

Herramienta: tags abreviados y tag funciones Parte 1

El tema que veremos a continuación, considero que es uno de los más importantes para todos aquellos que les gusta desarrollar y llevar a cabo aquellos efectos que se les vienen en mente o que se presentan en los diferentes vídeos y que desean de alguna manera poder “imitarlos” para sus propios proyectos. El tema consiste es una serie de abreviaciones, convenciones y un par de trucos más que el **KE** tiene en sus librerías para que lo que antes eran simples tags, ahora se conviertan en funciones, y por qué no, en efectos en sí mismos.

Parte de esta herramienta ya la empezamos a ver en el artículo [Abreviaciones de algunos Ass Tags y tags progresivos \[KE\]](#), y no está de más que los recordemos y ver en qué consistía cada uno de ellos:

tags abreviados

tag Ejemplo (valor constante) resultado

| | | |
|---|------------------------------------|--|
| 1 | <code>\\fscxy “\\fscxy147”</code> | <code>\\fscx147\\fscy147</code> |
| 2 | <code>\\frxy “\\frxy-23”</code> | <code>\\frx-23\\fry-23</code> |
| 3 | <code>\\frxz “\\frxz12”</code> | <code>\\frx12\\frz12</code> |
| 4 | <code>\\fryz “\\fryz345”</code> | <code>\\fry345\\frz345</code> |
| 5 | <code>\\frxyz “\\frxyz-360”</code> | <code>\\frx-360\\fry-360\\frz-360</code> |

Estas mismas cinco abreviaciones de tags también las podemos usar con la función **R** y todas sus nuevas actualizaciones que recién vimos:

tags abreviados

tag Ejemplo (valor aleatorio)

| | |
|---|---|
| 1 | <code>\\fscxy “\\fscxyR(100, 200, 10)”</code> |
| 2 | <code>\\frxy “\\frxyRs(20, 50)”</code> |
| 3 | <code>\\frxz “\\frxzRcs(-125, 215)”</code> |
| 4 | <code>\\fryz “\\fryzRd(22, 43)”</code> |
| 5 | <code>\\frxyz “\\frxyzRs(360)”</code> |

Ahora también podemos usar, y no solo con estos cinco tags sino también con todos, operaciones y/o funciones, con solo añadir un paréntesis después de los mismos:

Ejemplo:

- `“\\fscxy(120 + Rs(40))”`
- `“\\fr(45 * (-1) ^ syl.i)”`

Add Tags Language:

>>
Lua

"\\fr(45 * (-1) ^ syl.i)\\fad(200,0)"

Add Tags Language:

>>
Automation Auto-4

\\frxyz(120 – Rs(360) + 0.4 * fx.dur)\\fad(200,0)\\1a&HFF&

Entonces, cualquier tag lo podemos usar de las anteriores tres formas que acabamos de ver:

- Valor constante
- Valor aleatorio
- Operación y/o función

Esto nos ahorrará un poco de trabajo y hace que sea más simple el uso de cada uno de ellos. Y con el fin de seguir ahorrando tiempo y trabajo, el **KE** ya consta de otras abreviaciones que podemos usar de las tres formas ya mencionadas. Las primeras tres nuevas abreviaciones básicas que veremos son las siguientes:

nuevos tags abreviados

tag equivalencia

1 \\faxy \\fax + \\fay

2 \\xybord \\xbord + \\ybor

3 \\xyshad \\xshad + \\yshad

Ejemplo:

Add Tags Language:

>>
Lua

tag.oscill(fx.dur, 80, "\\faxyRms(0.2)"), "\\bord0\\shad0\\fad(0,120)"

Las anteriores ocho abreviaciones son las consideradas “básicas”, y a partir de ellas se abre un mundo de posibilidades ante nosotros, que espero poder explicarlas todas y no olvidar ninguna por el camino. Empecemos viendo un resumen de las abreviaciones vista hasta este momento y unas nuevas que ya merecen ser mencionadas:

tags abreviados

| tag | equivalencia |
|------------------------|---------------------------------|
| 1 <code>\fscxy</code> | <code>\fscx + \fscy</code> |
| 2 <code>\frxy</code> | <code>\frx + \fry</code> |
| 3 <code>\frxz</code> | <code>\frx + \frz</code> |
| 4 <code>\fryz</code> | <code>\fry + \frz</code> |
| 5 <code>\frxyz</code> | <code>\frx + \fry + \frz</code> |
| 6 <code>\faxy</code> | <code>\fax + \fay</code> |
| 7 <code>\xybord</code> | <code>\xbord + \ybord</code> |
| 8 <code>\xyshad</code> | <code>\xshad + \yshad</code> |

La primera abreviación (**`\fscxy`**) agrupa los dos tags de porcentajes de tamaño, y cuando la usamos con un valor aleatorio, cada uno de los dos tags adquiere valores diferentes entre ellos, según los parámetros ingresados, ejemplo:

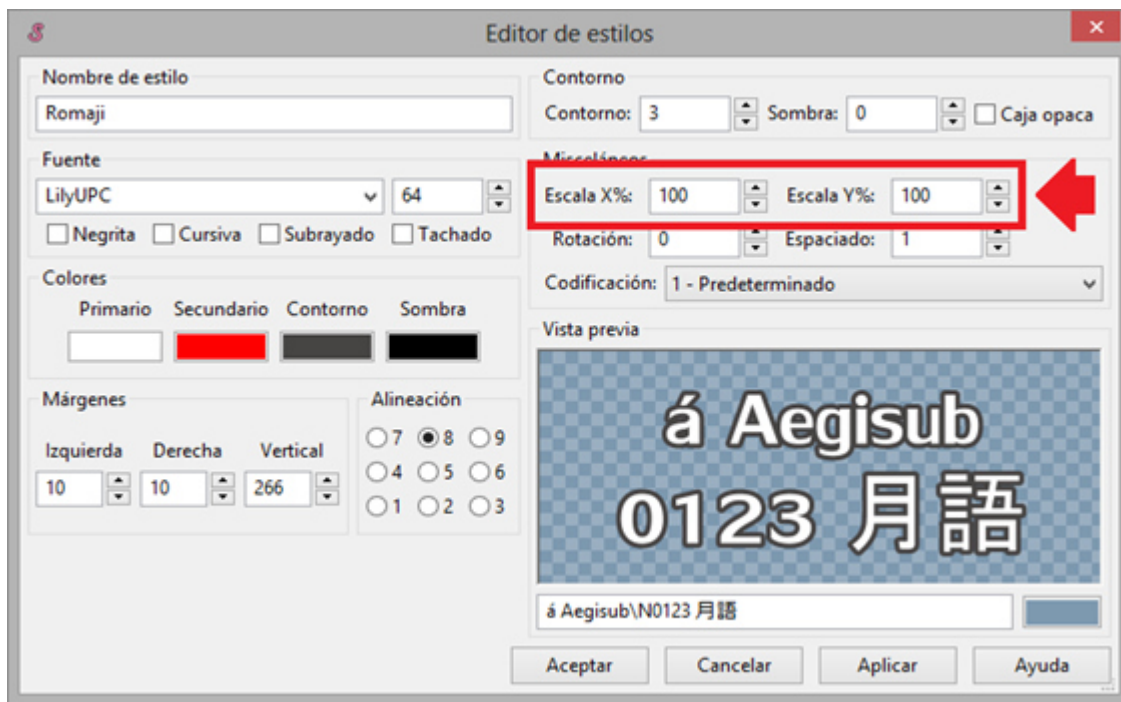
- `\fscxyR(80, 120) \fscx104\fscy86`

En el ejemplo anterior, 104 y 86 son posibles resultados aleatorios de la función **R**, cuyo valor mínimo sería 80 y valor máximo 120. El ejemplo anterior no garantiza que el valor aleatorio sea el mismo para ambos tags, y para que ello suceda, es necesario agregar la letra “i” al tag justo antes de la función **R**:

Ejemplo:

- `\fscxyiRc(40, 60) \fscx47.18\fscy47.18`

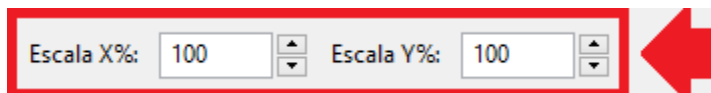
Lo que nos quiere indicar el ejemplo anterior es que el valor aleatorio que arroje la función **Rc** para el tag `\fscx`, será el mismo que para el tag `\fscy`, según los parámetros de la función random. Esta habilidad que le da la letra “i” a la abreviación `\fscxy` garantiza que la variación de tamaño de objeto karaoke sea siempre proporcional, siempre que los valores de escala en el Estilo de la línea a la que le apliquen el fx, sean iguales entre sí:



Para el caso en que estos dos valores no sean iguales entre sí en un Estilo, veremos las siguientes cuatro adaptaciones de los tags de escala de tamaño:

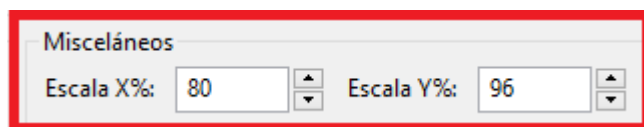
- `\\fscxr`
- `\\fscyr`
- `\\fscxyr`
- `\\fscxyir`

La letra “r” hace referencia al “ratio” o “razón” de la variación respecto a los valores de escala en un determinado Estilo de Línea



Ejemplo:

Supongamos que para cierto Estilo de Línea los valores de escala son los siguientes:



Entonces:

- `\\fscxr1` = `\\fscx80` porque $80 \times 1 = 80$
- `\\fscyr1` = `\\fscy96` porque $96 \times 1 = 96$
- `\\fscxr0.5` = `\\fscx40` porque $80 \times 0.5 = 40$
- `\\fscyr1.2` = `\\fscy115.2` porque $96 \times 1.2 = 115.2$
- `\\fscxr2.5` = `\\fscx200` porque $80 \times 2.5 = 200$
- `\\fscxyr1.5` = `\\fscx120\\fscy144` porque $80 \times 1.5 = 120$ y $96 \times 1.5 = 144$
- `\\fscxyr0.4` = `\\fscx32\\fscy38.4` porque $80 \times 0.4 = 32$ y $96 \times 0.4 = 38.4$

Así que el valor que coloquemos después de la “r” se multiplicará por el valor de escala original del Estilo de Línea al que le apliquemos el fx, entonces, si por ejemplo queremos que ambas escalas aumenten al doble de su valor original, lo único que tenemos que poner sería:

- `\\fscxyr2`

Si queremos que ambas escalas se modifiquen aleatoriamente de forma proporcional usamos `\\fscxyir`

Ejemplo:

- `\\fscxyirRd(1, 2) \\fscxyr1.6 \\fscxr1.6\\fscyr1.6 \\fscx128\\fscy153.6`

Para el próximo Tomo, seguiremos viendo más de las abreviaciones de los tags que podemos usar en el **Kara Effector** y que hará que hacer efectos sea una tarea aún más sencilla de lo que podríamos llegar a imaginarnos, así que estén pendiente de futuras entregas para que estén al tanto de todas las novedades.