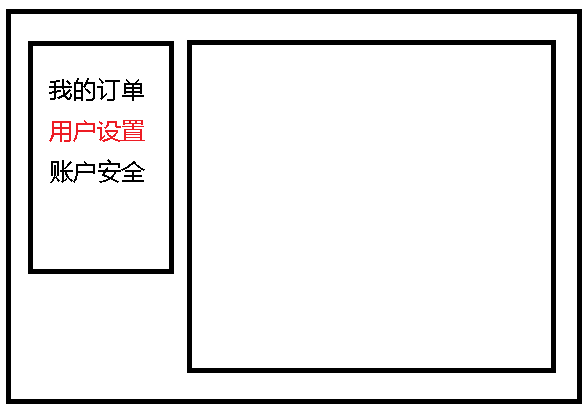
复习：

jQuery DOM操作练习：



1. jQuery中的事件处理相关函数——面试题

jQuery历史上先后出现的几个事件处理函数：

**(1)$(..).bind('类型', fn)**

$(..).unbind('类型')

$(..).one('类型', fn)

(2)bind()函数的简化版本

**$(..).click(fn)**

$(..).mousemove(fn)

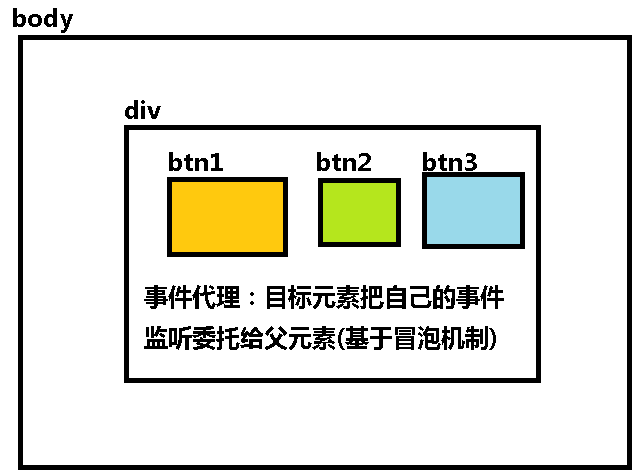
$(..).keyup(fn)

...

提示：上述函数的共同缺陷——只能为当前已有的DOM元素添加事件监听

**(3)$('parent').delegate('子元素选择器', '事件类型', fn)**

$('parent').undelegate('子元素选择器', '事件类型')



事件代理与事件源对象的事件绑定区别：

**事件源对象事件绑定**：每个源对象都必需一个监听函数。对后添加的元素无效。

**事件代理**：只需要给某个父元素绑定一个监听函数即可。对后添加的元素有效。

(4)$('target').live('事件类型', fn)——已过时

$('target').die('事件类型')——已过时

这两个形式与bind()相似，但底层是监听函数绑定在document上——事件代理，对后添加的元素也有效。

(5)**$('target').on('事件类型', fn)** —— 监听函数绑定在目标对象身上，等价于 $('target').bind( '', fn )

**$('parent').on('子元素选择器', '事件类型', fn)** —— 事件代理，等价于 $('parent').delegate('', '', fn)

$().off()

$('.btn').click(fn);

$('.btn').on('click', fn);

$('.parent').on( 'click','button', fn)

$('.parent').delegate( 'click','button', fn)

1. jQuery中的一个特殊事件

页面加载完成事件，原生DOM中有两种表示“页面加载完成”事件：

**window.onload（Load）：**页面加载完成，包括HTML、CSS、图片、Flash....

**window.ondomcontentloaded（DOMContentLoaded）**：页面内容加载完成，指DOM相关内容，如HTML、Script，不包括CSS、图片、Flash....，一般情况下，DOMContentLoaded事件会先与Load事件发生。

面试题： **$(document).ready(fn) 与 window.onload = fn 的区别**？

(1)$(document).ready()底层监听的是DOMContentLoaded事件，发生事件要先于window.onload。

(2)$(document).ready()底层的实现使用的addEventListener('DOMContentLoaded', fn, false)方法，而window.onload = fn是直接赋值。前者可以先后绑定多个监听函数，后者只能最后一次赋值有效。

1. jQuery动画函数——隐藏&显示动画

show( ) 显示

hide( ) 隐藏

使用定时器，控制元素的width、height、opacity等属性的改变。

1. jQuery动画函数——高度动画

slideUp( )

slideDown( )

slideToggle( )

使用定时器，控制元素的height等属性的改变。

5.jQuery动画函数——透明度动画

fadeIn( )

fadeOut( )

fadeToggle( )

使用定时器，控制元素的opacity属性的改变。

1. jQuery万能动画函数——animate( )函数

animate()可以对数值类型(如大小/位置/透明度/边框宽度/字体大小等)的CSS样式执行定时器动画，默认不能对非数值型(背景图片/字体/display等)属性做动画。

使用方法：

$(..).animate( {}, duration, fn )

**动画的串行(一个属性的改变完毕后再执行另一个)：**

$(..).animate( { 属性1: 值1 }, 1000)

$(..).animate( { 属性2: 值2 }, 1000)

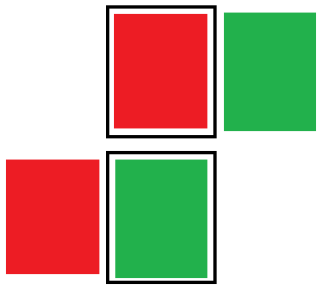
**动画的串行(多个属性的改变同时进行)：**

$(..).animate( { 属性1: 值1， 属性2: 值2 }, 1000)

练习：

(1)创建几个IMG元素，一旦被点击，则变大，同时变淡，最后消失

(2)创建两个撑满浏览器的区块，先显示出第一个，当它被点击时，慢慢的向左侧移走，同时第二个区块进入浏览器



作业：

点击第二个PAGE，还可以慢慢的返回第一个PAGE