正课:

1. 动画

2. 类数组对象操作

3. 添加自定义API

4. 插件\*\*\*\*\*

鄙视: $的原理:

$其实是创建jQuery类型子对象的意思，包括以下四种情况:

1. 如果传入选择器，$会查找DOM元素，并保存进jquery对象中。

优化: 如果传入的选择器简单，则调用按HTML查找的API  
 如果传入的选择器复杂，则调用按选择器查找的API

2. 如果传入DOM元素，$会直接封装DOM元素进jQuery对象

3. 如果传入HTML片段，$会创建新元素，封装进jQuery对象

4. 如果传入函数，$会绑定DOM内容加载后执行的事件

1. 动画:

2大类API:

1. 简单动画: 效果固定的三组API:

1. 显示隐藏: .show([ms]) .hide([ms]) .toggle([ms])

默认无参时: 封装的是: .style.display="block"|"none"

设置时间参数，可增加动画效果

2. 上滑下滑: .slideUp([ms]).slideDown([ms]) .slideToggle ()

3. 淡入淡出: .fadeIn([ms]) .fadeOut([ms]) .fadeToggle()

问题:

1. jQuery动画API都是用js的定时器实现的，效率不如css transition。

2. 效果是写死的，不可维护！

总结: 不推荐使用jQuery简化版动画API

但是: .show()/hide()/toggle()不加参数的情况是推荐的

2. 万能动画: 可对大多数css属性自定义动画效果

$().animate({

css属性: 目标值,

... : ...

},ms) //可让当前元素从当前状态，经过指定事件，过渡到目标样式

问题: 只支持单个数值的属性，甚至不支持颜色动画

排队和并发:

并发: 多个css属性同时变化

如何: 放在同一个animate内的多个css属性是并发变化

排队: 多个css属性先后依次变化

如何: 对同一个元素，先后调用多次animate，多个animate是不排队执行

动画结束后自动执行: 其实每个动画API都有最后一个回调函数参数。回调函数会在动画结束后自动执行

停止动画: $().stop()

问题:默认只停止当前正在播放的一个动画，队列中其它动画继续执行

解决: $().stop(true) 停止当前动画并清空队列

选择正在播放动画的元素: 选择器: :animated

2. 类数组对象操作:

jQuery为类数组对象，提供了简化版的API:

1. 遍历:

遍历当前jquery对象:

$().each(function(i,elem){ //实例方法

//i获得当前遍历到的位置

//elem获得当前正在遍历的DOM元素

//each中的this->elem，所以参数elem可省略

})

遍历所有类数组对象:

$.each(类数组对象,function(i,elem){ ... })//静态方法

2. 查找:

$("所有元素").index("要查找的元素")

3. 添加自定义API:

1. 在专门的.js文件中，定义函数并添加到jQuery.fn中:

jQuery.fn.自定义API=function(){

//this->将来调用改API的.前的jQuery对象

//因为已经是jQuery了，所以不用再$(this)

}

2. 在网页中，先引入jquery.js，再引入自定义.js文件，再编写自定义脚本，调用自定义API: $().自定义API()

4. 插件:

什么是: 拥有独立HTML内容，CSS样式和JS功能的网页部分。

为什么: 重用

何时: 只要页面中部分功能可能反复使用

如何:

选择: 3种来源:

1. jQuery官方: jQuery UI

1. 从官方下载

2. 按插件的要求，定义HTML内容

3. 先引入插件的css文件

4. 引入jquery.js和插件.js

5. 在自定义脚本中，找到插件的父元素，调用插件API

$("插件父元素").插件API()

插件API和插件名相同

特点:

侵入性: 插件根据自身的需要，为元素自动添加样式和属性。

优点: 简单！

缺点: 不便于维护！

2. 第三方插件:

3. 自定义插件:

前提: 已经用HTML,CSS,JS实现了插件效果

2种风格:

1. jQuery UI式（侵入式）

何时: 小范围使用，不考虑过多变化时——简单

如何:

封装:

1. 将插件的css提取到一个专门的.css文件中

要求: 一个插件内的css选择器必须以统一的父元素器开头！

2. 在专门的.js文件中，为jQuery类型添加自定义的API

在自定义API中，做两件事:

1. 自动查找子元素，侵入class和自定义扩展属性

2. 事件绑定

使用:

1. 按插件要求编写HTML，不用带class

2. 引入插件的css

3. 引入jquery.js和插件.js

4. 查找插件元素的父元素，调用插件函数

$("#my-accordion").accordion()

2. Bootstrap式（DIY式）

何时: 大范围公开使用时，必须考虑各种变化——可维护

如何:

封装:

1. 将插件的css提取到一个专门的.css文件中

要求: 一个插件内的css选择器必须以统一的父元素器开头！

2. 在插件.js中，自动查找带自定义扩展属性的元素，自动绑定事件。

使用:

1. 按插件要求编写HTML

2. 引入插件css

手动在HTML的元素上按需定义class

3. 引入jquery.js和插件.js