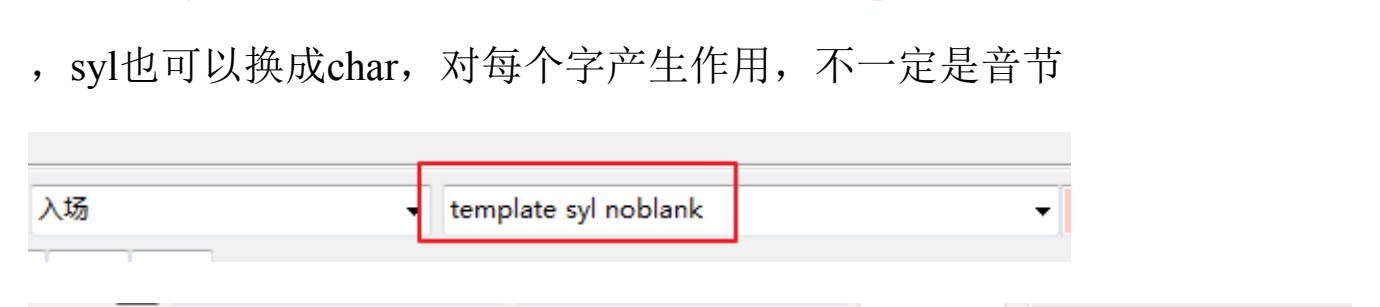
1.今天讨论一下关于aegisub中的自动化，这篇帖子适合新手们参考，但需要新手对ASS特效标签有一些了解，懂得如何去使用，以及auto中提供的内联变量，如果不懂可以参考这两位前辈的帖子，这篇帖子能教会新手一步一步的编写自己第一个特效脚本，主要要教会新手如何设置时间和模拟tcax中的基础特效

Template 特效标签

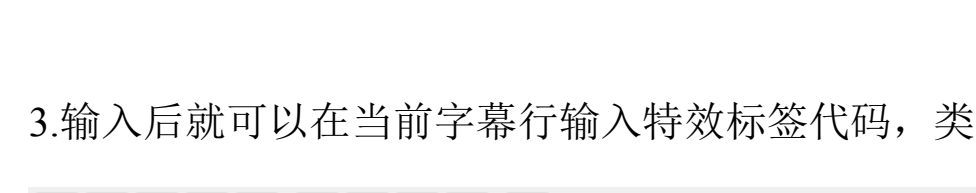
Template 内置变量

2.打开aegisub，在字幕时间控制面板，输入template syl noblank，表示按照音节来分割，忽略空行，syl也可以换成char，对每个字产生作用，不一定是音节


入场 | template syl noblank

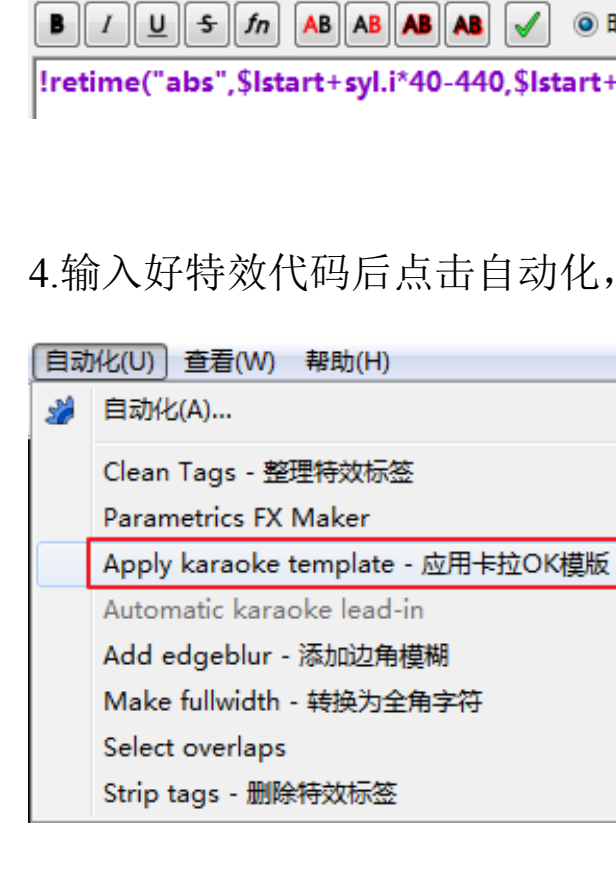
开始时间	结束时间	字/秒	样式	说话人	特效	文本
0:00:00.00	0:00:00.00		and	入场	template syl noblank	!retime("abs",$lstart+syl.i*40-440,$lstart+syl.i*40-40)!
0:00:00.00	0:00:00.00		and			

3.输入后就可以在当前字幕行输入特效标签代码，类似于下面

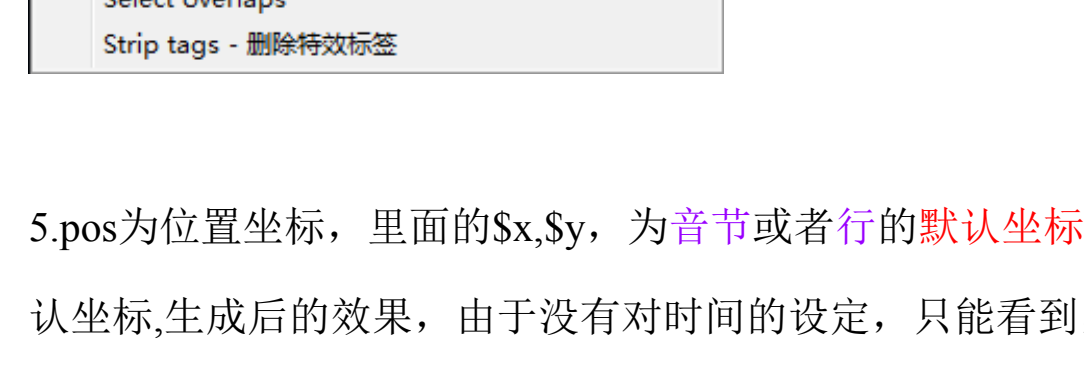


```
!retime("abs",$lstart+syl.i*40-440,$lstart+syl.i*40-40)!{\pos($x,$y)}
```

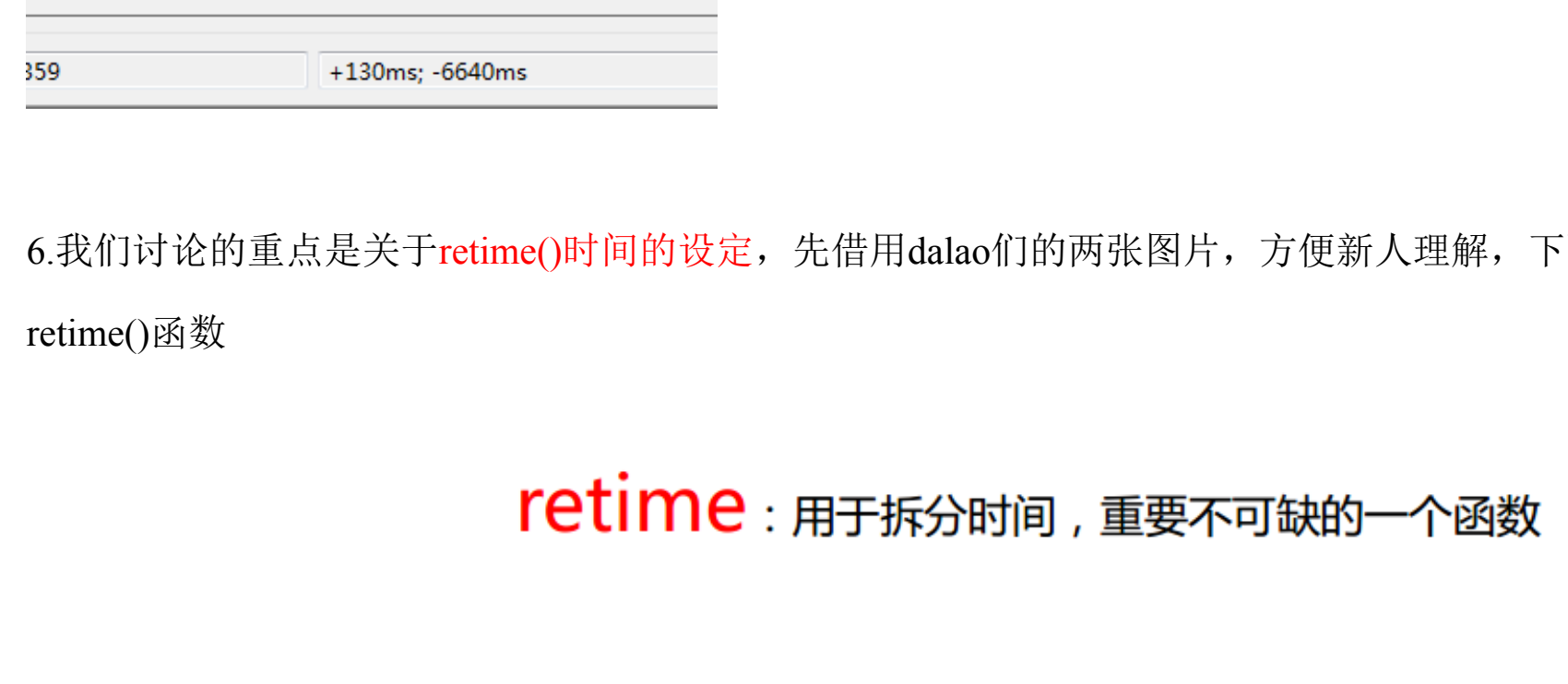
4.输入好特效代码后点击自动化，应用到卡拉OK模板


自动化(U)
查看(W)
帮助(H)
自动化(A)...
Clean Tags - 整理特效标签
Parametrics FX Maker
Apply karaoke template - 应用卡拉OK模板
Automatic karaoke lead-in
Add edgeblur - 添加边角模糊
Make fullwidth - 转换为全角字符
Select overlaps
Strip tags - 删除特效标签

5.pos为位置坐标，里面的$x,$y，为音节或者行的默认坐标，也可以使用$scenter,$smiddle,音节的默认坐标,生成后的效果，由于没有对时间的设定，只能看到，一行字幕一闪而过



6.我们讨论的重点是关于retime()时间的设定，先借用dalao们的两张图片，方便新人理解，下面是retime()函数


retime：用于拆分时间，重要不可缺的一个函数

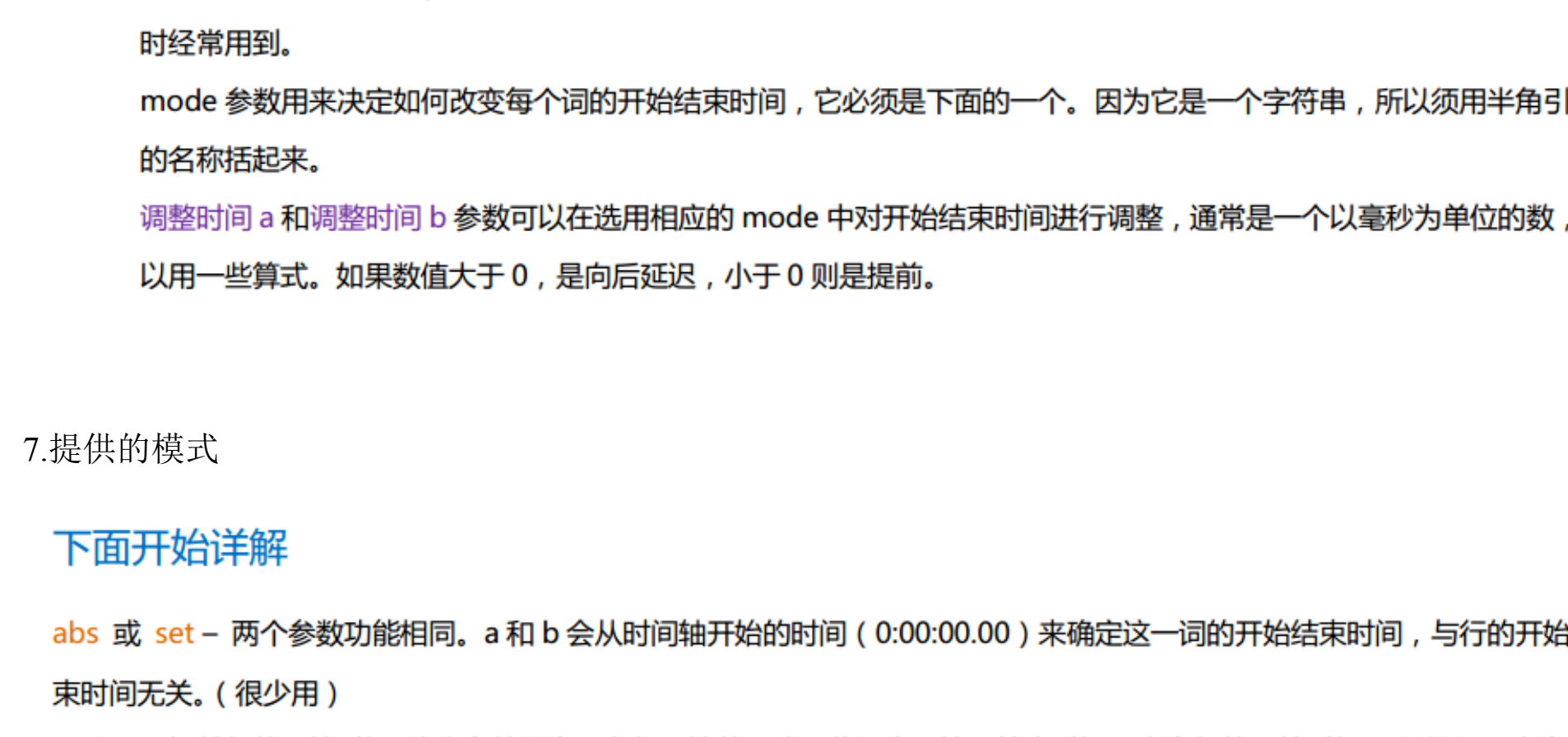
格式：retime(mode,调整时间 a,调整时间 b)

该函数通常在一个 template 或 code 中使用一次或多次。它可以以不同的方式来调整输出的每个词的开始结束时间，在做特效时经常用到。

mode 参数用来决定如何改变每个词的开始结束时间，它必须是下面的一个。因为它是一个字符串，所以须用半角引号将模式的名称括起来。

调整时间 a 和调整时间 b 参数可以在选用相应的 mode 中对开始结束时间进行调整，通常是一个以毫秒为单位的数，当然也可以用一些算式。如果数值大于 0，是向后延迟，小于 0 则是提前。

7.提供的模式


下面开始详解

abs 或 set – 两个参数功能相同。a 和 b 会从时间轴开始的时间（0:00:00.00）来确定这一词的开始结束时间，与行的开始结束时间无关。(很少用)

preline –提前行的开始时间 这个参数是为了在行开始前，让一些词先开始。结束时间固定为行的开始时间，所以 b 固定为 0，可以省略。

line – 使用每一行的开始结束时间。a 和 b 在行的开始结束时间基础上可以改变词的开始和结束时间。

start2syl – 是每个词的开始时间为行的开始时间，结束时间为每个词 K 的开始时间。

presyl – 与 preline 类似，只是结束时间换成了词 K 的开始时间。

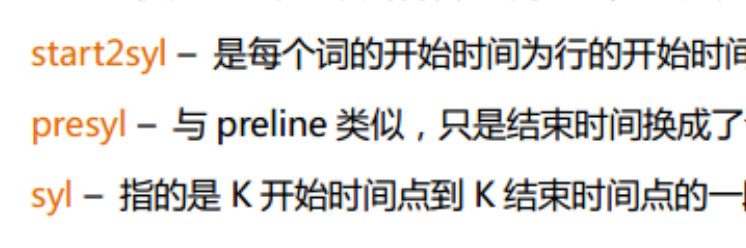
syl – 指的是 K 开始时间点到 K 结束时间点的一段时间。a 和 b 在词 K 的开始结束时间基础上可以改变词的开始和结束时间。

postsyl –每个词开始时间为 k 的结束时间，结束时间为 k 的结束时间加上调整时间 b。a 和 b 可调整时间。

syl2end – 每个词的开始时间为词 K 的结束时间，结束时间为行的结束时间，类似于 start2syl。

postline –每一行开始时间为该行的结束时间，结束时间为该行开始时间加上调整时间 b。a 和 b 可调整时间。

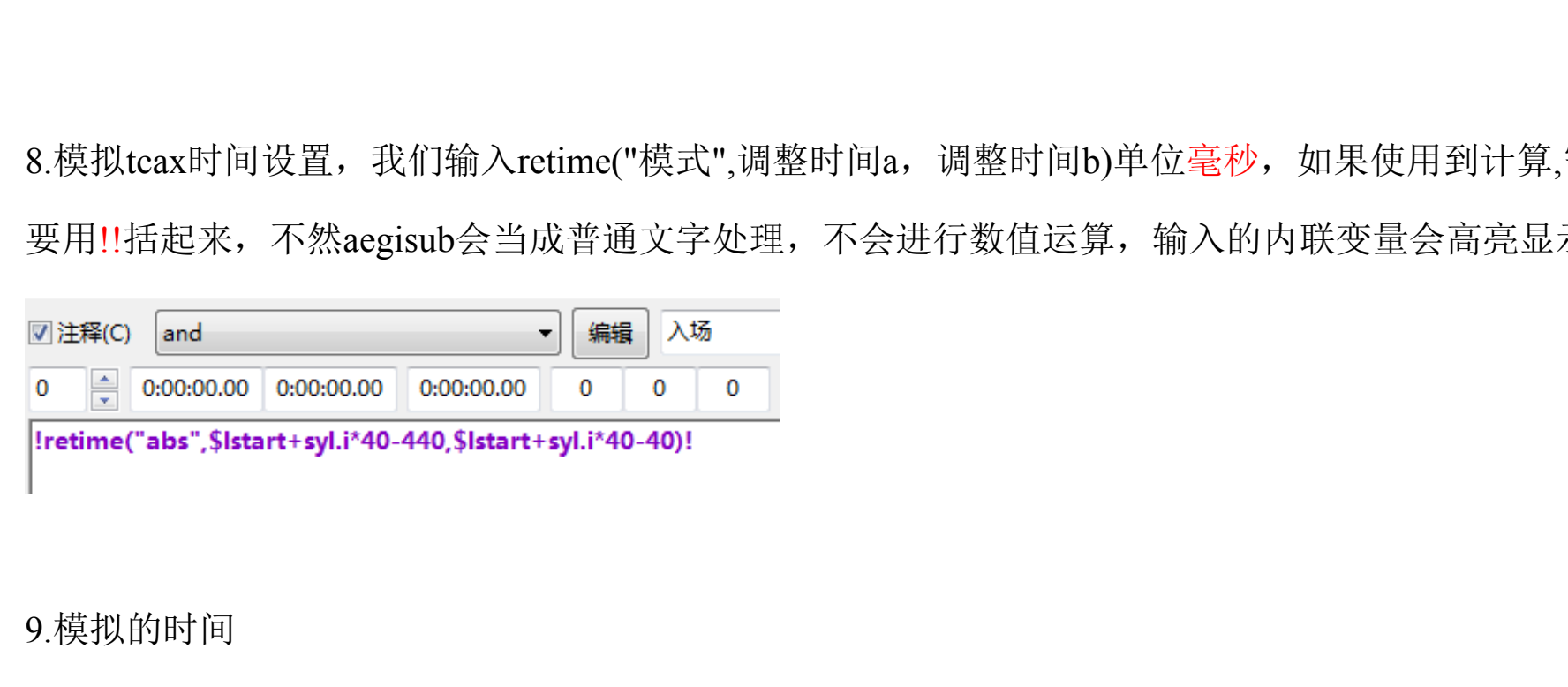
8.模拟tcax时间设置，我们输入retime("模式",调整时间a，调整时间b)单位毫秒，如果使用到计算,需要用!!括起来，不然aegisub会当成普通文字处理，不会进行数值运算，输入的内联变量会高亮显示


☑ 注释(C) and 编辑 入场

0 0:00:00.00 0:00:00.00 0:00:00.00 0 0 0

```
!retime("abs",$lstart+syl.i*40-440,$lstart+syl.i*40-40)!
```

9.模拟的时间



字/秒	样式	说话人	特效	文本
	and	入场	template syl noblank	!retime("abs",$lstart+syl.i*40-440,$lstart+syl.i*40-40)!
	and		tcax时间:	
	and			开始时间: _start + _j * _FD / 10 - _FD / 10 - duration / 10 $lstart+syl.i*1000/25-1000/25-1000/25*10/10
	and			结束时间: _start + _j * _FD / 10 - _FD / 10 $lstart+syl.i*1000/25-1000/25
	and			
	and	存在	template syl noblank	!retime("abs",$lstart+syl.i*40-40,$lend-($syln-syl.i-1)*40+40)!
	and		tcax时间:	
	and			开始时间: _start + _j * _FD / 10 - _FD / 10 $lstart+syl.i*1000/25-1000/25
	and			结束时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 $lend-($syln-syl.i-1)*1000/25+1000/25
	and			
	and	附加和表现	template syl noblank	!retime("syl",0,0)!
	and		tcax时间:	
	and			开始时间: _start + _elapk $sdur K值的持续时间
	and			结束时间: _start + _elapk + _k $sdur K值的持续时间
	and			
	and	退场	template syl noblank	!retime("abs",$lend-($syln-syl.i-1)*40+40,$lend-($syln-syl.i-1)*40+440)!
	and		tcax时间:	
	and			开始时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 $lend-($syln-syl.i-1)*1000/25+1000/25
	and			结束时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 + _FD $lend-($syln-syl.i-1)*1000/25+1000/25*10/10
	and			
	and	CMD		--------------------------------标准TCAX模板----------------------------------
	and			

10.详细的解释一下上面图片的参数，时间使用的tcax的时间，新人需要知道tcax里面的时间单位为10 ms，aegisub中的时间单位为1ms

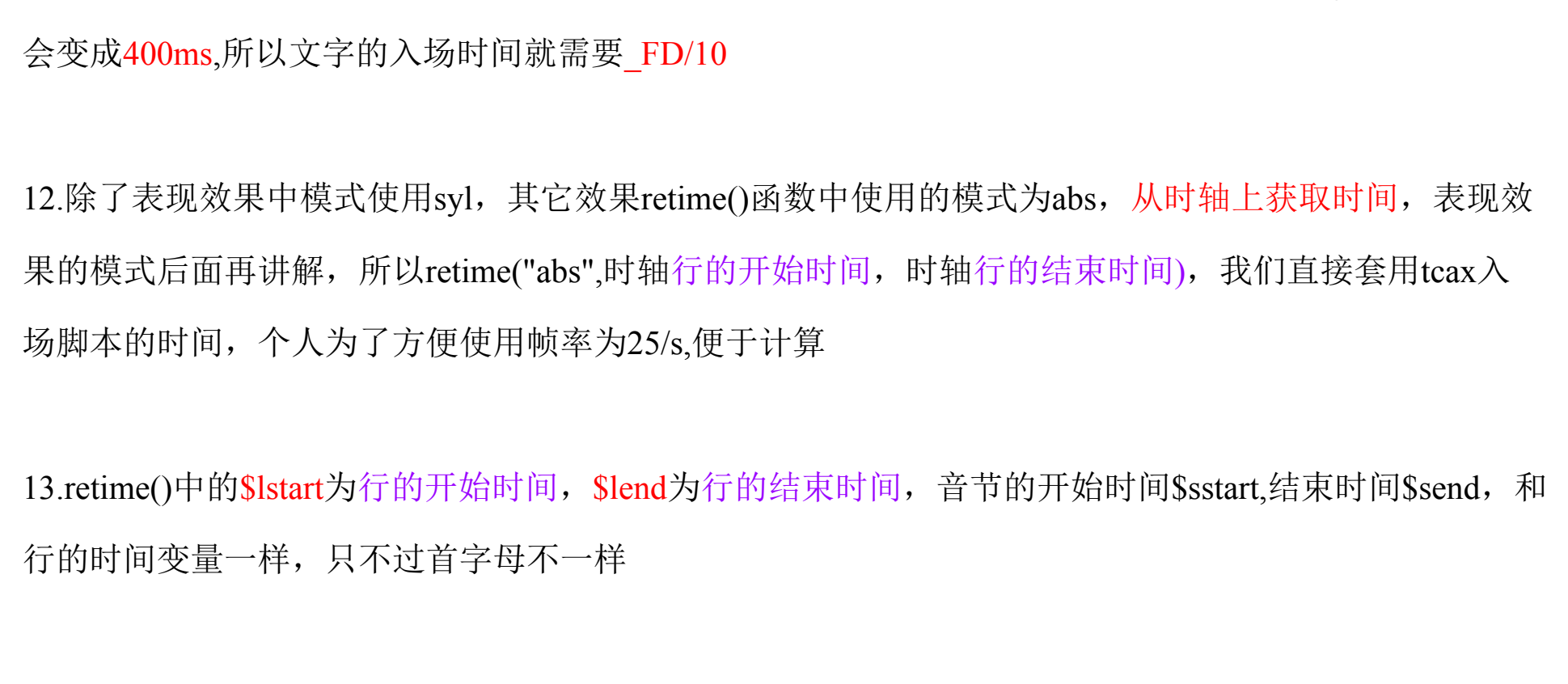
11.例如计算每一帧的持续时间，tcax中：_FD = 1000 / GetVal(val_FXFPS)，tcax里面tcc文件使用的视频帧率为23.975，为了方便个人使用的是25/s，计算的结果等于40ms每一帧，由于aegisub中单位为1ms，我们要字幕出现为每一帧出现一个，时间就应该为40ms，但tcax的时间单位为10ms，在aegisub里面就会变成400ms,所以文字的入场时间需要_FD/10

12.除了表现效果中模式使用syl，其它效果retime()函数中使用的模式为abs，从时轴上获取时间，表现效果的模式后面再讲解，所以retime("abs",时轴行的开始时间，时轴行的结束时间)，我们直接套用tcax入场脚本的时间，为了方便使用帧率为25/s,便于计算

13.retime()中的$lstart为行的开始时间，$lend为行的结束时间，音节的开始时间$sstart,结束时间$send，和行的时间变量一样，只不过首字母不一样

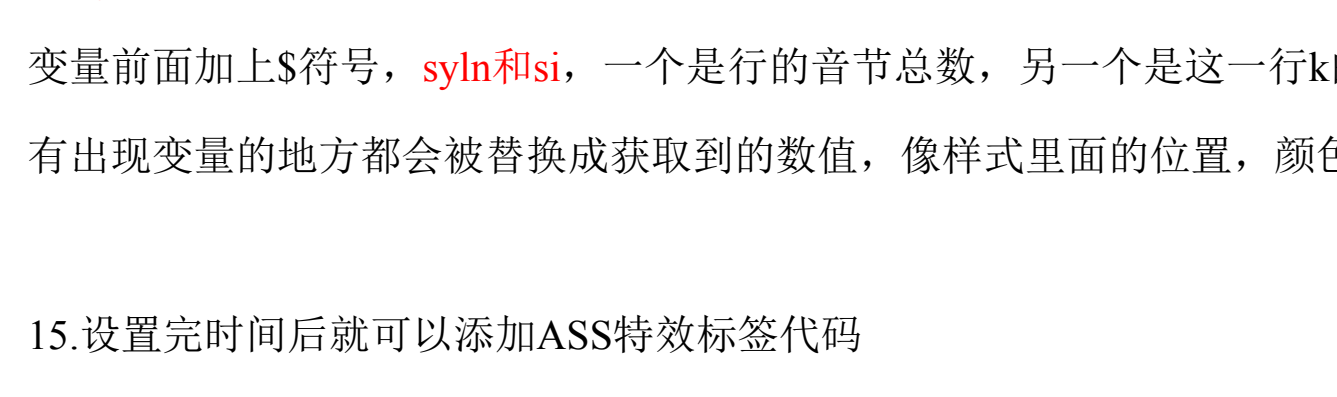
14.syl.i为音节计数器，处理第1个音节就为数值为1，第2个音节数值为2，以此类推，使用音节变量需要在变量前面加上$符号，syln和si，一个是行的音节总数，另一个是这一行k的数目，应用卡拉OK模板后，所有出现变量的地方都会被替换成获取到的数值，像样式里面的位置，颜色，透明度

15.设置完时间后就可以添加ASS特效标签代码



样式	说话人	特效	文本
and	入场	template syl noblank	!retime("abs",$lstart+syl.i*40-440,$lstart+syl.i*40-40)!{\move($center,!$middle-25!,$center,$middle,0,400)}
and		tcax时间:	
and			开始时间: _start + _j * _FD / 10 - _FD / 10 - duration / 10 $lstart+syl.i*1000/25-1000/25-1000/25*10/10
and			结束时间: _start + _j * _FD / 10 - _FD / 10 $lstart+syl.i*1000/25-1000/25
and			
and	存在	template syl noblank	!retime("abs",$lstart+syl.i*40-40,$lend-($syln-syl.i-1)*40+40)!{\pos($scenter,$middle)\3c!_G.ass_color(_G.HSV_to_RGB($si/$syln*390,0.9,0.8))!\bord2\blur2}
and		tcax时间:	
and			开始时间: _start + _j * _FD / 10 - _FD / 10 $lstart+syl.i*1000/25-1000/25
and			结束时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 $lend-($syln-syl.i-1)*1000/25+1000/25
and			
and	附加和表现	template syl noblank	!retime("syl",0,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-70,70)!)\t(0,$sdur,\fscx100\fscy100\frz0)\pos($scenter,$middle)}
and		tcax时间:	
and			开始时间: _start + _elapk $sdur K值的持续时间
and			结束时间: _start + _elapk + _k $sdur K值的持续时间
and			
and	退场	template syl noblank	!retime("abs",$lend-($syln-syl.i-1)*40+40,$lend-($syln-syl.i-1)*40+440)!{\move($scenter,$middle,$scenter,!$middle+25!,0,400)}
and		tcax时间:	
and			开始时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 $lend-($syln-syl.i-1)*1000/25+1000/25
and			结束时间: _end - (_n - _j - 1) * _FD / 10 + _FD / 10 + _FD $lend-($syln-syl.i-1)*1000/25+1000/25*10/10
and			
and	CMD		--------------------------------标准TCAX模板--------------------------------
and			

16.应用后的效果

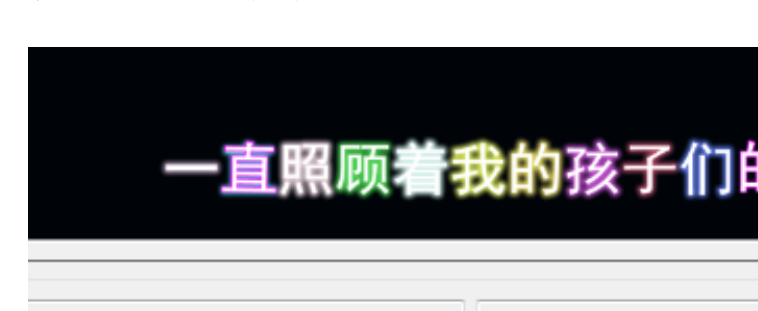


样式	说话人	特效	文本
and			
and	附加和表现	template syl noblank	!retime("syl",0,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-70,70)!)\t
and		tcax时间:	

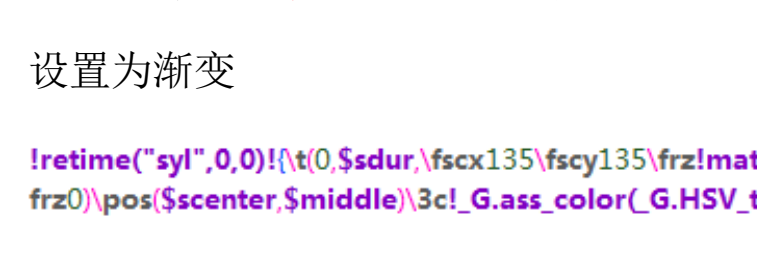
17.主要讲解一下关于!_G.ass_color(_G.HSV_to_RGB($si/$syln*390,0.9,0.8))!渐变，这两个_G.系列，一个作用是设定颜色，一个作用是HSV转成RGB格式，也就是转成&H00FF00& 这种，数值1 是色调，也就是【红橙黄绿青蓝紫】数值2是饱和度，也就是 这个颜色的【浓度】数值3 是亮度一般来讲 饱和度、亮度用1(满),可以使用小数

18.色调用角度度量，取值范围为0°~360°，从红色开始按逆时针方向计算，红色为0°，绿色为120°,蓝色为240°。它们的补色是:黄色为60°，青色为180°,品红为300°

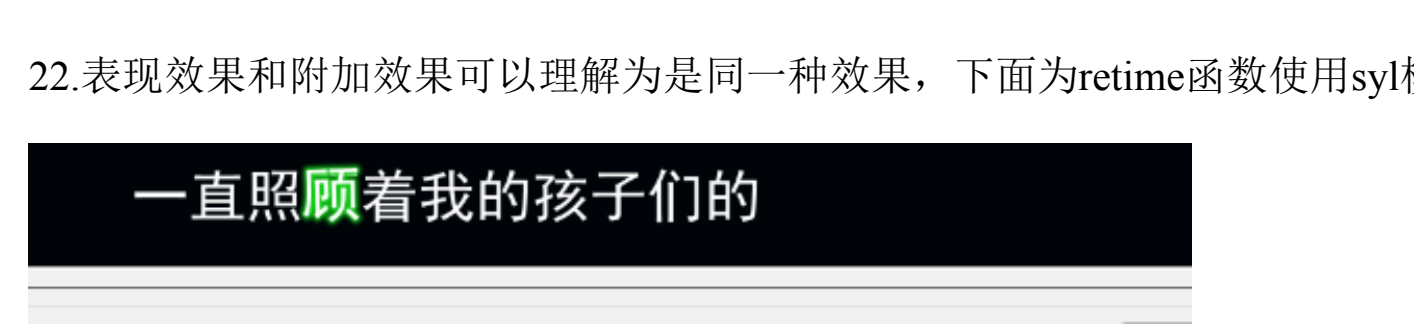
19.例如，一句歌里面，有10个音节，那么每一个音节的色调就是 1/10 *400 = 40，2/10 *400 = 80，3/10 *400 =120，4/10 *400 = 160，每个音节就会平分色调，就达到渐变色的效果了



20.也可以!_G.ass_color(math.random(50,255),math.random(50,255),math.random(50, 255))!，下面是效果，颜色偏淡，这个比较适用于整行文字，更换颜色效果，math.random()函数可以根据传入的数字从中随机生成一个数字



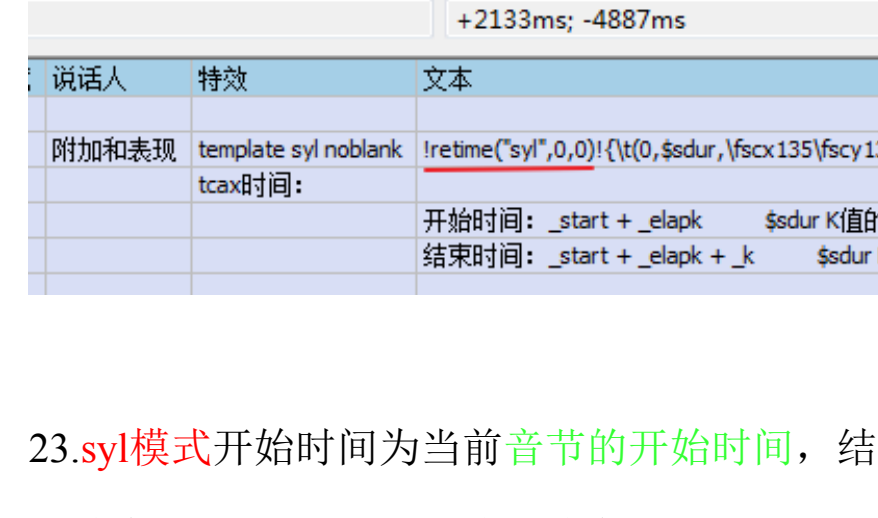
21.是时候讨论附加和表现效果的时间问题了，我们把in和main和out样式设置为白色，无边框，表现效果设置为渐变



```
!retime("syl",0,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-20,20)!)\t(0,$sdur,\fscx100\fscy100\
frz0)\pos($scenter,$middle)\3c!_G.ass_color(_G.HSV_to_RGB($si/$syln*390,0.9,0.8))!\bord2\blur2}
```

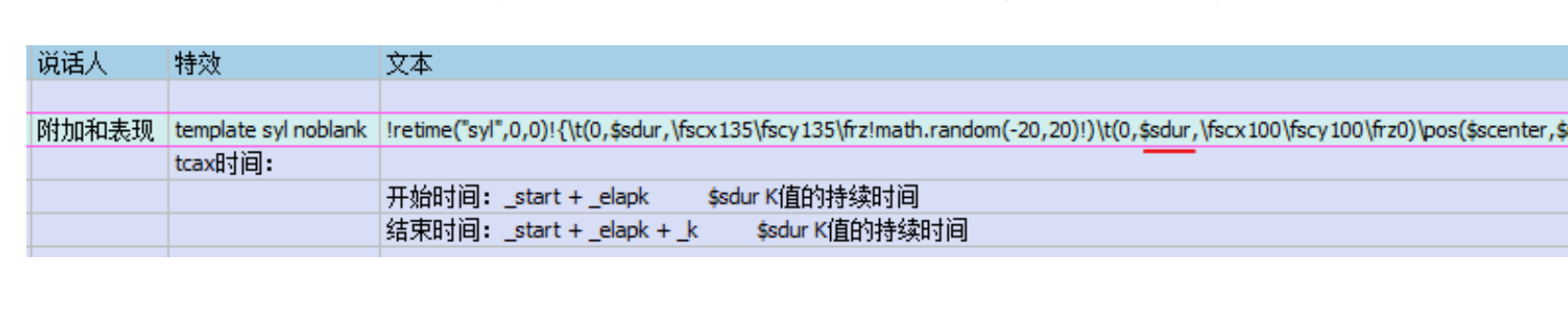
表现效果为渐变

22.表现效果和附加效果可以理解为是同一种效果，下面为retime函数使用syl模式的表现效果



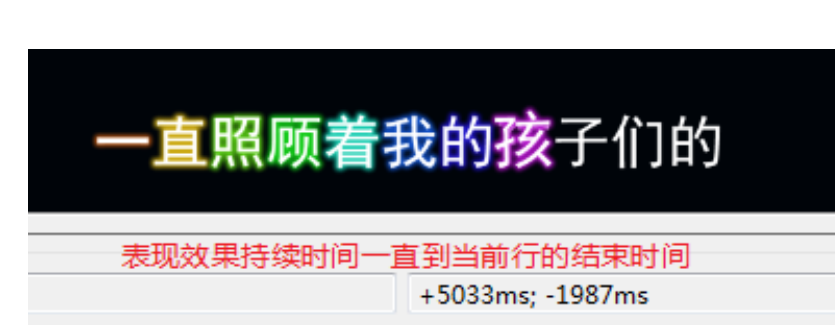
说话人	特效	文本
附加和表现	template syl noblank	!retime("syl",0,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-20,20)!)\t
	tcax时间:	
		开始时间: _start + _elapk $sdur K值的持续时间
		结束时间: _start + _elapk + _k $sdur K值的持续时间

23.syl模式开始时间为当前音节的开始时间，结束时间为音节的结束时间，相当于retime("syl",音节开始时间,音节结束时间)，里面的开始和结束时间可以写成0，意思是获取默认时间值，不修改，持续时间为$sdur，当前音节的持续时间，在这个时间内可以对音节添加各种效果，类似放大，缩小，旋转

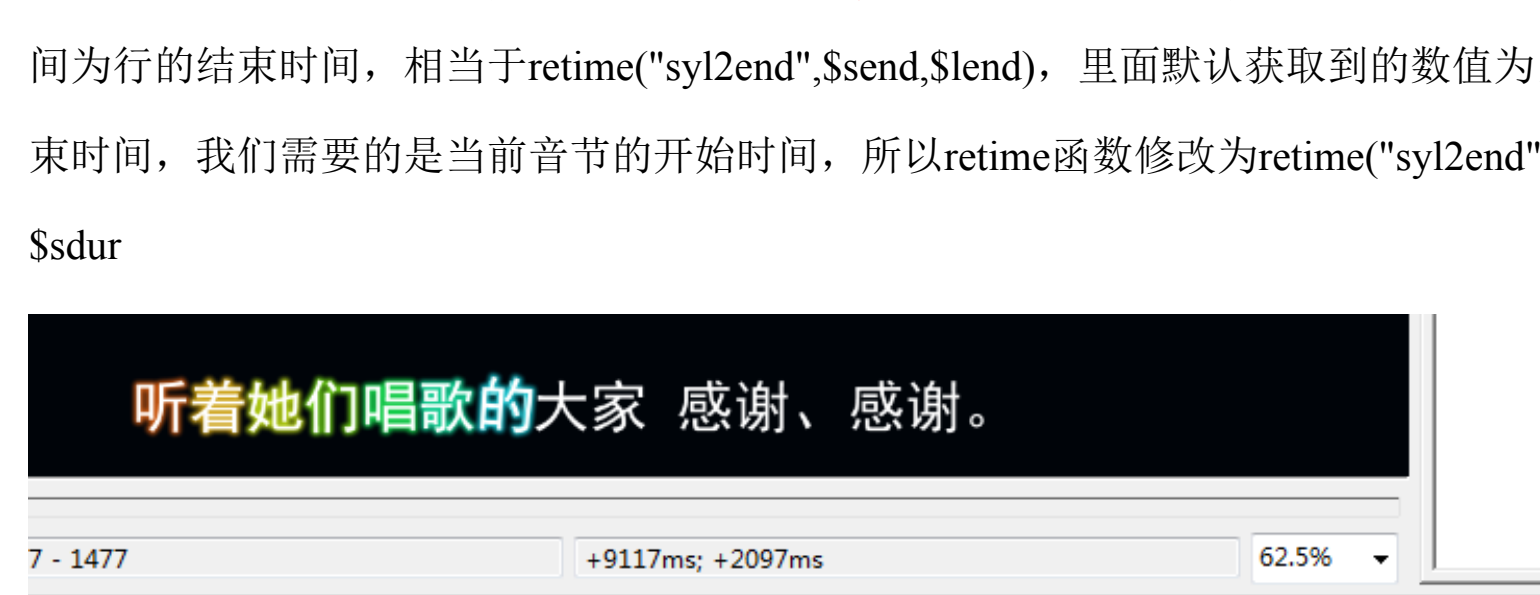


说话人	特效	文本
附加和表现	template syl noblank	!retime("syl",0,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-20,20)!)\t(0,$sdur,\fscx100\fscy100\frz0)\pos($scenter,$middle)
	tcax时间:	
		开始时间: _start + _elapk $sdur K值的持续时间
		结束时间: _start + _elapk + _k $sdur K值的持续时间

24.第2种，表现效果开始时间为音节(词)的开始时间，图片中的文字逐个渐变，持续时间一直到当前行的结束时间

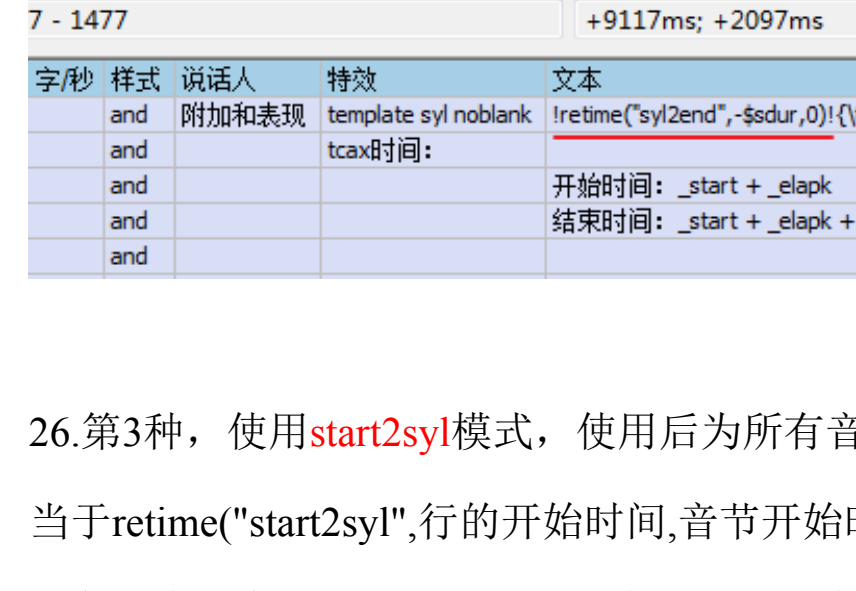


25.前面的表现效果retime()函数使用的模式为syl2end，表示每个词(音节)的开始时间为词K的结束时间，结束时间为行的结束时间，相当于retime("syl2end",$send,$lend)，里面默认获取到的数值为，k的结束时间，和行的结束时间，我们需要的是当前音节的开始时间，所以retime函数修改为retime("syl2end",-$sdur,0),所以$sstart=$send-$sdur



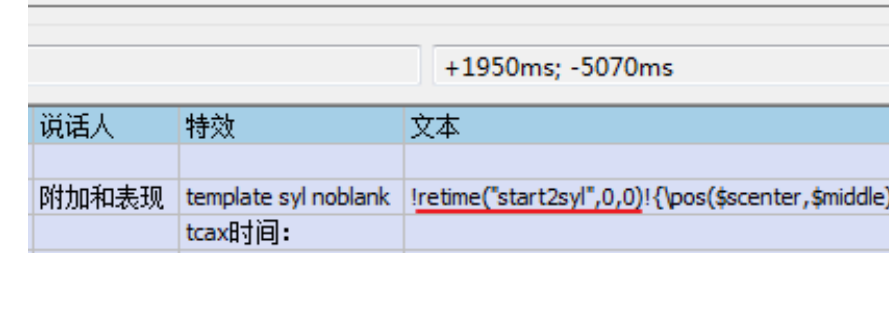
字/秒	样式	说话人	特效	文本
	and	附加和表现	template syl noblank	!retime("syl2end",-$sdur,0)!{\t(0,$sdur,\fscx135\fscy135\frz!math.random(-20,20)!)\t(0,$sdur,\fsc
	and		tcax时间:	
	and			开始时间: _start + _elapk $sdur K值的持续时间
	and			结束时间: _start + _elapk + _k $sdur K值的持续时间
	and			

26.第3种，使用start2syl模式，使用后为所有音节的开始时间为行的开始时间，结束时间为音节的开时间，相当于retime("start2syl",行的开始时间,音节开始时间)，使用的话修改为retime("start2syl",0,0)，表现效果为渐变颜色逐个文字消失，直到最后一个音节的结束，适和用在粒子特效表现时间



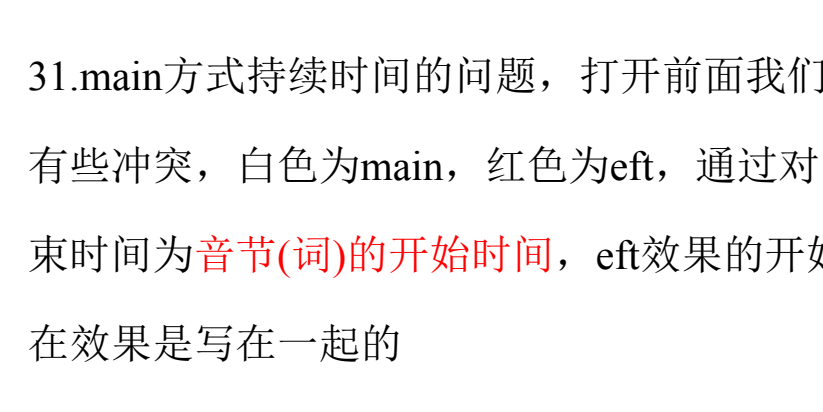
说话人	特效	文本
附加和表现	template syl noblank	!retime("start2syl",0,0)!{\pos($scenter,$middle)\3c!_G.ass_color(_G.HSV_

27.讲解了这么多，相信新手们基本可以用temp来模拟tcax的基本效果，下面我们通过观察特效表现效果间接编写脚本，我们试着不用tcax的时间公式，来编写第一个脚本，个人修改的模板



28.新手在使用其它的特效模板的时候，需要观察的是特效的入场，存在，表现，退场这四个效果，注意它们的各个效果的开始时间，和结束时间，可以记在记事本上，然后在考虑用哪种时间方式来实现里面的效果，tcax还是temp的时间方式

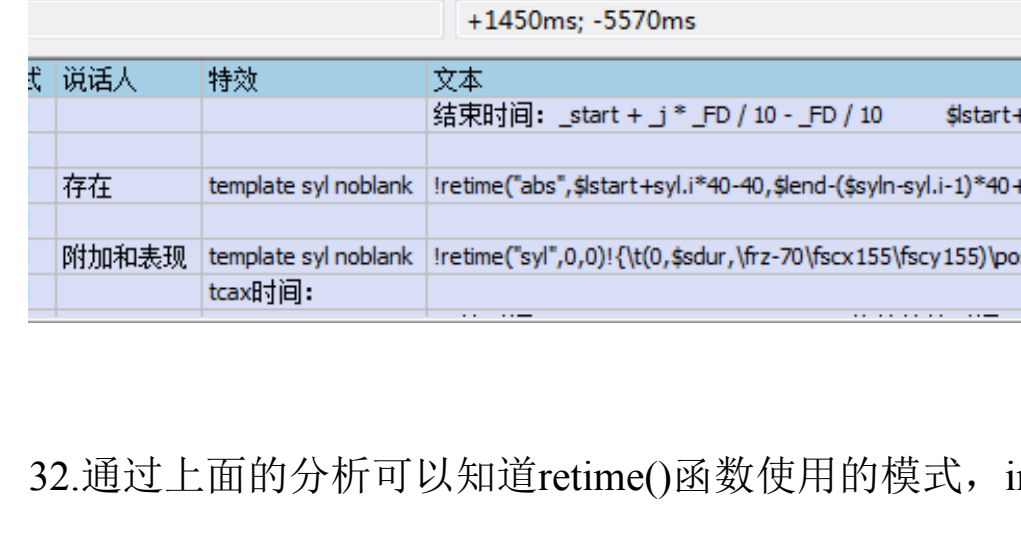
29.我们一个一个效果来实现，并讲解里面关于时间的问题，入场效果使用模式为start2syl，每个词的开始时间为行的开始时间，结束时间为音节K的开始时间



```
!retime("start2syl",-500+syl.i*40,0)!{\fry90\move($center,!$middle-25!,
$center,$middle,0,400)\1cFFFFFF\bord0\t(0,400,\fry0)}
```

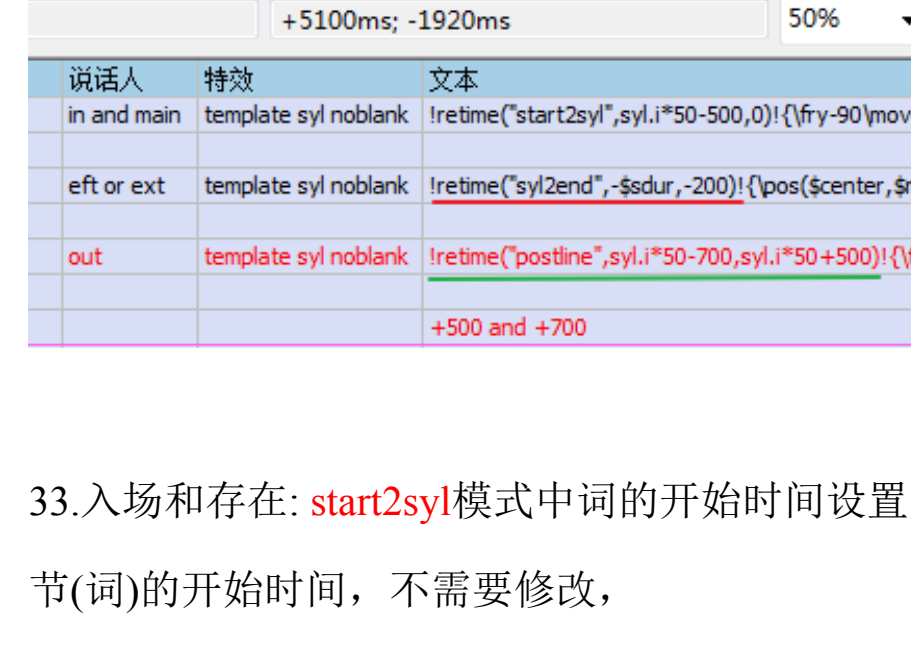
入场效果

30.我们设置的为-500+syl.i*40 , syl.i为计数器，第一个词的的开始时间提前为-500+1*40=-460,第一个音节的出现时间在行的开始时间的基础上提前460ms，第二个词的开始时间为-500+2*40=-420,提前了420ms，比第一个词延迟40ms，以此类推，后面每处理一个音节(词)，的开始时间都会比前一个音节的开始时间延迟40ms，这种时间差可以实现逐个文字下降的效果，模拟tcax里面的in_003，和in_008



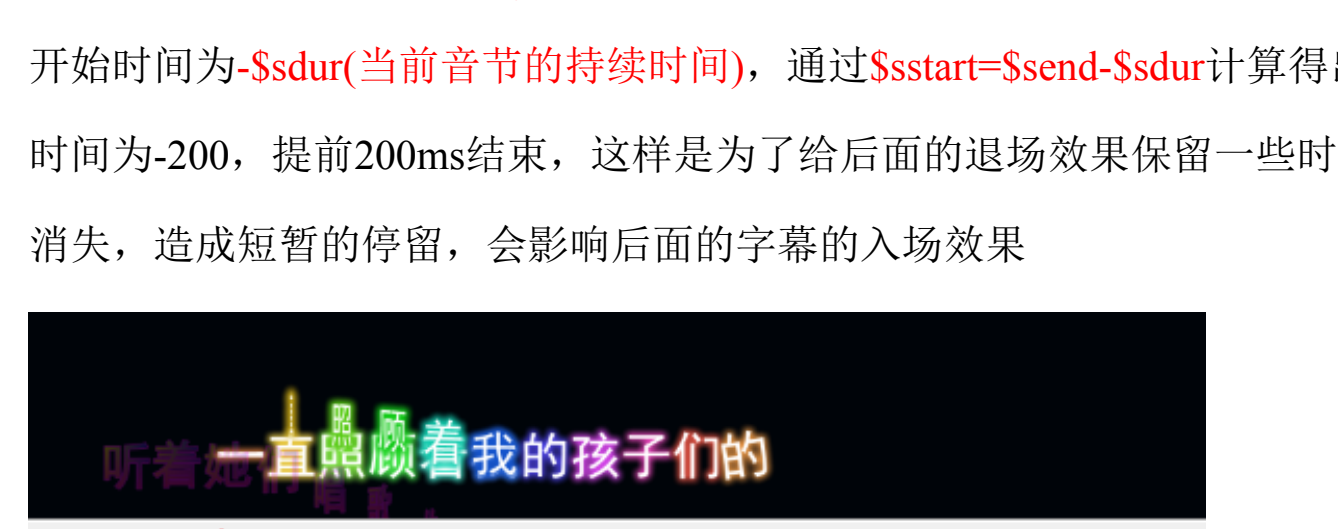
字/秒	样式	说话人	特效	文本
	and	入场	template syl noblank	!retime("start2syl",-500+syl.i*40,0)!{\fry90\move($center,
	and		tcax时间:	

31.main方式持续时间的问题，打开前面我们写好的tcax模板，应用效果后可以发现，表现效果，和存在方式，有些冲突，白色为main，红色为eft，通过对比前面第27的效果图，main方式的开始时间为行的开始时间，结束时间为音节(词)的开始时间，eft效果的开始时间为，音节的开始时间，结束时间为行的结束时间，入场和存在效果是写在一起的



说话人	特效	文本
		结束时间: _start + _j * _FD / 10 - _FD / 10 $lstart+syl.i
存在	template syl noblank	!retime("abs",$lstart+syl.i*40-40,$lend-($syln-syl.i-1)*40+40)!
附加和表现	template syl noblank	!retime("syl",0,0)!{\t(0,$sdur,\frz-70\fscx155\fscy155)\pos($s
	tcax时间:	

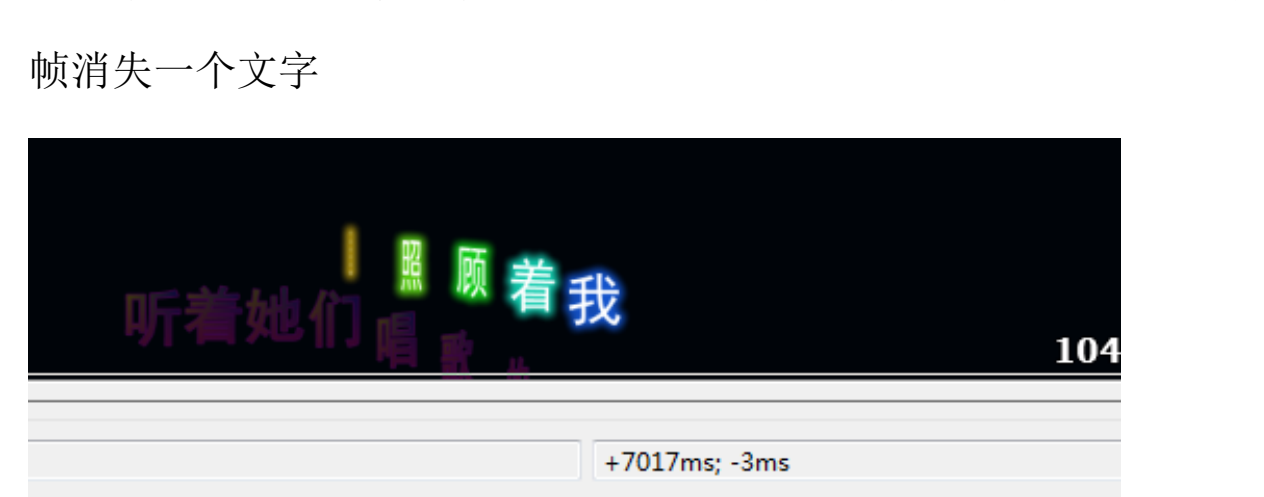
32.通过上面的分析可以知道retime()函数使用的模式，in和main使用start2syl，eft使用syl2end，out使用postline，最终编写的代码为



说话人	特效	文本
in and main	template syl noblank	!retime("start2syl",syl.i*50-500,0)!{\fry-90\moves4($scenter,!$smiddle+100!,$scent
eft or ext	template syl noblank	!retime("syl2end",-$sdur,-200)!{\pos($center,$middle)\t(0,$sdur,\fscx135\fscy135\f
out	template syl noblank	!retime("postline",syl.i*50-700,syl.i*50+500)!{\fry0\move($center,$middle,$center,!
		+500 and +700

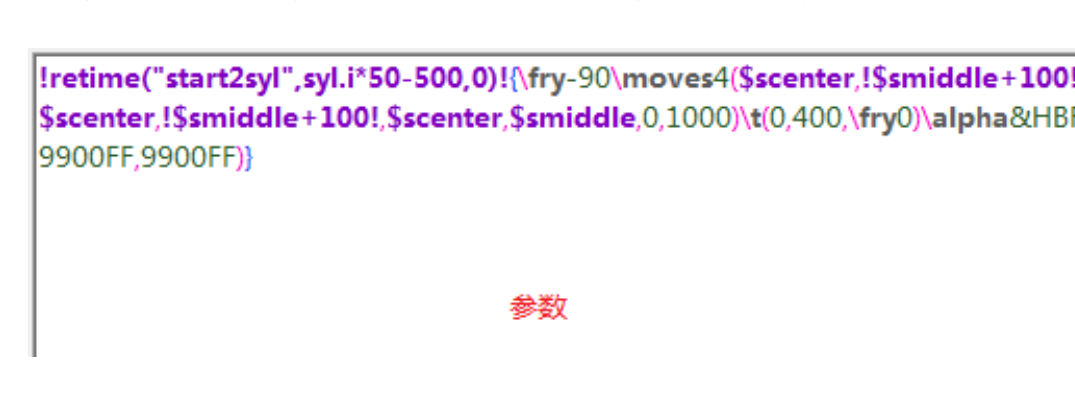
33.入场和存在: start2syl模式中词的开始时间设置为syl.i*50-500,这个大家可以自行调整，结束时间为每个音节(词)的开始时间，不需要修改，

34.表现效果使用的模式为syl2end，由于前面的main方式的结束时间为音节的开始时间，所以后面的eft效果的开始时间为-$sdur(当前音节的持续时间)，通过$sstart=$send-$sdur计算得出音节的开始时间，后面行的结束时间为-200，提前200ms结束，这样是为了给后面的退场效果保留一些时间，如果为0，表现效果不能及时的消失，造成短暂的停留，会影响后面的字幕的入场效果

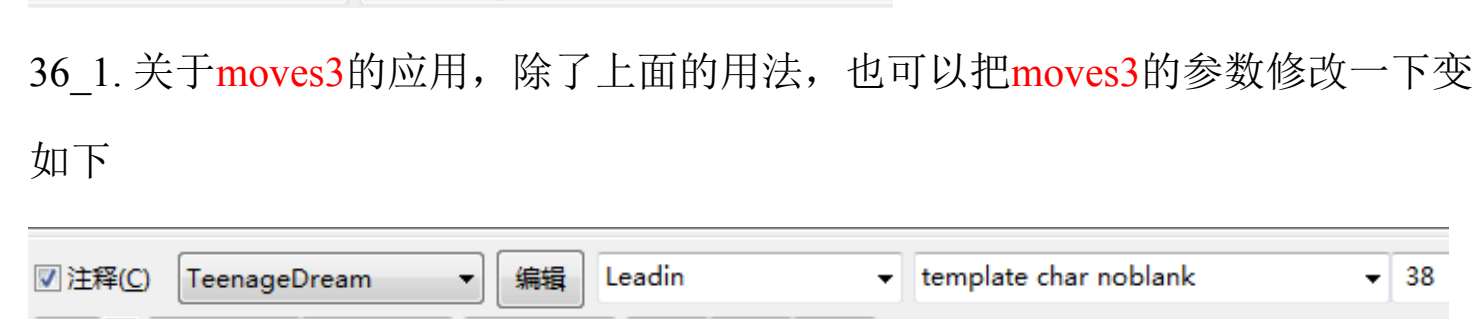


样式	说话人	特效	文本
and	in and main	template syl noblank	!retime("start2syl",syl.i*50-500,0)!{\fry-90\moves4($scenter,!$smiddle-
and			
and	eft or ext	template syl noblank	!retime("syl2end",-$sdur,0)!{\pos($center,$middle)\t(0,$sdur,\fscx135\
and			

35.退场效果使用的模式为postline，每一行的开始时间为当前行的结束时间，结束时间为该行的结束时间加上调整时间a，和调整时间b，a设置的和入场时间一样，后面b的调整时间为syl.i*50+500，和前面第30一样，造成一个时间差，文字逐个延迟消失，入场和退场的效果和tcax的相似，也是每一帧进入一个文字，退出也每一帧消失一个文字

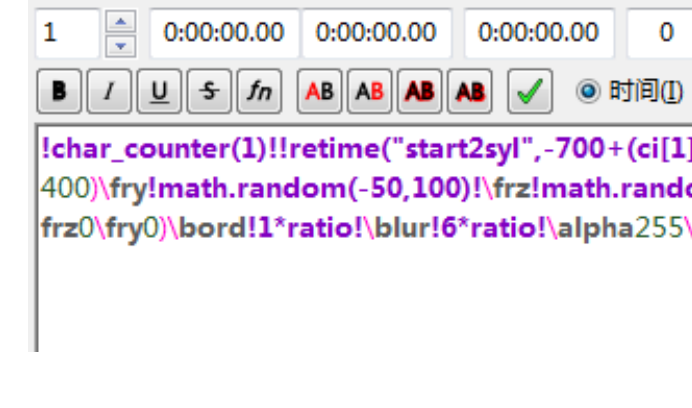


36.moves3或moves4，为Spline-move曲线移动，\moves3(x1,y1,x2,y2,x3,y3,0,0)，\moves4(x1,y1,x2,y2,x3,y3,0,0)，参数和move一样，如果是用于文字可以把最后两个坐标设置为音节的默认坐标，设置的参数和生成的效果

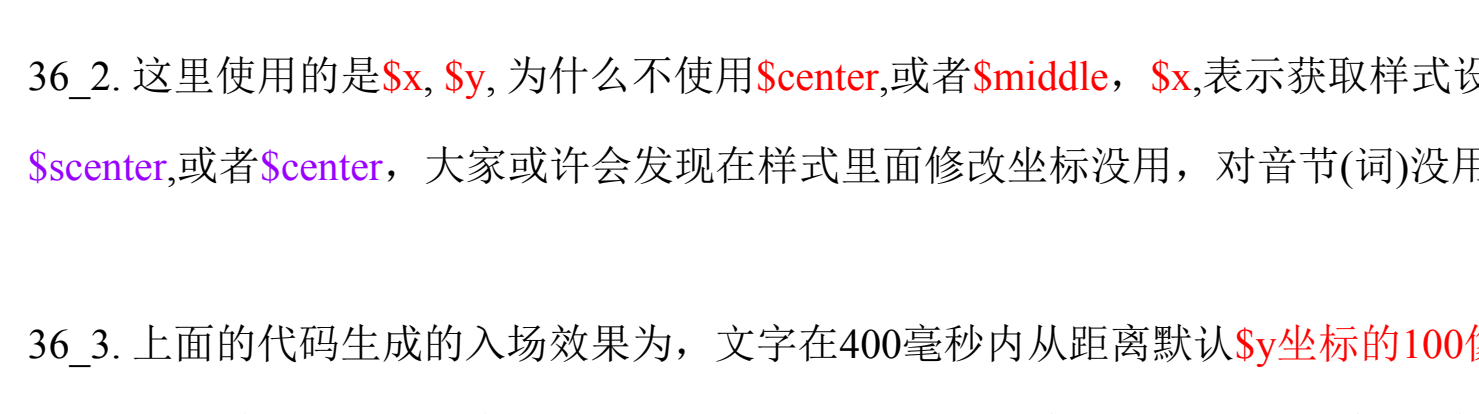


```
!retime("start2syl",syl.i*50-500,0)!{\fry-90\moves4($scenter,!$smiddle+100!,$scenter,!$smiddle-100!,
$scenter,!$smiddle+100!,$scenter,$smiddle,0,1000)\t(0,400,\fry0)\alpha&HBF&\1vc(00FFFF,FF0000,
9900FF,9900FF)}
```

参数



36_1. 关于moves3的应用，除了上面的用法，也可以把moves3的参数修改一下变成更好看的入场效果，代码如下


☑ 注释(C) TeenageDream 编辑 Leadin template char noblank 38

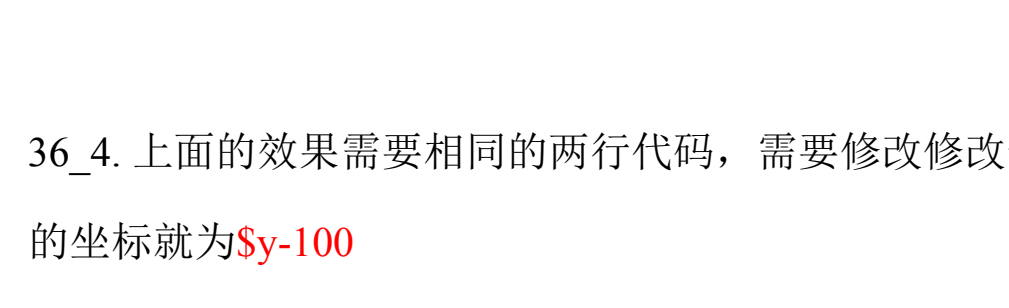
1 0:00:00.00 0:00:00.00 0:00:00.00 0 0 0

● 时间(I) ○ 帧(R) ☐ 显示原始字幕

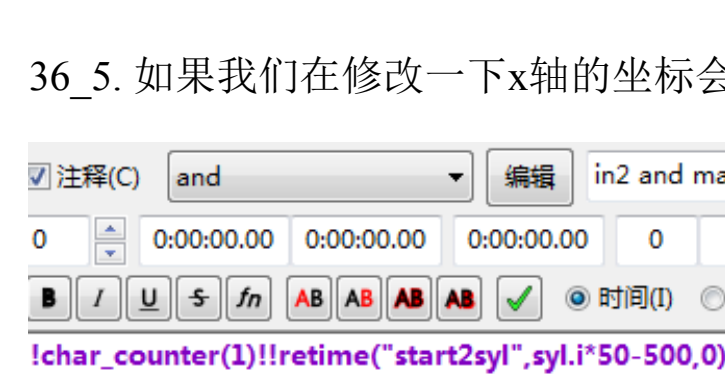
```
!char_counter(1)!!retime("start2syl",-700+(ci[1]-1)*50,0)!{\moves3($x,!$y+100!,$x,!$y-100!,$x,$y,0,
400)\fry!math.random(-50,100)!\frz!math.random(-90,90)!\fry!math.random(-100,200)!\t(0,700,\frx0\
frz0\fry0)\bord!1*ratio!\blur!6*ratio!\alpha255\3c!_G.ass_color(_G.HSV_to_RGB((ci[1]-1)/cn*360,1,1))!}
```

36_2. 这里使用的是$x, $y, 为什么不使用$center,或者$middle，$x,表示获取样式设置的x轴的坐标，如果是使用$scenter,或者$center，大家或许会发现在样式里面修改坐标没用，对音节(词)没用影响

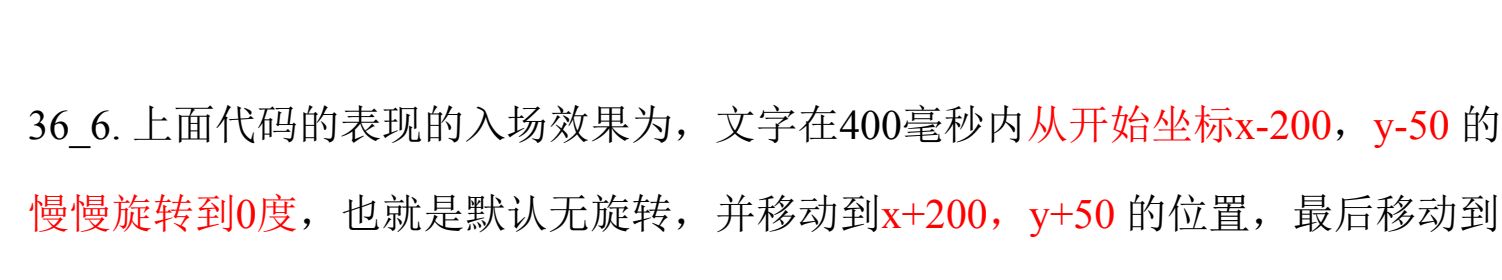
36_3. 上面的代码生成的入场效果为，文字在400毫秒内从距离默认$y坐标的100像素的下方,也就是 $y+100移动到$y坐标的上方100像素, $y-100 的位置，期间通过绕fry/x/z轴产生随机旋转的效果，最后回到默认的坐标位置



36_4. 上面的效果需要相同的两行代码，需要修改修改一下$y的坐标，上面的坐标为$y+100，下面的第二行的坐标就为$y-100



36_5. 如果我们在修改一下x轴的坐标会发生什么，代码如下

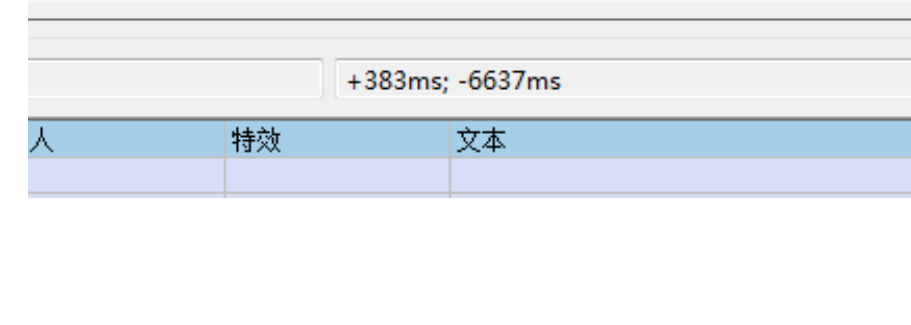

☑ 注释(C) and 编辑 in2 and main2 template syl noblank 37

0 0:00:00.00 0:00:00.00 0:00:00.00 0 0 0

● 时间(I) ○ 帧(R) ☐ 显示原始字幕

```
!char_counter(1)!!retime("start2syl",syl.i*50-500,0)!{\moves3(!$x-200!,!$y-50!,!$x+200!,!$y-50!,$x,$y,
0,400)\fry180\frz180\frx180\t(0,700,\frx0\frz0\fry0)\bord!1*ratio!\blur!6*ratio!\alpha255\3c
!_G.ass_color(_G.HSV_to_RGB((ci[1]-1)/cn*360,1,1))!}
```

36_6. 上面代码的表现的入场效果为，文字在400毫秒内从开始坐标x-200，y-50 的位置，绕frx/y/z轴180度，慢慢旋转到0度，也就是默认无旋转，并移动到x+200，y+50 的位置，最后移动到默认x和y轴的默认位置



36_7. 个人技术有限，关于moves3也就介绍这么多，其实moves4也是一样的，大家有兴趣可以尝试一下修改其它的参数来实现更好的表现效果

37.如果新手觉得可以的话也可以用temp写进入或者退场的效果，用tcax写表现效果也可以，相互使用，感觉讲解的也差不多的了，没什么细节问题了，剩下的就是需要新手多去练习，熟悉temp后再去用tcax写py特效脚本，那时候你就会发现，tcax也不是那么的难，希望对新手们有所帮助

CMD