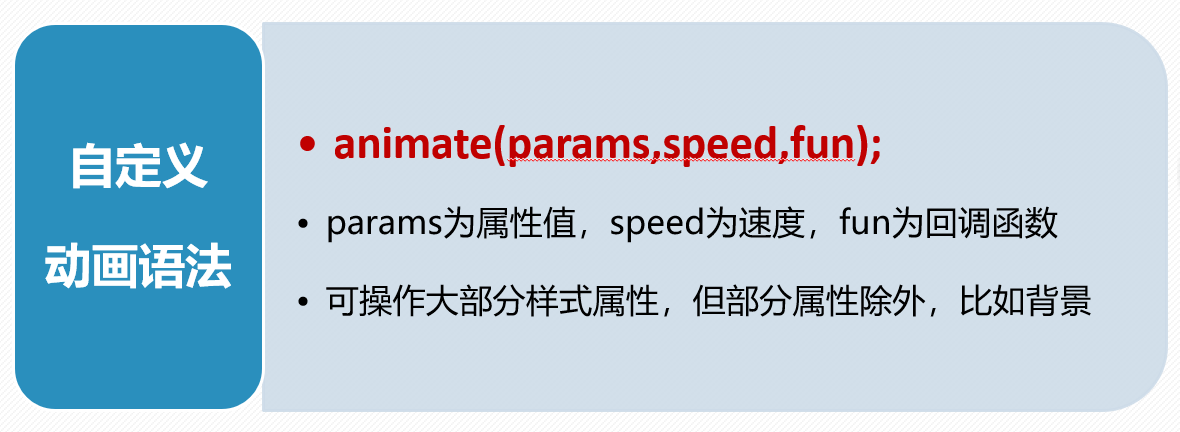
### 任务1：图片横向移动效果

#### 1.1自定义动画方法animate()

animate()方法的基本语法如下：

$(selector).animate({params},speed,callback);



##### 1.简单动画

单击“开始动画”,把<div>元素移至距左边250像素的位置，即left等于250px。

$("#btn1").click(function(){

$(".divStyle").animate({left:'250px’}, 2000);});

##### 2.累加动画

每单击一次“累加动画”按钮，把<div>元素移至距左边100像素的位置。

$(“#btn2”).click(function(){

$(“.divStyle”).animate({left:”+=100px”},”fast”);})

##### 3. 同时执行多个动画

每单击一次“执行多个动画”，<div>元素往左移动100px、高度增加50px。

$("#btn3").click(function(){

$(“.divStyle”).animate({left:”+=100px”,height:”+=50px”},”fast”);})

##### 4. 顺序执行多个动画

每单击一次“动画队列”，<div>元素先往左移动100px，后高度增加50px。

$(“#btn4”).click(function(){

$(“.divStyle”).animate({left:”+=100px”},”fast”)

.animate({height:”+=50px”},”

**动画顺序执行 称为“动画队列”**

#### 1.2动画回调函数

$("button").click(function(){

$("div").animate({left:'250px',top:'300px'})

.animate({opacity:'0.5'})

.animate({width:'150px'},'slow')

.css("border","5px solid orange");

});

**如果想在动画执行完毕后，为div添加边框，此时该代码达不到预期效果。出现这个问题的原因是css()方法并不是动画方法，**

**它不会被加入到动画队列进行排队，而是插队立即执行。如果要实现我们预期的效果，就必须使用回调函数让非动画方法实现排队。**

$("button").click(function(){

$("div").animate({left:'250px',top:'300px'})

.animate({opacity:'0.5'})

.animate({width:'150px'},'slow', function(){

$("div").css("border","5px solid orange");

});

});

#### 1.3停止元素的动画

网页中有时需要停止匹配元素正在进行的动画，这时要使用停止元素的动画方法stop()，它的语法结构为：

stop([clearQueue],[gotoEnd]);

参数clearQueue 和 gotoEnd 都为可选参数，为布尔值，即true或false，默认值都是false。clearQueue 代表是否要清空未执行完的动画队列，

gotoEnd代表是否直接将正在执行的那个动画跳转到末状态，而非动画队列中最后一个动画的末状态。

* stop(false,false):停止当前动画，并从目前的动画状态开始动画队列中的下一个动画
* stop(true,false): 停止所有动画，保持当前状态，瞬间停止
* stop(false,true): 停止当前动画，跳到当前动画的末状态，然后进入队列中的下一个动画
* stop(true,true): 停止所有动画，跳到当前动画的末状态

#### 1.4 判断元素是否处于动画状态

if(! $(elememt).is(":animated") ){ //判断元素是否处于动画状态

//如果当前没有进行动画，则添加新动画

}

#### 1.5延迟动画

delay()方法用于将队列中的函数或动画延时执行。它既可以推迟动画队列中函数的执行，也可以用于自定义队列。语法结构如下：

delay(duration, [queueName])