04【jQuery】 jQuery 其他方法

学习内容： jQuery 拷贝对象，多库共存，jQuery 插件

1. 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])

1. **deep:** 如果设为true 为深拷贝， 默认为false 浅拷贝
2. target: 要拷贝的目标对象 3. object1:待拷贝到第一个对象的对象。
3. objectN:待拷贝到第N个对象的对象。
4. **浅拷贝** 目标对象引用的被拷贝的对象地址，修改目标对象**会影响**被拷贝对象。
5. **深拷贝**，前面加**true**， 完全克隆，修改目标对象**不会影响**被拷贝对象。

2. jQuery 多库共存

**问题概述：** jQuery使用$作为标示符，随着jQuery的流行,其他 js 库也会用这$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：**

1. 把里面的 $ 符号 统一改为 jQuery。 比如 jQuery(''div'')

2. jQuery 变量规定新的名称：$.noConflict() var xx = $.noConflict();

3. jQuery 插件

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

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

**jQuery 插件常用的网站：**

1. jQuery 插件库 <http://www.jq22.com/>

2. jQuery 之家 http://www.htmleaf.com/ jQuery

**插件使用步骤：**

1. 引入相关文件。（jQuery 文件 和 插件文件）

2. 复制相关html、css、js (调用插件)。

jQuery 插件演示：

1. 瀑布流

2. 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）

当我们页面滑动到可视区域，再显示图片。

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

3. 全屏滚动（fullpage.js） gitHub： <https://github.com/alvarotrigo/fullPage.js>

中文翻译网站： <http://www.dowebok.com/demo/2014/77/>

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。

**【案例】：toDoList**

① 文本框里面输入内容，**按下回车，就可以生成待办事项。**

② 点击待办事项复选框，就可以把当前数据添加到已完成事项里面。

③ 点击已完成事项复选框，就可以把当前数据添加到待办事项里面。

**④ 但是本页面内容刷新页面不会丢失。**

【案例分析】

① 刷新页面不会丢失数据，因此需要用到本地存储 localStorage

**② 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**

③ 存储的数据格式：var todolist = [{ title : ‘xxx’, done: false}]

④ 注意点1： 本地存储 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)。

⑤ 注意点2： 获取本地存储数据，需要把里面的字符串转换为对象格式JSON.parse() 我们才能使用里面的数据。

**【案例】：toDoList 按下回车把新数据添加到本地存储里面**

① 切记： 页面中的数据，都要从本地存储里面获取，这样刷新页面不会丢失数据，所以先要把数据保存到本地存储里面。

② 利用事件对象.**keyCode**判断用户按下回车键（13）。

③ **声明一个数组，保存数据。**

④ 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。

⑤ 之后把 **最新从表单获取过来的数据**，追加到数组里面。

⑥ 最后 把数组存储给本地存储 (声明函数 savaDate())

**【案例】：toDoList 本地存储数据渲染加载到页面**

① 因为后面也会经常渲染加载操作，所以声明一个函数 load，方便后面调用

② 先要读取本地存储数据。（数据不要忘记转换为对象格式）

③ 之后遍历这个数据（$.each()），有几条数据，就生成几个小li 添加到 ol 里面。

④ 每次渲染之前，先把原先里面 ol 的内容清空，然后渲染加载最新的数据。

【案例】：toDoList 删除操作

① 点击里面的a链接，不是删除的li，而是删除本地存储对应的数据。

**② 核心原理：先获取本地存储数据，删除对应的数据，保存给本地存储，重新渲染列表li**

③ 我们可以给链接自定义属性记录当前的索引号

④ 根据这个索引号删除相关的数据----数组的splice(i, 1)方法

⑤ 存储修改后的数据，然后存储给本地存储

⑥ 重新渲染加载数据列表

⑦ 因为a是动态创建的，我们使用on方法绑定事件

【案例】案例：toDoList 正在进行和已完成选项操作

① 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。

② 点击之后，获取本地存储数据。

③ 修改对应数据属性 done 为当前复选框的checked状态。

④ 之后保存数据到本地存储

⑤ 重新渲染加载数据列表

⑥ 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

⑦ 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

【案例】：toDoList 统计正在进行个数和已经完成个数

① 在我们load 函数里面操作

② 声明2个变量 ：todoCount 待办个数 doneCount 已完成个数

③ 当进行遍历本地存储数据的时候，如果数据done为false， 则 todoCount++, 否则 doneCount++

④ 最后修改相应的元素 text()