字用于声明局部变量或对象。如果省略 var 关键字, 则声明一个全局变量。</p> <p>例: // Declare a global globalVariable = "Test" ;</p> <p>使用全局变量所面临的问题是本地和全局变量名称的冲突。此外, 很难调试和测试依赖于全局变量的代码。</p>
题目 5)	<p><b>JavaScript 中如何使用事件处理程序?</b></p> <p>事件是由用户生成活动 (例如单击链接或填写表单) 导致的操作。需要一个事件处理程序来管理所有这些事件的正确执行。事件处理程序是对象的额外属性。此属性包括事件的名称以及事件发生时采取的操作。</p>

章节	Ch10 - jQuery 基础、选择器
题目 1)	<p><b>在公司是怎么用 jquery 的?</b></p> <p>1 基本选择器。</p> <p>#id: id 选择器。</p> <p>.class: 类选择器。</p> <p>标签选择器: 根据指定的元素名匹配元素。</p> <p>群主选择器: 将每一个匹配的元素合并并返回, 以分号隔开。</p> <p>*: 通配符选择器, 匹配所有的元素。</p> <p>2 层次选择器</p> <p>空格: 选取指定元素的后代元素。</p> <p>&gt;: 选取指定下的所有子元素。</p> <p>+: 选取紧接着的兄弟元素。</p> <p>~: 选取元素后的所有兄弟元素。</p>

### 3. 过滤选择器

#### 1 基本过滤选择器

:first(): 选取第一个元素  
:last(): 选取最后一个元素  
:not(selector): 除去所有给定选择器所匹配的元素  
:even(): 选取索引为偶数的所有元素 (索引从 0 开始)  
:odd(): 选取索引为奇数的所有元素 (索引从 0 开始)  
:eq(index): 选取索引=index 的元素 (索引从 0 开始)  
:gt(index): 选取索引>index 的元素 (索引从 0 开始)  
:lt(index): 选取索引<index 的元素 (索引从 0 开始)  
:header(): 选取所有标题  
:animated(): 选取当前正在执行动画的所有元素

#### 2 内容过滤选择器

:contains(text): 选取含有文本内容为"text"的元素。  
:empty(): 选取不包含子元素或文本的空元素  
:has(selector): 选取含有选择器所匹配的元素。  
:parent(): 选取含有子元素或包含文本的元素。

#### 3 可见性选择器

:hidden(): 选取所有不可见元素。  
:visible(): 选取所有可见元素

#### 4 属性选择器

[attribut]: 选取拥有此属性的元素。  
[attribut =value]: 选取属性值为 value 的元素。  
[attribut !=value]: 选取属性值不等于 value 的元素。  
[attribut ^=value]: 选取属性值以 value 开始的元素。  
[attribut \$=value]: 选取属性值以 value 结束的元素。  
[attribut \*=value]: 选取属性值含有 value 的元素。  
[selector1][selector2][selector3]: 用属性选择器合并成一个复合属性选择器, 满足多个条件, 每选择一次, 缩小一次范围。

#### 5 子元素选择器

:nth(安丝)-child(index/even/odd/epuation): 选取每个父元素下的第 index 个子元素或奇偶元素(index 从一算起)。  
:first-child: 选取每个父元素的第一个子元素  
:last-child: 选取每个父元素的最后一个子元素  
:only-child: 如果某个元素是它父元素中唯一的子元素, 那么将会被匹配。如果父元素中含有其它元素, 则不会被匹配。

#### 4. 表单对象属性选择器

:enabled(): 选取所有可用元素。

	:disabled(): 选取所有不可用的元素 :checked(): 选取所有被选中的元素(单选框, 复选框)。 :selected(): 选取所有被选中的选项元素(下拉框)。
<b>题目 2)</b>	<b>为什么要使用 jquery?</b>
	因为 jQuery 是轻量级的框架, 大小不到 30kb, 它有强大的选择器, 出色的 DOM 操作的封装, 有可靠的事件处理机制, 完善的 ajax, 出色的浏览器的兼容性, 而且支持链式操作, 隐式迭代, 行为层和结构层的分离, 还支持丰富的插件, jquery 的文档也非常的丰富。
<b>题目 3)</b>	<b>jquery 中的选择器 和 css 中的选择器有区别吗?</b>
	jQuery 选择器支持 CSS 里的选择器。 jQuery 选择器可用来添加样式和添加相应的行为。 CSS 中的选择器是只能添加相应的样式。
<b>题目 4)</b>	<b>使用 jquery 遇到过哪些问题, 你是怎么解决的?</b>
	1. 前台拿不到值, 是 JSON 可能会出现错误(一般是多了一个空格等), 这编译是不会报错的。 2. 类库冲突: 使用 jQuery.noConflict() 方法将 \$ 控制权渡让给第一个实现它的那个库。 1. 使用 jQuery 替代 \$ 进行渡让。 2. 使用 \$ 以传递参数的形式进行渡让。 3. 使用返回值进行渡让。 4. 使用匿名函数进行渡让。 3. 以前在处理一个数据问题时, 发现 jQuery.ajax() 方法返回的值一直有问题, 清除缓存后数据无误, 多次测试后发现返回的值都是之前的值, 并且一直未执行 url (后台为 JAVA, 设置断点一直未进入)。 后来在网上查找原因发现是未设置 type, 如果没设置 jQuery.ajax 的 type="Post", 那么 ajax 就会默认 type="Get", 这就会导致之前数据被缓存起来。加上 type="Post", 问题解决!
<b>题目 5)</b>	<b>你觉得 jquery 中的选择器有什么优势?</b>
	简单的写法 \$('ID') 来代替 document.getElementById() 函数。 支持 CSS1 到 CSS3 选择器。 完善的处理机制(就算写错了 id 也不会报错)。

<b>章节</b>	<b>Ch11 - jQuery 中的事件</b>
<b>题目 1)</b>	<b>\$(document).ready() 是个什么函数? 为什么要用它?</b>
	ready() 函数用于在文档进入 ready 状态时执行代码。当 DOM 完全加载 (例如 HTML 被完全解析 DOM 树构建完成时), jQuery 允许你执行代码。使用 \$(document).ready() 的最大好处在于它适用于所有浏览器, jQuery 帮你解决了跨浏览器的难题。
<b>题目 2)</b>	<b>JavaScript window.onload 事件和 jQuery ready 函数有何不同?</b>
	JavaScript window.onload 事件和 jQuery ready 函数之间的主要区别是, 前者除了要等待 DOM 被创建还要等到包括大型图片、音频、视频在内的所有外部资源都完全加载。如果加载图片和媒体内容花费了大量时间, 用户就会感受到定义在 window.onload 事件上的代码在执行时有明显的延迟。 另一方面, jQuery ready() 函数只需对 DOM 树的等待, 而无需对图像或外部资源加载的等待, 从而执行起来更快。使用 jQuery \$(document).ready() 的另一个优势是你可以在网页

	里多次使用它, 浏览器会按它们在 HTML 页面里出现的顺序执行它们, 相反对于 onload 技术而言, 只能在单一函数里使用。鉴于这个好处, 用 jQuery ready() 函数比用 JavaScript window.onload 事件要更好些。
<b>题目 3)</b>	<b>jQuery 里的 each() 是什么函数? 你是如何使用它的?</b>
	<p>each() 函数就像是 Java 里的一个 Iterator, 它允许你遍历一个元素集合。你可以传一个函数给 each() 方法, 被调用的 jQuery 对象会在其每个元素上执行传入的函数。有时这个问题会紧接着上面一个问题, 举个例子, 如何在 alert 框里显示所有选中项。我们可以用上面的选择器代码找出所有选中项, 然后我们在 alert 框中用 each() 方法来一个个打印它们, 代码如下:</p> <pre> \$('[name=NameOfSelectedTag] :selected').each(function(selected) {     alert(\$(selected).text()); }); </pre> <p>其中 text() 方法返回选项的文本。</p>
<b>题目 4)</b>	<b>\$(this) 和 this 关键字在 jQuery 中有何不同?</b>
	这对于很多 jQuery 初学者来说是一个棘手的问题, 其实是个简单的问题。\$(this) 返回一个 jQuery 对象, 你可以对它调用多个 jQuery 方法, 比如用 text() 获取文本, 用 val() 获取值等等。而 this 代表当前元素, 它是 JavaScript 关键词中的一个, 表示上下文中的当前 DOM 元素。你不能对它调用 jQuery 方法, 直到它被 \$() 函数包裹, 例如 \$(this)。
<b>题目 5)</b>	<b>JavaScript 函数中自定义函数有哪些?</b>
	1. 定义函数、函数调用、匿名函数、定义一个类、调用类的属性和方法

<b>章节</b>	<b>Ch12 - jQuery 中的 DOM 操作</b>
<b>题目 1)</b>	<b>你是如何将一个 HTML 元素添加到 DOM 树中的?</b>
	可以用 jQuery 方法 appendTo() 将一个 HTML 元素添加到 DOM 树中。这是 jQuery 提供的众多操控 DOM 的方法中的一个。你可以通过 appendTo() 方法在指定的 DOM 元素末尾添加一个现存的元素或者一个新的 HTML 元素。
<b>题目 2)</b>	<b>遍历节点的元素有那些? 请说明一下</b>
	<p>children() : 只考虑子元素, 不考虑后代元素。</p> <p>next() : 同辈紧邻后面一个元素。</p> <p>nextAll() : 同辈紧邻后面所有兄弟元素。</p> <p>prev() : 同辈紧邻前一个兄弟元素。</p> <p>prevAll() : 同辈紧邻前所有兄弟元素。</p> <p>siblings() : 同辈所有兄弟元素。</p> <p>find('span') : 返回被选元素的后代元素, 括号内必填写, 如果查找所有后代使用 "*", 起查找作用。</p>

	<p><code>filter('div')</code>：指定选择器的 xx 元素，括号内必填写，符合条件的同级元素，非后代元素，起过滤作用。</p> <p><code>has( 'div')</code>：符合条件的后代元素，不包含自身，括号内必填写，起过滤作用。</p> <p><code>parents()</code>：获取所有祖先元素，参数为筛选条件。</p> <p><code>closest( 'new')</code>：用来取得最近的匹配元素，包括自身。首选检查自身是否符合，如果符合返回元素本身；如果不匹配，向上查找父元素，逐级向上直到匹配到选择器的元素。如果什么没找到，返回一个空的 jq 对象。必须填写筛选条件，且只能找到一个元素。</p> <p><code>parentsUntil()</code>：截止到 xx 位置的祖先节点。</p>
<b>题目 3)</b>	<b>你是如何删除节点的？</b>
	<p><code>remove()</code>：删除节点，并删除元素节点绑定的事件。</p> <p><code>empty()</code>：清空节点元素的内容，类似 <code>\$.html('')</code>。</p>
<b>题目 4)</b>	<b>操作 DOM 的时插入节点有几种方法，说出用的最多的两个？</b>
	一共有 8 种方法，用的最多的是就是 <code>append</code> 和 <code>appendTo</code> 这两种。
<b>题目 5)</b>	<b>DOM 操作有哪几类？</b>
	<p>DOM 操作有三类：DOM-core、HTML-DOM、CSS-DOM</p> <p>续：</p> <p>请写出 DOM Core 常见的 7 个方法。</p> <p>答案：</p> <ol style="list-style-type: none"> <li>1、创建节点 <code>createElement()</code> <code>createTextNode()</code></li> <li>2、复制节点 <code>cloneNode()</code></li> <li>3、插入节点 <code>appendChild()</code> <code>insertBefore()</code></li> <li>4、删除节点 <code>removeChild()</code></li> <li>5、替换节点 <code>replaceChild()</code></li> <li>6、查找和设置节点属性 <code>setAttribute()</code> <code>getAttribute()</code></li> <li>7、查找节点 <code>getElementById()</code> <code>getElementsByName()</code> <code>hasChildNode()</code></li> </ol>

章节	Ch13 - 表单校验																																				
题目 1)	<b>为什么要使用 JavaScript 表单验证?</b>																																				
	JavaScript 可用来在数据被送往服务器前对 HTML 表单中的这些输入数据进行验证。 被 JavaScript 验证的这些典型的表单数据有： 用户是否已填写表单中的必填项目？ 用户输入的邮件地址是否合法？ 用户是否已输入合法的日期？ 用户是否在数据域 (numeric field) 中输入了文本？																																				
题目 2)	<b>提交表单验证的方式有哪些?</b>																																				
	第一种：在 form 标签中增加 onsubmit 事件来判断表单提交是否成功 第二种：通过 button 按钮来触发表单提交事件 onclick="submitForm();"，会忽略掉其他标签中的属性，比如 form 标签中的 onsubmit 属性就失效了。这时为了进行表单验证，可以将验证代码放在 submitForm();方法中进行验证。 第三种：将 onsubmit 事件放在 submit 标签中，而不是 form 标签中，此时表单验证失效，点击提交按钮表单直接提交 第四种：为 submit 按钮添加上 onclick 事件，其中该事件用于表单提交的验证，功能类似于在 form 标签中增加了 onsubmit 事件一样 第五种：通过 button 按钮来触发表单提交事件 onclick="submitForm();"，会忽略掉其他标签中的属性，比如 form 标签中的 onsubmit 属性就失效了。这时为了进行表单验证，可以将验证代码放在 submitForm();方法中进行验证																																				
题目 3)	<b>jquery 表单选择器有哪些？返回什么？</b>																																				
	<table><tr><th>选择器</th><th>描述</th><th>返回</th><th>示例</th></tr><tr><td>:enabled</td><td>选择所有可用元素</td><td></td><td>集合元素</td></tr><tr><td colspan="4">\$("#form1 :enabled");选取 ID 为"form1"的表单内的所有可用元素</td></tr><tr><td>:disabled</td><td>选取所有不可用元素</td><td></td><td>集合元素</td></tr><tr><td colspan="4">\$("#form2 :disabled");选取 id 为"from2"的表单内的所有不可用元素</td></tr><tr><td>:checked</td><td>选取所有被选中的元素</td><td></td><td>集合元素</td></tr><tr><td colspan="4">\$("input:checked")选取所有被选中的&lt;input&gt;元素</td></tr><tr><td>:selected</td><td>选取所有被选中的选项元素（下拉）</td><td>集合元素</td><td></td></tr><tr><td colspan="4">\$("select:selected");选取所有被选中的选项元素</td></tr></table>	选择器	描述	返回	示例	:enabled	选择所有可用元素		集合元素	\$("#form1 :enabled");选取 ID 为"form1"的表单内的所有可用元素				:disabled	选取所有不可用元素		集合元素	\$("#form2 :disabled");选取 id 为"from2"的表单内的所有不可用元素				:checked	选取所有被选中的元素		集合元素	\$("input:checked")选取所有被选中的<input>元素				:selected	选取所有被选中的选项元素（下拉）	集合元素		\$("select:selected");选取所有被选中的选项元素			
选择器	描述	返回	示例																																		
:enabled	选择所有可用元素		集合元素																																		
\$("#form1 :enabled");选取 ID 为"form1"的表单内的所有可用元素																																					
:disabled	选取所有不可用元素		集合元素																																		
\$("#form2 :disabled");选取 id 为"from2"的表单内的所有不可用元素																																					
:checked	选取所有被选中的元素		集合元素																																		
\$("input:checked")选取所有被选中的<input>元素																																					
:selected	选取所有被选中的选项元素（下拉）	集合元素																																			
\$("select:selected");选取所有被选中的选项元素																																					
题目 4)	<b>你在工作中是怎么使用 jquery 是怎么选取所有密码框的</b>																																				
	请手写以下：\$("password")选取所有的密码框																																				
题目 5)	<b>如果你在工作中有一些特殊处理的内容（时间格式文本）请写出你的代码。</b>																																				
	// 对 Date 的扩展，将 Date 转化为指定格式的 String // 月(M)、日(d)、小时(h)、分(m)、秒(s)、季度(q) 可以用 1-2 个占位符， // 年(y)可以用 1-4 个占位符，毫秒(S)只能用 1 个占位符(是 1-3 位的数字) // 例子： // (new Date()).Format("yyyy-MM-dd hh:mm:ss.S") ==> 2006-07-02 08:09:04.423 // (new Date()).Format("yyyy-M-d h:m:s.S") ==> 2006-7-2 8:9:4.18 Date.prototype.Format = function (fmt) { //author: meizz																																				

	<pre> var o = {     "M+": this.getMonth() + 1, //月份     "d+": this.getDate(), //日     "h+": this.getHours(), //小时     "m+": this.getMinutes(), //分     "s+": this.getSeconds(), //秒     "q+": Math.floor((this.getMonth() + 3) / 3), //季度     "S+": this.getMilliseconds() //毫秒 }; if (/y+/.test(fmt)) fmt = fmt.replace(RegExp.\$1, (this.getFullYear() + "").substr(4 - RegExp.\$1.length)); for (var k in o)     if (new RegExp("(" + k + ")").test(fmt)) fmt = fmt.replace(RegExp.\$1, (RegExp.\$1.length == 1) ? (o[k]) : (("00" + o[k]).substr(("" + o[k]).length))); return fmt; } 调用： var time1 = new Date().Format("yyyy-MM-dd"); var time2 = new Date().Format("yyyy-MM-dd HH:mm:ss"); </pre>
--	---

章节	Ch14 - 1 号店网页制作一（特效）
题目 1)	jquery 对象和 dom 对象是怎样转换的？
	<p>jquery 转 DOM: jQuery 对象是一个数组对象，可以通过[index]得到相应的 DOM 对象，还可以通过 get[index]去得到相应的 DOM 对象。</p> <p>案例如下：</p> <pre> &lt;body&gt;     &lt;input type="text" name="username" id="username" value="小王老师" /&gt;     &lt;script&gt;         //获取 jQuery 对象         var \$username=\$("#username");         alert(\$username.val());          //转换成 DOM 对象         //(1) jQuery 对象是一个数组对象，可以通过[index]的方法，来得到相应的 DOM 对象         var username=\$username[0];         alert(username.value);          //jQuery 本身提供，通过.get(index)方法，得到相应的 DOM 对象         //转化为 DOM 对象         var username1 = \$username.get(0);         alert(username1.value);     &lt;/script&gt; </pre>

	<pre> &lt;/body&gt; DOM 转 jQuery: \$(DOM 对象)。 案例如下: 首先需要引入 jQuery 函数库 &lt;script src="../../AJAX-JavaScript-JQuery/jquery-1.8/jquery-1.8.0.min.js"&gt;&lt;/script&gt;  &lt;body&gt;   &lt;input type="text" name="username" id="username" value="王老师" /&gt;   &lt;script&gt;     //获取 DOM 对象     var username=document.getElementById("username");     alert(username.value);      //转换为 jquery 对象 转换后就可以使用 jQuery 中的方法了     var \$username=\$(username);     alert("^^^"+\$username.val());   &lt;/script&gt;  &lt;/body&gt; </pre>
<b>题目 2)</b>	<b>你是如何使用 jquery 中的 ajax 的?</b>
	<p>如果是一些常规的 ajax 程序的话, 使用 load(),\$.get(),\$.post(), 一般使用的是\$.post() 方法。如果需要设定 beforeSend(提交前回调函数),error(失败后处理),success(成功后处理)及 complete(请求完成后处理)回调函数等, 一般使用的是\$.ajax()。</p> <p>其中 ajax 的基本语法是:</p> <p>1.url: 要求为 String 类型的参数, (默认为当前页地址) 发送请求的地址。</p> <p>2.type: 要求为 String 类型的参数, 请求方式 (post 或 get) 默认为 get。注意其他 http 请求方法, 例如 put 和 delete 也可以使用, 但仅部分浏览器支持。</p> <p>3.timeout: 要求为 Number 类型的参数, 设置请求超时时间 (毫秒)。此设置将覆盖\$.ajaxSetup()方法的全局设置。</p> <p>4.async: 要求为 Boolean 类型的参数, 默认设置为 true, 所有请求均为异步请求。如果需要发送同步请求, 请将此选项设置为 false。注意, 同步请求将锁住浏览器, 用户其他操作必须等待请求完成才可以执行。</p> <p>5.cache: 要求为 Boolean 类型的参数, 默认为 true (当 dataType 为 script 时, 默认为 false), 设置为 false 将不会从浏览器缓存中加载请求信息。</p>



#### 6.data:

要求为 Object 或 String 类型的参数, 发送到服务器的数据。如果已经不是字符串, 将自动转换为字符串格式。get 请求中将附加在 url 后。防止这种自动转换, 可以查看 processData 选项。对象必须为 key/value 格式, 例如{foo1:"bar1",foo2:"bar2"}转换为 &foo1=bar1&foo2=bar2。如果是数组, JQuery 将自动为不同值对应同一个名称。例如 {foo:["bar1","bar2"]}转换为&foo=bar1&foo=bar2。

#### 7.dataType:

要求为 String 类型的参数, 预期服务器返回的数据类型。如果不指定, JQuery 将自动根据 http 包 mime 信息返回 responseXML 或 responseText, 并作为回调函数参数传递。可用的类型如下:

xml: 返回 XML 文档, 可用 JQuery 处理。

html: 返回纯文本 HTML 信息; 包含的 script 标签会在插入 DOM 时执行。

script: 返回纯文本 JavaScript 代码。不会自动缓存结果。除非设置了 cache 参数。注意在远程请求时 (不在同一个域下), 所有 post 请求都将转为 get 请求。

json: 返回 JSON 数据。

jsonp: JSONP 格式。使用 SONP 形式调用函数时, 例如 myurl?callback=?, JQuery 将自动替换后一个 "?" 为正确的函数名, 以执行回调函数。

text: 返回纯文本字符串。

#### 8.beforeSend:

要求为 Function 类型的参数, 发送请求前可以修改 XMLHttpRequest 对象的函数, 例如添加自定义 HTTP 头。在 beforeSend 中如果返回 false 可以取消本次 ajax 请求。

XMLHttpRequest 对象是唯一的参数。

```
function(XMLHttpRequest){
    this;    //调用本次 ajax 请求时传递的 options 参数
}
```

#### 9.complete:

要求为 Function 类型的参数, 请求完成后调用的回调函数 (请求成功或失败时均调用)。参数: XMLHttpRequest 对象和一个描述成功请求类型的字符串。

```
function(XMLHttpRequest, textStatus){
    this;    //调用本次 ajax 请求时传递的 options 参数
}
```

#### 10.success: 要求为 Function 类型的参数, 请求成功后调用的回调函数, 有两个参数。

(1)由服务器返回, 并根据 dataType 参数进行处理后的数据。

(2)描述状态的字符串。

```
function(data, textStatus){
    //data 可能是 xmlDoc、jsonObj、html、text 等等
    this;    //调用本次 ajax 请求时传递的 options 参数
}
```

#### 11.error:

要求为 Function 类型的参数, 请求失败时被调用的函数。该函数有 3 个参数, 即 XMLHttpRequest 对象、错误信息、捕获的错误对象(可选)。ajax 事件函数如下:

```
function(XMLHttpRequest, textStatus, errorThrown){  
    //通常情况下 textStatus 和 errorThrown 只有其中一个包含信息  
    this;    //调用本次 ajax 请求时传递的 options 参数  
}
```

#### 12.contentType:

要求为 String 类型的参数, 当发送信息至服务器时, 内容编码类型默认为 "application/x-www-form-urlencoded"。该默认值适合大多数应用场合。

#### 13.dataFilter:

要求为 Function 类型的参数, 给 Ajax 返回的原始数据进行预处理的函数。提供 data 和 type 两个参数。data 是 Ajax 返回的原始数据, type 是调用 jQuery.ajax 时提供的 dataType 参数。函数返回的值将由 jQuery 进一步处理。

```
function(data, type){  
    //返回处理后的数据  
    return data;  
}
```

#### 14.dataFilter:

要求为 Function 类型的参数, 给 Ajax 返回的原始数据进行预处理的函数。提供 data 和 type 两个参数。data 是 Ajax 返回的原始数据, type 是调用 jQuery.ajax 时提供的 dataType 参数。函数返回的值将由 jQuery 进一步处理。

```
function(data, type){  
    //返回处理后的数据  
    return data;  
}
```

#### 15.global:

要求为 Boolean 类型的参数, 默认为 true。表示是否触发全局 ajax 事件。设置为 false 将不会触发全局 ajax 事件, ajaxStart 或 ajaxStop 可用于控制各种 ajax 事件。

#### 16.ifModified:

要求为 Boolean 类型的参数, 默认为 false。仅在服务器数据改变时获取新数据。服务器数据改变判断的依据是 Last-Modified 头信息。默认值是 false, 即忽略头信息。

#### 17.jsonp:

要求为 String 类型的参数, 在一个 jsonp 请求中重写回调函数的名字。该值用来替代在 "callback=?" 这种 GET 或 POST 请求中 URL 参数里的 "callback" 部分, 例如 {jsonp:'onJsonPLoad'}会导致将 "onJsonPLoad=?" 传给服务器。

#### 18.username:

要求为 String 类型的参数, 用于响应 HTTP 访问认证请求的用户名。

	<p>19.password: 要求为 String 类型的参数, 用于响应 HTTP 访问认证请求的密码。</p> <p>20.processData: 要求为 Boolean 类型的参数, 默认为 true。默认情况下, 发送的数据将被转换为对象 (从技术角度来讲并非字符串) 以配合默认内容类型"application/x-www-form-urlencoded"。如果要发送 DOM 树信息或者其他不希望转换的信息, 请设置为 false。</p> <p>21.scriptCharset: 要求为 String 类型的参数, 只有当请求时 dataType 为"jsonp"或者"script", 并且 type 是 GET 时才会用于强制修改字符集(charset)。通常在本地和远程的内容编码不同时使用。</p> <p>案例代码:</p> <pre>\$(function(){     \$('#send').click(function(){         \$.ajax({             type: "GET",             url: "test.json",             data: {username:\$("#username").val(), content:\$("#content").val()},             dataType: "json",             success: function(data){                 \$('#resText').empty();    //清空 resText 里面的所有内容                 var html = "";                 \$.each(data, function(commentIndex, comment){                     html += '&lt;div class="comment"&gt;&lt;h6&gt;' +                         comment['username']                         + '&lt;/h6&gt; &lt;p class="para"&gt;' +                         comment['content']                         + '&lt;/p&gt;&lt;/div&gt;';                 });                 \$('#resText').html(html);             }         });     }); });</pre>
题目 3)	<p><b>阐述一下异步加载 JS。</b></p> <ol style="list-style-type: none"> <li>1. 异步加载的方案: 动态插入 script 标签</li> <li>2. 通过 ajax 去获取 js 代码, 然后通过 eval 执行</li> <li>3. script 标签上添加 defer 或者 async 属性</li> </ol>

<b>题目 4)</b>	<b>在 jquery 中你是如何去操作样式的?</b>
	<p>attr()获取或设置样式。</p> <p>addClass() 来追加样式。</p> <p>removeClass() 来删除样式。</p> <p>toggle() 来切换样式。</p> <p>hasClass() 判断样式。</p> <p>5.什么是 Ajax 和 JSON, 它们的优缺点。</p>
<b>题目 5)</b>	<b>什么是 Ajax 和 JSON, 它们的优缺点。</b>
	<p>Ajax 是全称是 asynchronous JavaScript andXML, 即异步 JavaScript 和 xml, 用于在 Web 页面中实现异步数据交互, 实现页面局部刷新。</p> <p>优点: 可以使得页面不重载全部内容的情况下加载局部内容, 降低数据传输量, 避免用户不断刷新或者跳转页面, 提高用户体验</p> <p>缺点: 对搜索引擎不友好; 要实现 ajax 下的前后退功能成本较大; 可能造成请求数的增加跨域问题限制;</p> <p>JSON 是一种轻量级的数据交换格式, ECMA 的一个子集</p> <p>优点: 轻量级、易于人的阅读和编写, 便于机器 (JavaScript) 解析, 支持复合数据类型 (数组、对象、字符串、数字)</p>

<b>章节</b>	<b>Ch15 - 1 号店网页制作二 (特效)</b>
<b>题目 1)</b>	<b>你一般用什么去提交数据, 为什么?</b>
	<p>一般我会使用的是\$.post() 方法。</p> <p>如果需要设定 beforeSend(提交前回调函数),error(失败后处理),success(成功后处理) 及 complete(请求完成后处理)回调函数等, 这个时候我会使用\$.ajax()</p>
<b>题目 2)</b>	<b>如何来设置和获取 HTML 和文本的值?</b>
	<p>html()方法 类似于 innerHTML 属性 可以用来读取或者设置某个元素中的 HTML 内容 注意: html() 可以用于 xhtml 文档 不能用于 xml 文档</p> <p>text() 类似于 innerText 属性 可以用来读取或设置某个元素中文本内容。</p> <p>val() 可以用来设置和获取元素的值</p>
<b>题目 3)</b>	<b>你在 ajax 中使用过 JSON 吗, 你是如何用的?</b>
	<p>使用过, 在\$.getJSON() 方法的时候就是。</p> <p>因为 \$.getJSON() 就是用于加载 JSON 文件的</p>
<b>题目 4)</b>	<b>\$.getScript()方法和 \$.getJSON() 方法有什么区别?</b>
	<p>1 \$.getScript() 方法可以直接加载.js 文件, 并且不需要对 javascript 文件进行处理, javascript 文件会自动执行。</p> <p>2 \$.getJSON() 是用于加载 JSON 文件的, 用法和\$.getScript()</p>
<b>题目 5)</b>	<b>怎样给 jquery 动态附加新的元素?那么怎样给新生成的元素绑定事件呢?</b>
	<p>jQuery 的 html()可以给现在元素附加新的元素</p> <p>直接在元素还未生成前就绑定肯定是无效的, 因为所绑定的元素目前根本不存在。</p>

所以可以通过 live 和 livequery 来动态绑定事件
---------------------------------

菜鸟教程