正课:

1. 事件:

2. 动画:

3. 类数组对象操作:

1. 事件:

绑定:

已废弃: $(...).live("事件名",fn) delegate

.die("事件名",fn)

将网页中所有事件集中绑定在document上。效率极低

终极简化: $(...).on(...) 绑定

.off(...) 解绑

$(...).on("事件名",fn) -> bind

$(...).on("事件名","selector",fn) -> delegate

强调: on代替delegate，要事件名在前，选择器在后

在绑定事件时，向事件处理函数中传入数据:

问题: 不能将自定义数据，直接放入处理函数的参数中，在事件触发时，会被事件对象替换。

解决: 使用绑定方法的data参数，传入对象。在对象中携带数据。

$(...).on("事件名",data,function(e){

{x:1}

e->event

e.data.x->1

{x:1}

})

模拟触发: 用代码模拟触发某个元素上绑定的事件处理函数

$(...).trigger("事件名") 触发找到的元素上绑定的事件函数

页面加载后自动执行:

DOMContentLoaded : DOM内容加载完成就触发

HTML和JS加载完成后，就提前触发

准备工作包括: 事件绑定，DOM操作

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

简写: $().ready(function(){....})

$(function(){....}) 反复$()的fn，默认DOM内容加载后就提前执行

其实: 都不用

只要放在body结尾的script中的代码，默认DOM内容加载后提前执行。

window.onload: 整个页面内容加载完成才触发

HTML, JS, CSS, 图片全部完成，才触发

特殊的事件: hover = mouseover+mouseout

何时: 只要同时响应mouseover和mouseout时，可简写：

$(...).hover(

function(){...mouseover...},

function(){...mouseout...}

);

更简化: 如果能够将两个函数统一为一个函数，则可省略第二个函数，表示无论over还是out都执行相同操作。

2. 动画:

1. 简单动画: 3种固定效果:

1. 显示/隐藏: $(...).show/hide/toggle();

问题:如果不给任何参数: display:none， 瞬间显示隐藏

解决: 指定参数: speed: 1. 三档("slow","normal","fast")

2. 上滑/下滑: $(...).slideUp/slideDown/slideToggle();

3. 淡入/淡出: $(...).fadeIn/fadeOut/fadeToggle();

缺点: 效果写死在库函数中，无法维护！

解决: class+transition

所有动画API的最后一个参数，都可指定回调函数

$(...).show(speed, callback)

callback: 在动画结束后自动执行，this->当前元素

2. 万能动画:

$(...).animate({//目标样式

css属性:值, css属性:值, ...

},speed,callback);

不足: 只能对单个数值的属性提供渐变

并发和排队:

并发: 所有属性同时改变

1. 放在同一个animate中的多个css属性

2. 不同元素上调用的animate是并发

排队: 每个属性先后变化

连续链式调用的多个animate的css属性是排队执行

练习:

$(...).css("属性",function(i,val){//自动在每个元素上执行

i //当前元素的下标

val //当前元素的指定属性值

this //当前DOM元素...

return 要修改给当前元素的属性的值

})

停止动画: $(...).stop();

问题: 只能停止队列中正在执行的一个动画

后续动画依然会执行

解决: $(...).stop(true); //停止当前动画，清空队列

练习: :animated 选择器 可选择正在播放动画的元素

DOM/day04素材/5\_3\_adv.html

3. 类数组对象的操作:

类数组对象不是数组类型，无法使用数组的API

jQuery为类数组对象添加了一组遍历和查找的API

$(...).each(function(i){...this//当前正在遍历的DOM元素...})

类似于数组的forEach，对每个元素执行相同操作

var i=$(elem).index("selector")

在所有满足右侧selector条件的元素中，找elem的位置