复习：

jQuery是一个JS函数库，100+函数，用于简化DOM操作，屏蔽浏览器兼容性。函数分为四类：

(1)DOM操作

(2)事件处理

(3)动画

(4)AJAX

jQuery提供的DOM操作函数：

(1)查找元素——选择器 $('选择器')

基本选择器：#id .class div \* s1,s2

层级选择器：

p c p > c

prev + next prev ~ nextAll

属性选择器

[attr] [attr="value"]

表单元素选择器

:text :password :radio :checkbox

:submit :reset :button :image

:file :hidden :input

:checked :selected :disalbed :enabled

基本过滤选择器

:first :last :eq(i) :gt(i) :lt(i) :odd :even

:not $('div:not(.box)')

子元素过滤选择器

:first-child :last-child :nth-child(i/odd/even/3n+2) :only-child

内容过滤选择器

:contains(txt) :has(s1)

:empty :parent

可见性过滤选择器

:hidden :visible

(2)操作元素的属性

(3)操作元素的内容

(4)操作元素的值

(5)操作元素的样式

(6)遍历元素

(7)添加元素

(8)删除元素

(9)替换元素

(10)克隆元素

今日目标：

(1)操作元素属性、值、内容、样式——重点

(2)增删改元素——重点

(3)遍历元素——重点

1.操作元素的属性

DOM：

div.setAttribute('title', 'xxx');

div.getAttribute('title')

jQuery：

$('div').attr('title', 'xxx'); //设置

var v = $('div').attr('title'); //读取

练习：鼠标悬停在小图上，则下方显示出对应的大图

2.操作元素的内容

DOM:

div.innerHTML = 'xxx';

var h = div.innerHTML;

div.innerText = 'xxx';

var t = div.innerText;

jQuery：

$('div').html('xxx'); //设置

var h = $('div').html(); //读取

$('div').text('xxx'); //设置

var t＝$('div').text(); //读取

练习：按钮，默认显示数字0；点击一次其上数字+1

3.操作元素的值

DOM：

input.value = 'Tom';

var v = input.value;

jQuery：

$('input').val( 'Tom' ); //设置

var v = $('input').val( ); //读取

练习：创建一个表单，包含用户名、密码、性别、所在城市，提交按钮被点击后，alert()出来用户所有的输入和选择内容，并清空所有的输入

4.操作元素的样式

DOM：

div.style.color = '#f00'

var c = div.style.color

div.className = 'box'

var n = div.className

<div class="box alert warning"></div>

jQuery：

$('div').css('color','#f00') //设置某个样式

var c = $('div').css('color') //读取某个样式

$('div').addClass('box') //添加class

$('div').removeClass('box') //删除class

var b = $('div').hasClass('box') //是否有class

$('div').toggleClass('box') //切换class（有则去除，无则添加）

提示：操作样式最好通过控制class来实现！添加/删除class名时不要添加.号！

练习：双态按钮：每个按钮在“按下”-“弹起”两个状态间切换 白底黑字<=>黑底白字

5.遍历元素

DOM：

//获得父元素

div.parentNode

//获取子元素

div.childNodes

//下一个兄弟

div.nextSibling

//上一个兄弟

div.previousSibling

jQuery：

$('div').parent() //获得父元素

$('div').children() //获得直接子元素

$('div').children('选择器') //获得直接子元素

$('div').find('选择器') //查找所有子元素

$('div').next() //下一个兄弟

$('div').nextAll() //后面所有兄弟

$('div').prev() //上一个兄弟

$('div').prevAll() //前面所有兄弟

$('div').siblings() //所有的兄弟

$('div').siblings('选择器') //所有的兄弟

练习：实现一个标签页的页头

<ul class="tabs">

<li><a href="#tc10">十元套餐</a></li>

<li><a href="#tc20">二十元套餐</a></li>

<li class="active"><a href="#tc30">三十元套餐</a></li>

</ul>

<div class="panes">

<div id="tc10">十元套餐详情: lorem </div>

<div id="tc20">二十元套餐详情: lorem </div>

<div class="active" id="tc30">三十元套餐详情: lorem </div>

</div>

6.添加新元素

DOM：

var p = document.createElement('p');

p.innerHTML = '';

p.onclick = function(){}

div.appendChild(p);

jQuery——八个方法：

var p = $('<p>AA<a>BB</a>CC</p>'); //创建jQuery对象，其中封装着新的DOM元素

p.click(function(){})

$('div').append( p );

-------------------------------

$('div').append('<p>AA<a>BB</a>CC</p>');

$('div').prepend('<p>AA<a>BB</a>CC</p>');

练习：点击按钮添加新的彩色div块

&times;

6.删除元素

DOM：

div.removeChild( p );

p.parentNode.removeChild( p );

jQuery：

$('p').remove( ) //删除当前选定元素

$('div').empty( ) //清空选定元素的中的内容

<=>$('div').html('')

7.替换元素

DOM：

parent.replaceChild(newChild, oldChild)

jQuery：

$('new').replaceAll( '.old' ) //返回新的元素

$('old').replaceWith( '.new' ) //返回被替换的元素

练习：选择英雄

8.克隆元素

DOM：

var copy = div.cloneNode() //浅克隆

var copy = div.cloneNode(true) //深克隆

jQuery：

var copy = $('div').clone() //深克隆，不克隆事件监听函数

var copy = $('div').clone(true) //深克隆，并克隆事件监听函数

练习：改进“英雄选择”——点击某个英雄飞机后，上方选择区域出现一个当前飞机的一个副本

9.jQuery中事件处理

面试题：jQuery中多少个事件绑定方法？各自区别？

历史上，先后出现了如下的事件绑定方法：

(1)bind() & unbind()

用法： $('.btn').bind('click', function(){})

$('.btn').unbind('click')

练习：创建一个按钮：“退出登录”，被点击一次后，显示文字“请稍候...”，此后再也不能被点击

(2)one()

用法： $('.btn').one('click', function(){})

说明： 监听函数只能执行一次，此方法相当于bind()中调用了unbind()

(3)bind函数的简化版本

$('.btn').bind('click', fn) => $('.btn').click(fn)

$('.btn').bind('mouseover', fn) => $('.btn').mouseover(fn)

$('.btn').bind('keyup', fn) => $('.btn').keyup(fn)

....

注意：这样的简化函数只有23个！

上述三个函数共同的缺陷：只能对当前DOM树上已经存在的元素绑定监听函数！后添加的元素无法绑定！

(4)delegate() & undelegate()

用法：$('.parent').delegate('.btn', 'click', function(){})

说明：每个目标元素自己不处理事件，利用“冒泡机制”，传递给父元素来处理。

使用父元素代理子元素的特定事件好处：

1)可以对当前尚未存在的元素生效！

2)可以减少监听函数的数量

this

document.body.childNodes[3].firstChild.nextSibling

document.getElementById()

document.getElementsByClassName()