

# ASS Tags of Aegisub

本文由和路雪。基于Aegisub官方指南自翻整理编写，不保证涵盖官方指南所有内容，如有疏漏，还望见谅。

本文的主要目的为方便做时间轴的同学迅速上手、快捷查阅代码；本文大部分代码配有英文代码名称，本人英语能力有限，为避免翻译过程中产生误解给您带来不便，如有需要，还请参考原文。

本文仅为示例，请在实践中熟悉代码。

此外，本文代码排列顺序与官方指南并非完全一致，顺序实则为本人擅自划分类别排列，还请注意。请擅用搜索功能，以便快速定位到您想查找的代码。

编者使用软件版本为2.1.9，如代码格式无效，请注意是否是由您使用的软件版本过高或者过低导致的。


本文可自由转载，但请保留编者以及本页信息。

## 参考资料：

[http://aegisub.cellosoft.com/docs/ASS\\_Tags](http://aegisub.cellosoft.com/docs/ASS_Tags)

网上流传最广的出处不可考的Aegisub教程（开头为介绍Aegisub英文操作界面）

## 文中使用符号说明：

- \* → 代码注意事项
- 【】 → 代码简要说明
- e.g. → 代码使用举例
- () → 解释、强调、补充以及吐槽等
-  → 高阶代码，只在VSFilter2.39+的版本使用可能  
These tags are only available in VSFilter 2.39 and later, and might not be safe for softsubbing.

## 部分中英对照参考：

圆括号	parentheses	→	()
尖括号	angle brackets	→	< >
逗号	comma	→	,
反斜线符号	backslash	→	\
大括号	override blocks	→	{ }
斜体	<i>Italics</i>		

(J-u)J L-u-L)[GiveMeFive!]

# ASS Tags of Aegisub

## 00 代码格式说明

反斜线符号\+代码名称+参数（有可能为多个）

**\*所有代码必须是英文格式，中文标点无效（生效了的代码是彩色的）；**

\*尖括号<>内的参数必填；

\*<>本身不需要输入代码；

\*[]内参数可以省略，具体用法参照相关内容。

e.g.

\fs<size>            \fs50            【字体大小设置为50】

\pos (<x>, <y>)   \pos(470, 260)   【字幕位置定位于点(<x>, <y>)】

\move (<x1>, <y1>, <x2>, <y2> [, <t1>, <t2>])

\move (100, 150, 300, 350) 【字幕从点 (100, 150) 移动至 (300, 350)，持续时间为从字幕出现开始到字幕消失为止】

## 01特殊代码 Special characters （不需写在 {} 内）

\n – 软性换行 Soft line break

【只在换行方式 (Wrapping mode) 2(详情见下) 下有效，其他方式下则被替换为空格】

\N – 硬性换行 Hard line break 【无视换行方式，强制换行】

\h – 硬性空格 Hard space 【在硬性空格前后不会自动换行，且其在一行首尾时不会被缩进】

换行方式 Wrap Style: \q<style> (Override tag)

**\*此为Override tag，需要 {}，写在这里仅为方便对照查阅；**

\*共计四种，可在“文件→配置”中更改。

0：智能换行，尽量使每行宽度一致，不能保持等宽时上行较宽。只有\N指令强制换行有效；

1：于行尾词处换行，仅\N断行有效；

2：不自动换行，\n及\N断行有效；

3：智能换行，尽量使每行宽度一致，不能保持等宽时下行较宽。只有\N指令强制换行有效。

**02 普通代码 Override tags**（必须写在 {} 内，且影响之后所有的文字（当行代码），直到有第二个相同类型的代码出现以覆盖其效果）

° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \* ° ° ° ° ° \* \* \* \* \*

可使用面板按钮便捷设置——



斜体 **Italics:**        \iI \iO

粗体 **Bold:**        \b1 \b0 \b<weight>\*

\*大部分字体不支持自定义字体粗细；如支持，更改的粗细应为100的倍数；

\*100最细，400正常，700为粗体，900为极粗。

e.g. {\b100}How {\b300}bold {\b500}can {\b700}you {\b900}get?



**阴影深度 Shadow distance (extended):** `\xshad<depth> \yshad<depth>`

【横向的阴影深度(x)和纵向的阴影深度(y)可分别设置】

\*<size>可设为0，可为小数，也可负数。

e.g. `\xshad5\yshad0` `\xshad-6\yshad4`

自挂东南枝

自挂东南枝

**模糊边缘 Blur edges:** `\be1` 【模糊边缘】 `\be0` 【不模糊边缘】

\*字体有边框时以边框颜色为准模糊边缘，无边框时以文字颜色为准模糊文字边缘；

\*边缘模糊效果在字体较小时比较明显，但是使用可以使文字看过去更柔和。

e.g. `\be1` `\be0`

自挂东南枝

自挂东南枝

(嗯……左边的柔和一点，看出来了吧……)

**模糊边缘 Blur edges(gaussian kernel):** `\blur<strength>`

\*字体有边框时以边框颜色为准模糊边缘，无边框时以文字颜色为准模糊文字边缘；

\*设置值必须为整数；

\*因为模糊效果为渐变色，设置值过高没有意义（末端几乎透明）。

e.g. `\fs40\be10` `\fs40\be20`

自挂东南枝

自挂东南枝

(左右两者，基本看不出来区别=\_\_\_=)

**模糊边缘 Blur edges(gaussian kernel):** `\blur<strength>`

\*字体有边框时以边框颜色为准模糊边缘，无边框时以文字颜色为准模糊文字边缘；

\*设置值可为小数；

\*用法与 `\be<strength>` 几乎一样，但是因为算法不同，在设置值较高的时候有很好的表现，但要注意设置值过高会给CPU带来压力。

e.g. `\fs40\be5` `\fs40\be10`

自挂东南枝

自挂东南枝

**设置颜色 Set color**

`\1c&H<bb><gg><rr>&` (= `\c&H<bb><gg><rr>&`) 【主要颜色 (原色)】

`\2c&H<bb><gg><rr>&` 【卡拉 ok效果中的次要 (第二) 颜色】

`\3c&H<bb><gg><rr>&` 【字体边框颜色】

`\4c&H<bb><gg><rr>&` 【字体阴影颜色】

\*&.....&不可以省略；

\*颜色代码参照<http://en.wikipedia.org/wiki/Hexadecimal>，蓝绿红顺序；

\*可以用软件的选色盘查找。(参见下图)

\*可用面板按钮快捷     设置。

\*常见颜色代码：

正红： `&H0000FF&`

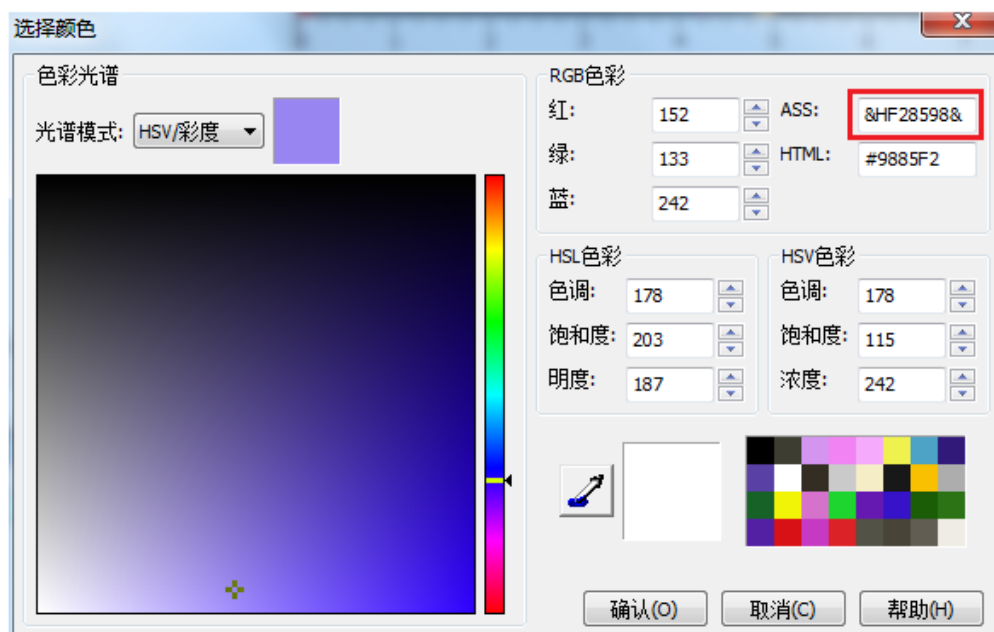
正黄： `&H00FFFF&`

正绿： `&H00FF00&`

纯白： `&HFCFCFC&`

纯黑： `&H000000&`

# ASS Tags of Aegisub



## 设置透明度 Set alpha

\alpha&H<aa> 【一次性设置以下四者的透明度】

\1a&H<aa> 【文字原色透明度】

\2a&H<aa> 【卡拉 ok效果中的第二颜色透明度】

\3a&H<aa> 【字体边框透明度】

\4a&H<aa> 【字体阴影透明度】

\*经实践Set alpha代码内的&和H均可省略不写；

\*<aa>采用十六进制（00【不透明】，37【25%透明】80【50%透明】，FF【100%透明】）；

\*<aa>不区分大小写；

\*超过80效果就不明显了（也许是我打开方式不对\_(:3」∠)\_）。

e.g.                    \alpha00                    \alpha37                    \alpha80

自挂东南枝   自挂东南枝   自挂东南枝

e.g.                    \1a0\3a0                    \1a80\3a0                    \1aFF\3a0

自挂东南枝   自挂东南枝   自挂东南枝

## 字体编码 Font encoding: \fe<id>

\*可在样式编辑器中设置。

e.g. \fe0 【ANSI码 (USA)】    \fe128 【SHIFT\_JIS 码 (Japan)】

0 - ANSI	0 - ANSI编码	161 - Greek	161 - 希腊文
1 - Default	1 - 默认编码	162 - Turkish	162 - 土耳其文
2 - Symbol	2 - 符号编码	163 - Vietnamese	163 - 越南文
77 - Mac	77 - Mac编码	177 - Hebrew	177 - 希伯来文
128 - Shift_JIS	128 - 日文Shift_JIS	178 - Arabic	178 - 阿拉伯文
129 - Hangeul	129 - 韩文Hangeul	186 - Baltic	186 - 波罗的海文
130 - Johab	130 - 韩文Johab	204 - Russian	204 - 俄文
134 - GB2312	134 - 简体中文GB2312	222 - Thai	222 - 泰文
136 - Chinese BIG5	136 - 繁体中文BIG5	238 - East European	238 - 东欧语系
		255 - OEM	255 - OEM编码

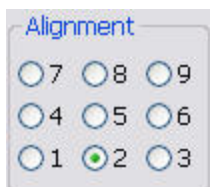
字幕定位、移动和淡入淡出。

栅格定位 Line alignment: \an<pos>

\*\pos和\move会覆盖\an指令效果，但是\an的特征会保留；

(如\pos(300,500)\an1, 字幕是居左的, 在字幕有两行以上的情况下每行开头均居左, 否则是居中)

\*屏幕划分方法:



e.g. \an2 【屏幕下方中间（各位置间距可在样式管理中重新设置）】

栅格定位 Line alignment(legacy): \a<pos>

\*\pos和\move会覆盖\ a指令效果，但是\ a的特征会保留（同\ an）；

\*屏幕划分方法:

• 1: Bottom left	1: 底端最左侧
• 2: Bottom center	2: 底端居中
• 3: Bottom right	3: 底端最右侧
• 5: Top left	5: 顶端最左侧
• 6: Top center	6: 顶端居中
• 7: Top right	7: 顶端最右侧
• 9: Middle left	9: 中部最左侧
• 10: Middle center	10: 中部居中
• 11: Middle right	11: 中部最右侧

静态定位 Set position: \pos(<x>, <y>)

\*准确度与分辨率有关;

\*可双击屏幕设置;

\*点(<x>,<y>)的位置不是绝对的(不见得是整条字幕的中心),有时与Alignment的选择有关。

**移动命令 Movement:** \move(<x1>, <y1>, <x2>, <y2> [, <t1>, <t2>])

\*t1和t2的单位为毫秒;

\*一行文本中，不得有多于一条的\move命令，也不能同时有\move和\pos;

\*移动只可能是匀速的。

【从位置 (x1,y1)移动到(x2,y2) [字幕开始时间之后t1毫秒开始移动; 从t1开始, 移动过程持续t2时间], 若t1,t2省略, 默认移动时间为此栏字幕条持续时间】

e.g.

`\move(100,150,300,350)` 【字幕从 (100, 150) 移动至 (300, 350), 持续时间为整条字幕时间】

`\move(100,150,300,350,500,1500)` 【同上，但字幕从其出现 500ms 后开始移动，1.5s 后移动结束】

旋转原点 Rotation origin: \org(<x>,<y>)

\*一旦旋转/缩放中心被设定, 则它就是固定不动的;

\*将该点设于极远处、配合快速特效可以造成跳跃的视觉效果；（反正我不会用OTZ）

\*每条文本最多只能有一个org命令。





## ASS Tags of Aegisub

**\frx<amount>** 【绕 X轴（水平轴）旋转(为正值时字向后倒，为负时字向前倾)】

e.g.

\frx60

\frx~60



`\fry<amount>` 【绕 Y轴（竖直轴）旋转(为正值时字左大右小，为负时字左小右大)】

e.g.

\fr{60}

\fry~60



其它尝试:

e.g.

\frx40\fry60

\frx~30\fry~30\frz30



 文本倾斜 Text shearing: \fax<factor> \fax<factor>

### 【文本倾斜效果】

\*一般而言<factor>取较小的值，小于2的值效果都非常明显；

\*在旋转命令之后，倾斜所应用的轴为旋转之后的轴，与初始轴无关。

e.g.

\fax1

\fay1



e.g. 【配合旋转】

\fr90\fax0.5

 $\frac{90}{100} = 0.9$ 

卡拉OK特效

卡拉OK特效 Karaoke effect:

\*<duration>单位为厘秒 (cs) , 100cs = 1000ms = 1s。

\k<duration> 【卡拉 ok跳色效果，一个字一个字间断变色】

\*`<duration>`的时间经过之后执行跳色命令。

e.g. { \k50} { \k80}Think { \k25}of { \k60}me

【50cs之后 "Think"从次要颜色跳为主要颜色， 80cs后"of"跳为原色， 25cs后"me"跳为原色】





# ASS Tags of Aegisub

e.g. `\clip (1, m 50 0 b 100 0 100 100 50 100 b 0 100 0 0 50 0)`

【绘图1: 1, 显示圆内字幕】

## 动态变换 Animated transform

`\t(<style modifiers>)` 【在字幕持续时间内由\t前的状态变成()内状态】

`\t(<accel>, <style modifiers>)` 【同上, 速度非线性】

`\t(<t1>, <t2>, <style modifiers>)` 【在t1~t2时间内由\t前的状态变成()内状态】

`\t(<t1>, <t2>, <accel>, <style modifiers>)` 【同上, 速度非线性】

\*只有以下tags可以用于变换:

Font	Geometry	Other effects
<code>\fs</code>	<code>\fscx</code>	<code>\bord</code>
<code>\fsp</code>	<code>\fscy</code>	<code>\shad</code>
<code>\c</code>	<code>\frx</code>	<code>\dip</code>
<code>\1c</code>	<code>\fry</code>	
<code>\2c</code>	<code>\frz</code>	
<code>\3c</code>	<code>\fr</code>	
<code>\4c</code>		
<code>\alpha</code>		
<code>\1a</code>		
<code>\2a</code>		
<code>\3a</code>		
<code>\4a</code>		



`\fax \fay \be \blur \xbord \ybord \xshad \yshad \iclip`

\*t1和t2单位为毫秒, 时间为累计时间;

\*t1~t2的时间间隔为特效发生的时间, 如果未指定, 则为整条字幕持续时间;

\*<accel>可以实现非线性速度变化, 具体为: <accel> 等于 1, 速度为线性; <accel> 在0和1之间, 开始速度快结束速度慢; <accel> 大于 1, 开始速度慢结束速度快。

\*自己实践的时候发现\i)clip不好用, 不如说和其代码本身说明是矛盾的, 无解ING

e.g.

`{\1c&HFF0000&\t(\1c&H0000FF&)}Hello!` 【“Hello!”由蓝色变为红色】

`{\an5\t(0,5000,\frz3600)}Wheee` 【“Wheee”在5s内匀速逆时针转10圈】

`{\an5\t(0,5000,0.5,\frz3600)}Wheee` 【“Wheee”在5s内减速逆时针转10圈】

`{\an5\fscx0\fscy0\t(0,500,\fscx100\fscy100)}Boo!` 【Boo!在0.5s内由一个点变成文字】

## 03 绘图代码 Drawing tags

激活绘图命令 Toggle drawing mode: `\p<0/1/..n>`

`\p1` 【激活绘图命令, 1: 1绘图】

`\p0` 【关闭绘图】

`\p2` 【激活绘图命令, 用2倍的分辨率绘图, 2: 1绘图】

`\p4` 【激活绘图命令, 用大8倍 ( $2^{(4-1)}$ ) 的分辨率绘图, 8: 1绘图】

\* $n > 0$ , n为所扩大的范围, 用 $2^{(n-1)}$ 转化 (可使所做图像更精细)。

基线偏移 Baseline offset: `\pbo<y>` 【图像沿y轴平移】

\*该命令只应用于y轴 (纵向)。

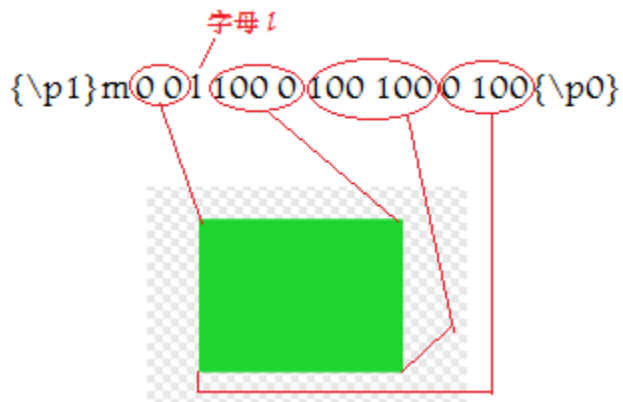
e.g. `\pbo-50` 【图像沿Y轴向上平移50像素】 `\pbo100` 【图像沿Y轴向下平移100像素】

## 04 绘图命令 Drawing commands

\*这些命令必须遵循\clip (矢量作图) 的代码格式, 或者位于 {\p#}和{\p0}之间。

正方形 **Square**     {\p1}m 0 0 1 100 0 100 100 0 100{\p0}

\*改变数字可以画出各种四边形, 请自行探索~



椭圆 **Rounded square**     {\p1}m 0 0 s 100 0 100 100 0 100 c {\p0}

【请参照Square的图示理解代码】

\*此例中c等于 "p 0 0 100 0 100 100"。 (具体的请参照后文的补充解释)



(几近) 圆形 **Circle (almost)**

       {\p1}m 50 0 b 100 0 100 100 50 100 b 0 100 0 0 50 0{\p0}

\*此处第二个b可省略。



补充解释:

**移动 Move:**             m <x> <y>     【绘制点移动至点(x, y)】

\*所有的绘图命令都必须用这个标记开始;

\*如果此前有个未封闭的形状, 系统自动将其封闭。

(不封闭) **移动 Move (no closing) :**             n <x> <y>     【绘制点移动至点(x, y)】

\*如果此前有个未封闭的形状, 不封闭现有的形状。

**线条绘制 Line:**             l <x> <y>

【以现在所在位置和(x, y) 坐标为两端点绘制一条直线, 同时绘制点移动至(x, y)】

