正课:

1. 事件:

2. 动画:

3. 类数组对象操作:

4. 添加自定义API:

1. 事件:

1. 页面加载后执行: 2个时机3 1

1. DOM内容加载完成: 仅包括html, js

DOMContentLoaded

何时: 如果不依赖css和图片的初始化操作(比如,事件绑定)，都应该放在DOMContentLoaded中提前执行，让用户可以不必等待图片和css，就可提前使用

如何: $(document).ready(function(){

... ...

})

简化: $().ready(function(){

更简化: $(function(){ ... }); （$第四种重载）

鄙视: $的原理:**四种重载**

1. 查找并封装DOM元素

如果选择器简单，使用的是按HTML查找

如果选择器复杂，才使用querySelectorAll

2. 直接封装现有DOM元素

3. 创建新元素或片段

4. 绑定DOM内容加载提前执行的处理函数

2. 整个页面加载完成: 包括: html, js, css, 图片 .... **3 1**

window.onload=function(){ ... }

何时: 必须依赖css或图片的行为，必须放在onload中

问题: **赋值方式绑定**事件，只能绑定一个处理函数，后绑定会把先绑定的覆盖掉

解决: $(window).load(function(){ ... }) （给window绑事件）

2. 鼠标事件:

mouseover mouseout

问题: 进出子元素，也会频繁触发父元素上的处理函数

解决: mouseenter mouseleave 3 2

优点: 即使频繁进出子元素，也不会反复触发父元素的处理事件

更简化: 如果同时绑定enter和leave，就可以只绑定hover

.hover( //相当于同时绑enter和leave 3 3

function(){ ... }, //给enter

function(){ ... } //给leave

)

更简化: 如果两个处理函数可统一为一个函数 3 3

.hover(

function(){ ... } //即给enter，又给leave

)

3. 模拟触发: 3 4

$elem.trigger("事件名")  *trigger：触发*

可简写为: $elem.事件名();

2. 动画:

1. 简单动画: 3组效果:3 5 3 6品牌显示 3 7手风琴

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

强调: .show()/.hide() 如果不加参数，默认是display:block/none，不带动画效果——推荐

上滑下滑（隐藏显示）: .slideUp(ms) .slideDown(ms) **.slideToggle(ms)**

淡入淡出: .fadeIn() .fadeOut()  **.fadeToggle()**

原理: 用js的定时器修改的内联样式:

问题: 1. 效率不如css的transition

2. 写死的效果，不可维护

*扣动画：看源代码有没有显示样式变化：有,jq做的,去js中找；没:css做的,去css中找*

2. 万能动画: 可对多数css属性选择应用动画效果 pm

$elem.animate({

css属性名: 目标值,

... : ...

},ms)

强调: 只支持单个数值的css属性

排队和并发:

1. 并发: 多个css属性，同时变化

如何: 在一个animate内的多个css属性，默认并发

2. 排队: 多个css属性，排队先后变化

如何: 对一个元素，先后调用多个animate，则排队执行

其它动画API，也有排队效果。

本质: 调用动画API，不是立刻启动动画的意思。只是将动画加入队列而已。

动画播放后执行:

每个动画API的最后一个参数，都是一个回调函数，会在动画播放后自动执行!

停止动画: .stop()

问题: 默认只能停止队列中当前正在播放的一个动画，后续动画依然执行

解决: .stop(true) 清空队列

选择器: :animated 选择或判断一个正在播放动画的元素

延迟执行: .delay(ms)

3. 类数组对象操作:

each: 对类数组对象中每个元素执行相同操作

2种用法:

1. 仅遍历jQuery对象中的每个DOM元素

$(...).each(function(i,elem){

//i 当前位置

//elem 当前元素对象

})

2. 可对所有索引数组和类数组对象，提供遍历操作

$.each(arr/类数组对象, function(i,elem){

... ...

})

鄙视: $(...).each() vs $.each()

其实就是实例方法和静态方法的比较

index: 在数组中查找一个指定元素的位置

var i=$(所有元素).index(要找的元素);

简写: 如果在同一个父元素下找直接子元素的位置

$(子元素).index()

4. 添加自定义方法:

3步:

1. 验证是否先引入了jQuery

2. 验证jQuery的原型对象中是否包含该方法

3. 向jQuery的原型对象中添加新方法

this->$( ... ) 不用$(this)封装!