正课:

1. 事件:

页面加载后执行

鼠标事件

模拟触发

2. 动画:

简单动画， 万能动画

3. 类数组对象操作

4. 扩展自定义jQuery函数

1. 事件:

鄙视: onload vs $(document).ready()

加载后执行: 2种:

1. onload 当所有页面内容(html,css,js,图片)加载完自动执行——后

何时: 依赖css和图片的操作，必须放在onload中

如何: window.onload=function(){...}

注意：一个页面只能执行一次onload

2. DOMContentLoaded 仅DOM内容(html和js)加载完成——提前

何时: 不依赖于css和图片的操作，可放在DOMContentLoaded中提前执行，可让用户提前使用

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

可简写为: $().ready(function(){ ... ... })

还可简写: $(function(){ ... ... })

ES6: $(()=>{ ... ... })

其实: 放在body结尾的script中的代码，默认就是DOM内容加载后执行.

注意：一个页面可以增加多个ready

鼠标事件:

1. mouseover mouseout

进出子元素，同样会触发父元素上的事件处理函数

2. mouseenter mouseleave

进出子元素，不会反复触发父元素上的事件处理函数

3. hover: 同时绑定进入和移出事件

$(...).hover(fn1,fn2)

fn1绑定给进入事件，fn2绑定给移出事件

其实简化的是enter和leave，进出子元素，不再触发父元素的事件。

其实可以更简化:

如果fn1和fn2可统一为同一个函数，就只需要一个函数即可

模拟触发: 触发另一个元素上的事件处理函数

$(...).trigger("事件名")

其实可以更简单: $(...).事件名();

2. 动画:

简单动画: 3大效果:

1. 显示隐藏: $(...).show/hide/toggle(/\*ms\*/)

默认: 使用display瞬间显示/隐藏

如果添加持续时间，才有动画效果

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

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

最大的通用问题: 效果代码是在jQuery库中写死的。

解决: 用transition代替简单动画

只有show/hide()最常用，用来代替.css("display",..)

动画播放后，自动执行:

其实每个动画函数，都有一个回调函数参数。

动画播放结束后，会自动回调传入的函数对象

$(...).动画函数(ms, fn)

问题: callback hell 回调地狱: 因为多个函数，必须保证依次先后执行的顺序，导致回调嵌套很深。

解决: 排队

万能动画:

$(...).animate({

要修改的css属性，及其目标值

},ms,fn);

排队和并发:

排队: 后一个动画，必须等待前一个动画播放完，才能开始播放

如何: 用链式操作，用同一个元素调用多个动画函数，默认排队执行.

原理: jQuery为每个元素，都定义了一个动画队列。每次调用动画函数，都不会立刻执行动画。而是先加入动画队列等待。队列中的动画函数，默认排队依次执行。

并发: 一个元素的多个属性，同时在动画中修改

如何: 在一个animate中，同时修改多个css属性。

停止动画:

$(...).stop() 默认只停止当前动画，队列中剩余动画会继续执行！

$(...).stop(true) 停止动画，并清空队列！

延迟执行: $(...).delay(ms).动画函数()

选择器: :animated 选择正在播放动画的元素

3. 类数组对象操作:

问题: jQuery对象是类数组对象，无法使用数组的API: forEach, indexOf ... ...

解决: jQuery中提供了操作类数组对象的简化版API

包括:

$(...).get(i) 获得jq对象内的i位置的DOM元素

$(...).size() 等效于length，获得jq对象中的DOM元素个数

$(...).each(fn) 对jq对象中每个DOM元素执行相同的操作, 类似于数组的forEach

如何: $(...).each( (i,elem)=>{

i //当前位置

elem //当前DOM元素

})

$(...).index(DOM/$jq) 类似于数组的indexOf

4. 扩展自定义jQuery函数

何时: 如果jQuery的函数不够用

如何: 将函数添加到jQuery.fn中

强调:

1. 添加函数之前，必须先验证已经提前引入了jQuery函数库。

2. 使用自定义函数前，必须在网页中引入自定义的库.js文件

3. 必须先引入jquery.js，再引入自定义函数库