正课:

1. 替换/克隆:

2. 事件:

3. 动画:

4. 类数组对象API:

5. 添加自定义API:

1. 替换/克隆:

克隆: var $newImg=$img.clone()

2. 事件:

标准:

1. 普通绑定: $().on("事件名",function(){ ... })

.addEventListener(...);

解绑: $().off("事件名",原函数对象)

2. 利用冒泡:

$parent.on("事件名","判断条件选择器",function(){

this->e.target

})

2个好处:  
 1. 自动使用"判断条件选择器"先筛选符合条件的元素再进入函数执行——不用自己写if判断

2. 将this->e.target，不用自己写e和e.target了

21种常见事件提供了简写: $().事件名(处理函数)

页面加载后执行:

1. 仅DOM内容加载后就提前执行: DOMContentLoaded

仅等待DOM内容: HTML和JS

可执行: 事件绑定，添加/删除/遍历/修改元素的内容

不能执行: 获取/修改CSS, 获取图片大小

如何: $(document).ready(function(){ ... })

$().ready(function(){ ... })

$(function(){ ... })

总结: 几乎所有jQuery代码都要放在$(function(){...})中

2. 所有页面内容加载完成后，才执行: window.onload

等待所有网页内容: HTML,CSS,JS,图片

都可执行

如何: $(window).load(function(){ ... })

总结: 只有依赖CSS和图片的代码才放在$(window).load()

鼠标事件:

mouseover mouseout的问题:

频繁进出子元素，也会冒泡触发父元素上的鼠标事件

jQuery改为用mouseenter和mouseleave代替

优点: 即使频繁进出子元素，也不会冒泡触发父元素上的鼠标事件。

如果同时绑定了mouseenter和mouseleave，其实只需要绑定一个hover就可以:

$().hover(//mouseenter+mouseleave

function(){ ... },//给mouseenter

function(){ ... } //给mouseleave

)

如果两个处理函数刚巧可以合并为一个处理函数，其实只写一个处理函数也行

模拟触发: 没有单击到按钮，也能执行按钮的处理函数，而不用将代码写2遍

$(btn).trigger("事件名") $(btn).trigger("click")

其实可简写为:$(btn).click()

3. 动画:

1. 简单动画: 写死的三种固定动画效果:

1. 显示隐藏: .show() .hide() .toggle()

不带参数，默认使用display:block/none控制显示隐藏，瞬间变化，没有动画效果

给一个持续时间参数: .show(ms) .hide(ms) .toggle(ms)

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

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

问题1: 用js代码写死的效果，不可维护

问题2: 用js定时器实现的动画效果，效率不如css动画

2. 万能动画: 可对任意css属性实施动画效果

$().animate({

css属性:目标值,

... : ...

},持续ms)

强调: 只支持单个数值的属性

不支持:不是数值的属性

不支持颜色渐变——不如transition

排队和并发:

排队: 多个css属性先后依次变化

并发: 多个css属性同时变化

放在一个animate函数内的多个css属性是并发变化

停止动画: $().stop();

问题: 默认只能停止队列中正在播放的一个动画，后续动画依然会执行

解决: $().stop(true) 停止当前动画并清空队列

选择器: :animated 选择和判断正在播放动画的元素

动画播放后执行:

每个动画API都有最后一个参数——回调函数

回调函数会在动画播放后自动调用执行

总结: 首选CSS: 效率高，可维护

和交互有关的动画，只能用JS定时器/JQUERY

和动态添加/删除DOM元素有关的动画，也必须用JS定时器/jQuery

4. 类数组操作:

2个API:

1. 遍历:

//实例方法

$()->jQuery()->new jQuery()

$().each(function(i,elem){ //forEach(function(elem,i,arr))

//i 获得当前位置

//elem 获得当前元素对象

})

//静态方法

//$->jQuery

$.each(类数组对象/数组,function(i,elem){

})

2. 查找:

$("所有元素").index("要找的元素")

在"所有元素"的jQuery对象中，查找"要找的元素"的位置

如果在同一个父元素下找子元素的位置:

$(要找的子元素).index()

5. 添加自定义API:

其实就是在jQuery的原型对象中添加自定义函数:

jQuery.fn=jQuery.prototype

定义新函数:

jQuery.fn.新函数=function(){

}

调用新函数:

$(...).新函数()