

Abreviaciones de algunos Ass Tags y tags progresivos [KE]

Abreviaciones de algunos tags [KE] :

Las siguientes cinco abreviaciones hacen posible que al llamar un solo tag, podamos aplicar dos o más de ellos al mismo tiempo. Las abreviaciones son las siguientes:

- `\fscxy`
- `\frxy`
- `\frxz`
- `\fryz`
- `\frxyz`

`\fscxy` es equivalente a los tags `\fscx` y `\fscy` al tiempo, y hay dos formas diferentes de hacerlo.

Ejemplo:

LUA:

- `"\fscxy120" = \fscx120\fscy120`
- `"\fscxy80" = \fscx80\fscy80`

Automation Auto-4:

- `\fscxy200 = \fscx200\fscy200`
- `\fscxy25 = \fscx25\fscy25`

O sea que siempre que usemos valores constantes en esta abreviatura, dichos valores serán los mismos para los tags a los que equivale. En el caso de que queramos poner un valor aleatorio (random), debemos poner la función junto con la abreviatura. Ejemplo:

LUA:

- `"\fscxyR(100,200)" = \fscx132\fscy157`
- `"\fscxyRd(10,50)" = \fscx34.9\fscy18.7`

Automation Auto-4:

- `\fscxyRm(0,1) = \fscx0.854\fscy0.051`
- `\fscxyRc(23,37) = \fscx25.67\fscy33.78`

Entonces, al adjuntarle a la abreviación la función random, garantizamos que a los tags equivalentes les corresponda un valor diferente para cada uno de forma aleatoria.

Las otras cuatro abreviaturas corresponden a los siguientes tags:

- `\frxy = \frx\fry`
- `\frxz = \frx\frz`
- `\fryz = \fry\frz`
- `\frxyz = \frx\fry\frz`

Y de forma similar a los ejemplos de vistos en la abreviatura \fscxy, son aplicables las mismas condiciones, tanto para los valores constantes, como para los valores aleatorios al usar la función **R** o alguna de sus nuevas modificaciones.

Estas cinco abreviaciones no consisten un efecto en sí mismas, pero nos ayudan a ahorrar trabajo en los mismos. Disponibles en el **KE** versión **3.2.9.5** o superior.

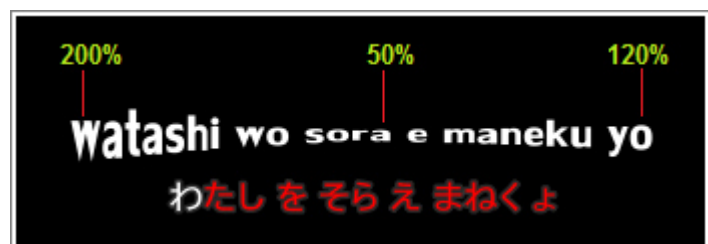
tags progresivos [KE] :

Esta es otra de las actualizaciones que se pueden usar a partir de la versión **3.2.9.5** del **Kara Effector**, y lo que hace es interpolar todos los valores dentro de una tabla adjunta a un tag, respecto al **module1** y al **module**, según así lo dispongamos.

Ejemplo:

Template Type [fx]:	Char
Add Tags Language:	Lua
<code>"\fscy{200, 50, 120}"</code>	

Dado que es un **Template Type: Char**, cada una de los caracteres de la línea tomarán los valores en el tag \fscy empezando desde el 200%, pasando por el 50% hasta llegar al 120%:



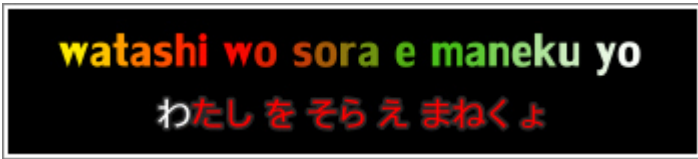
La interpolación de los diferentes porcentajes en la escala de la Font respecto al eje “y” se hizo en relación al **module1**.

Ejemplo:

Template Type [fx]:	Char
Add Tags Language:	Automation Auto-4
<code>\1c{"&H00FFFF&", "&H0000FF&", "&H16C047&", "&HFFFFFF&"}</code>	

Los colores de la tabla adjunta son:

- **&H00FFFF& = Amarillo**
- **&H0000FF& = Rojo**
- **&H16C047& = Verde**
- **&HFFFFFF& = Blanco**



De los dos ejemplos vistos anteriormente podemos deducir que al adjuntar normalmente la **tabla** al tag, los valores de la misma se interpolarán respecto al **module1**. Ahora para que los valores de la tabla se interpolen respecto al **module**

(respecto al **loop**) debemos añadir el signo menos (-) justo en medio del tag y la **tabla**:

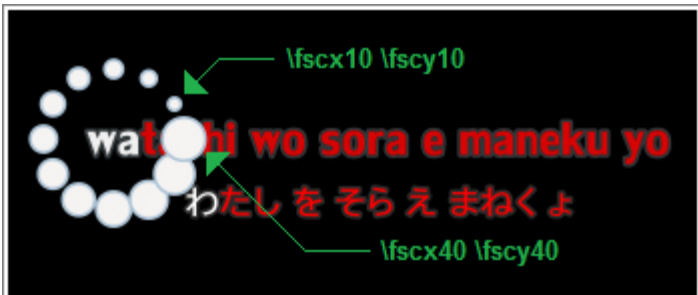
Ejemplo:

Template Type [fx]:	Syl
Pos in "X" =	fx.pos_x + math.polar(360*j/maxj, 70, "x")
Pos in "Y" =	fx.pos_y + math.polar(360*j/maxj, 70, "y")
loop =	12
Return [fx]:	shape.circle

Y en **Add Tags** ponemos:

Add Tags Language:	Automation Auto-4
\fscxy-{10, 40}	

Aplicando, con los tiempos de un **hi-light**, veremos esto:



Al usar una de las abreviaciones vistas hace poco se ahorra aún más trabajo, dado que:

`\fscxy-{10, 40} = \fscx-{10, 40}\fscy-{10, 40}`

Entonces los valores de la **tabla** se aplicarán a dichos tags de forma progresiva, interpolándose desde el 10% hasta llegar al 40%, como se puede ver en la imagen anterior.

A este mismo ejemplo le podemos aplicar la cantidad de tags progresivos, abreviaciones o normales que queramos, todo depende de los resultados que estemos necesitando, ejemplo:

`\fscxy-{10, 40}\lc-{"&H00FFFF&","&H0000FF&"}`

Las combinaciones posibles son infinitas, solo resta probar.