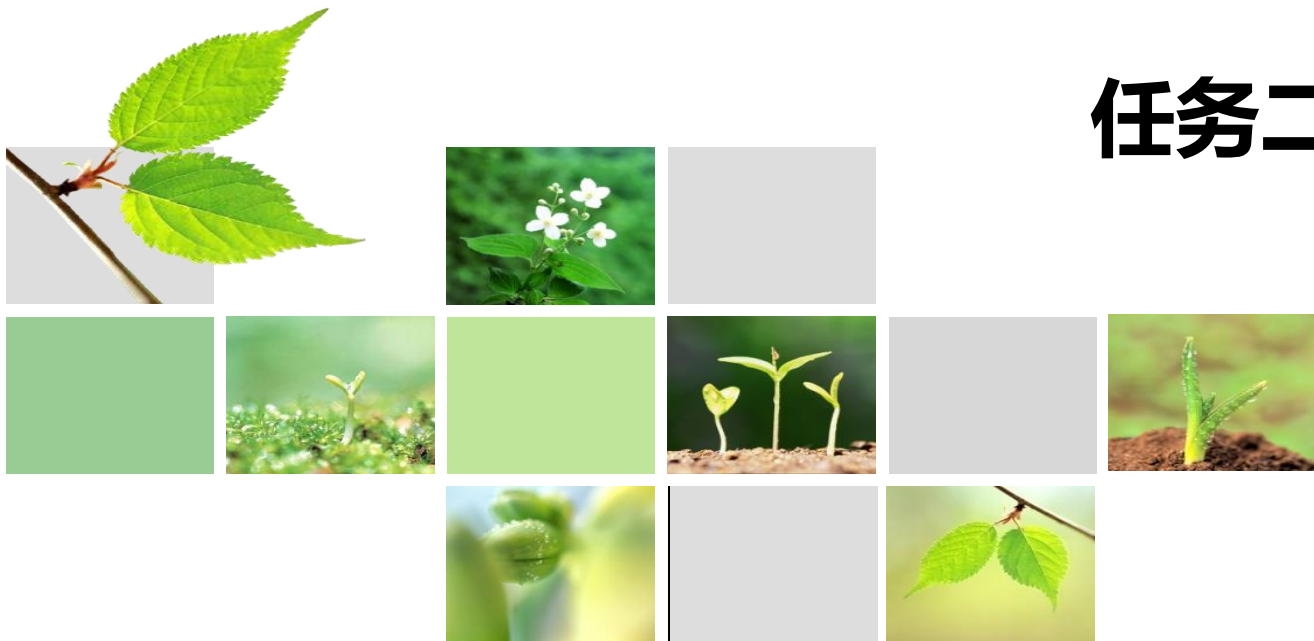


任务二：图片横向移动效果



图片横向移动效果



- 不仅美观，而且在节省页面空间，使有限的空间可以显示多个版面的信息给用户，且多个版面间的切换可以使用多种动画效果。例如，站酷网首页

(<http://www.zcool.com.cn/>) 中通过单击左右箭头使用

淡



图片横向移动效果



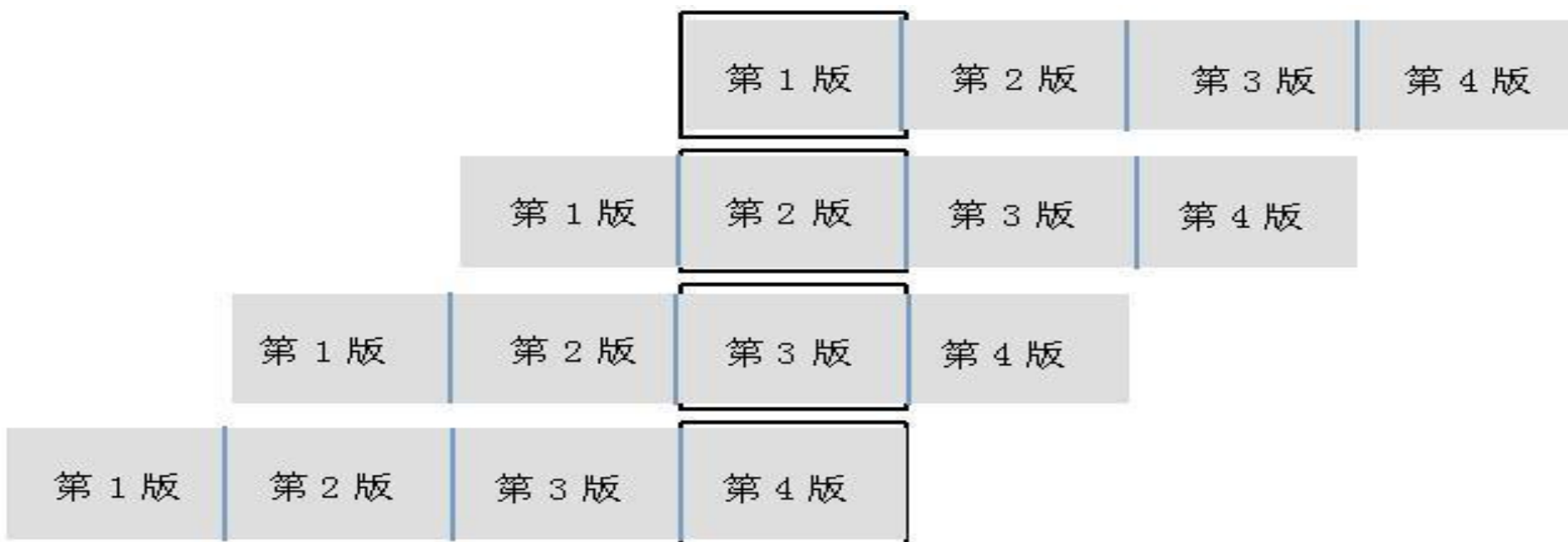
- 任务描述



任务分析



1.设计页面HTML结构，应用CSS样式。



任务分析



- 2.通过jQuery选择器获取“向右箭头”，为其绑定click事件。
- 3.获取要控制的“图片内容所在的DIV元素”（class属性为“v_content_list”）和“图片展示区域的DIV元素”（class属性为“v_content”）。
- 4.通过控制“图片内容所在的DIV元素”的left属性值来改变其水平位置，以一个版面的宽度为单位进行变化。

任务分析



5.当动画到最后一个版面或者第一个版面时，“图片内容所在的DIV元素”的left属性值的特殊设置。

6.左上角的箭头旁边的蓝色圆点需随动画一起切换，来标识当前所处的版面。结果HTML代码和CSS代码来实现。

7.当用户单击“向右箭头”或“向左箭头”过快时，动画累积，导致动画效果与用户操作不一致。因此，新动画需要在元素不处于动画状态时再添加。

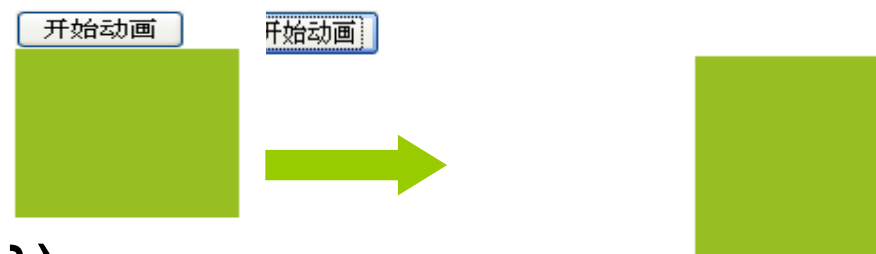
图片横向移动效果



相关知识点

- 自定义动画实现
 - **animate(params,speed,fun);**
params为高、宽等属性及值，**speed**为速度，**fun**为回调函数
- 简单动画
 - 单击“开始动画”按钮，把<div>元素移动到距离左边250px的位置，即**left**属性等于250像素为止。

```
$("#button").click(function(){  
    $("#div").animate({left:'250px'}, 3000);});
```



动画控制



- 累加、累减动画

```
$("#btn").click(function(){  
    $("#pos").animate({left:"+=20px"},fast);})
```

- 同时执行多个动画

```
$("#pos").animate({left:"+=20px",height:"+=20px"},fast);
```

- 顺序执行多个动画

```
$("#pos").animate({left:"+=20px"},fast).animate({  
height:"+=20px"},fast);
```

动画是顺序执行的。称为“动画队列”

动画控制



- 动画回调函数

在上例基础上给div加边框样式

在最后一个动画的回调函数里添加，可将css设置添加到动画队列。

- 停止动画`stop([clearQueue], [gotoEnd])`

`stop()`方法：停止当前正在进行的动画，以当前状态开始队列中接下来的动画。

`stop(true)`方法：清空动画队列。

`gotoEnd`：到达当前动画的末状态，而非队列的末状态。

- 判断元素是否处于动画状态

```
if(!$("#ele").is(":animated")){
```

```
//如果当前没有进行动画，则添加新动画.....}
```

常见问题



- 用户动作很快，导致动画效果与光标的动作不一致，怎么办？

```
$("#panel").hover(function(){  
    $(this).stop().animate({height:"150",width:"300"},200);  
  
    },function(){  
    $(this).stop().animate({height:"22",width:"60"},200);  
  
    }  
));
```

进一步思考



- 上面是一个动画，如果是下面的组合动画呢

```
$("#panel").hover(function(){  
    $(this).stop()  
        .animate({height:"150"},200)  
        .animate({width:"300"},300);;  
  
    },function(){  
        $(this).stop()  
            .animate({height:"22"},200)  
            .animate({width:"60"},300);;  
    }  
});
```

任务实现



- 选择器获取相关元素

```
$("#span.next").click(function(){ //“向  
    右箭头” 绑定单击事件  
  
    var $parent = $(this).parents("div.v_show"); //根据当前  
    元素获取祖先元素  
  
    var $v_show = $parent.find("div.v_content_list"); //获取 “图片  
    内容所在的DIV元素”  
  
    var $v_content = $parent.find("div.v_content"); //获取 “图片  
    展示区域的DIV元素”  
  
});
```

总结



- jQuery中的动画是实现良好交互效果的核心部分
- show()和hide()方法通过控制透明度的变化实现显示与隐藏
- slideUp()和slideDown()方法通过控制高度的变化实现显示与隐藏
- animate () 方法可自定义动画，用来定义css用于动画效果的属性