

1. jQuery事件注册

单个事件注册

语法:

```
element.事件(function(){});  
  
$('div').click(function(){  
  
}
```

其他事件和原生基本一致

比如: mouseover mouseout blur focus change keydown keyup resize scroll等

1.1. 事件处理on()绑定事件

on()方法在匹配元素上绑定一个或多个事件的事件处理函数

语法:

```
element.on(events,[selector],fn)
```

- events:一个或多个用空格分隔的事件类型, 如click 或keydown
- selector: 元素的子元素选择器
- fn:回调函数, 即绑定在元素身上的侦听函数

```
$('div').on({  
  mouseenter: function () {  
    $(this).css('background', 'skyblue');  
  },  
  click: function () {  
    $(this).css('background', 'red');  
  },  
  mouseleave: function () {  
    $(this).css('background', 'green');  
  }  
});
```

如果事件处理程序相同:

```
$('#div').on('mouseenter mouseleave',function(){
    // alert(111);
    $(this).toggleClass('current');
})
```

可以事件委派操作，事件委派的定义就是，把原来加给子元素的身上的事件绑定在父元素身上，就是把事件委派给父元素

```
$('#ul').on('click','li',function(){
    alert('hello world');
})
```

在此之前有bind() live() delegate()等方法来处理事件绑定或者事件委派，最新版本的请用on来代替他们

动态创建的元素，click() 没有办法绑定事件，on()可以给动态生成的元素绑定事件

1.2. 事件处理off()解绑事件

off()方法可以移除通过on()方法添加的事件处理程序

```
$('#p').off()    //解绑p元素所有事件处理程序
$('#p').off('click') //解绑p元素上面的点击事件 后面的off是侦听函数名
$('#ul').off('click','li'); //解绑事件委托
```

如果有的事件只想触发一次，可以使用one()来绑定事件

2.3. 自动触发事件trigger()

有些事情希望自动触发，比如轮播图自动播放功能跟点击功能一致。可以利用定时器自动触发右侧按钮点击事件，不必鼠标点击触发

```
element.click()    第一种简写形式

element.trigger('type') //第二种自动触发模式

element.triggerHandler(type) //第三种自动触发模式
```

2. jQuery事件对象

事件被触发，就会有事件对象的产生

```
element.on(events,[selector],function(event){})
```

阻止默认行为: `event.preventDefault()` 或者 `return false`

阻止冒泡: `event.stopPropagation()`

3. jQuery对象拷贝

如果想要把某个对象拷贝（合并）给另外一个对象使用，此时可以使用`$.extend()`方法

语法：

```
$.extend([deep],target,object1,[objectN]);
```

- `deep`:如果设为`true`为深拷贝，默认为`false` 浅拷贝
- `target`:要拷贝的目标对象
- `object1`: 待拷贝到第一个对象的对象

浅拷贝是把被拷贝的对象复杂数据类型中的地址拷给目标对象，修改目标对象会影响被拷贝对象
深拷贝 前面加`true`，完全克隆（拷贝的对象，而不是地址），修改目标对象不会影响被拷贝对象

4. jQuery多库共存

问题概述：

jQuery使用作为标识符，随着*jQuery*的流行，其他*js*库也会用这作为标识符，这样一起使用就会引起冲突

客观需求：

需要一个解决方案，让jQuery和其他的js库不存在冲突，可以同时存在，这就叫做多库共存

jQuery解决方案：

把里面的\$符号统一改为jQuery。比如：`jQuery('div')`

jQuery变量规定新的名称：`.noConflict()varxx =.noConflict();`

5. jQuery插件

jQuery功能比较有限，想要更复杂的特效效果，可以借助jQuery插件完成

注意：这些插件也是依赖jQuery来完成了，所以必须要先引入jQuery文件，因此也称为jQuery插件

jQuery插件常用网址：

jQuery插件库：<http://www.jq22.com/>

jQuery之家：<http://www.htmleaf.com/>

jQuery插件使用步骤：

- 引入相关文件（jQuery文件和插件文件）
- 复制相关的HTML css js（调用插件）

图片懒加载（图片使用延迟加载在可提高网页下载速度，它也能帮助减轻服务器负载）
当我们的页面滑动到可视区域，再显示图片

vscode查找替换快捷键：`Ctrl+h`

我们使用jQuery插件库EasyLazyload 注意：此时的js引入文件和js调用必须写到DOM元素（图片）最后面

全屏滚动（fullpages.js）

bootstrap js插件

bootstrap框架也是依赖于jQuery开发的，因此里面的js插件使用，也必须引入jQuery文件

综合案例todolist

todolist分析

刷新页面不会丢失数据，因此需要本地存储localStorage

核心思路：不管按下回车，还是点击复选框，都是把本地存储的数据加载到页面中，这样保证刷新关闭页面不会丢失数据

数据存储的数据格式：`var todolist=[{title:'xxx',done:false}]`

注意点1：本地存储的localStorage里面只能存储字符串格式，因此需要把对象转换为字符串
`JSON.stringify(data)`

注意点2：获取本地存储数据，需要把里面的字符串转换为JSON.parse()我们才能使用里面的数据

todolist按下回车把新数据添加到本地存储里面

切记：页面中的数据，都要从本地存储里面获取，这样刷新时不会丢失数据，所以要先把数据保存到本地存储里面

利用事件对象.keyCode判断用户按下回车键 (13)

声明一个数组，保存数据

先要读取本地存储原来的数据 (声明函数getData()),放到这个数组里面

之后把最新从表单获取过来的数据追加到数组里面

最后把数组存储给本地存储 (声明函数saveDate())

todolist本地存储数组渲染加载到页面

因为后面经常渲染加载操作，所以声明一个函数load方便后面调用

先要读取本地存储数据 (数据不要忘记转换为对象格式)

之后遍历这个数据(\$.each()),有几条数据，就生成几个li添加到ol里面

每次渲染之前，先把原先里面的ol的内容清空，然后渲染 加载最新的数据

todolist删除操作：

点击里面的a链接，不是删除的li，而是删除本地储存对应的数据

核心原理：先获取本地储存数据，删除对应的数据，保存给本地存储，重新渲染列表li

我们可以给链接自定义记录当前的索引号

根据这个索引号删除相关的数据--数组的splice(i,1)方法

存储修改后的数据然后存储给本地存储

重新渲染加载数据列表

todolist正在进行和已完成选项操作

当我们点击了小的复选框，修改本地存储数据，再重新渲染数据列表

点击之后，获取本地存储数据

修改对应数据属性done为当前复选框的checked状态

之后保存到本地存储里面

重新渲染加载数据列表

load加载函数里面新增一个条件，如果当前数据的done为true就是已经完成的，就把列表渲染加载到ul里面

如果当前数据的done为false则是待办事项，就把列表渲染加载到ol里面