

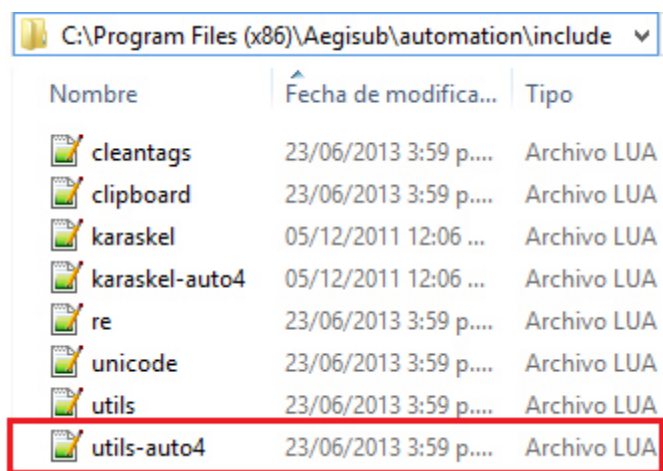
Las Funciones Compartidas de Automation Auto-4 y del Kara Effector

Cada vez estamos más cerca de comprender a cada una de las herramientas con las que podemos contar en el **Kara Effector** a la hora de hacer un Efecto para nuestros Subtítulos. Siguiendo con este orden de ideas, continuaremos viendo las funciones y variables que aún nos restan por ver.

Las Funciones Compartidas de Automation Auto-4 y del Kara Effector:

En el **Aegisub** se pueden usar una serie de funciones que hacen parte de la Librería del **Automation Auto-4**, que lleva por nombre: **utils auto-4**

La Librería **utils auto-4** es un archivo .lua que está en la carpeta “**include**” del **Aegisub**, y en ella están todas las funciones que se pueden usar para hacer Efectos en los **Templates**:



Todas la funciones de esta Librería se pueden usar en el **Kara Effector**. Esas funciones son:

- **ass_color(r, g, b)**
- **ass_alpha(a)**
- **ass_style_color(r, g, b, a)**
- **alpha_from_style(scolor)**
- **color_from_style(scolor)**
- **HSV_to_RGB(H, S, V)**
- **HSL_to_RGB(H, S, L)**
- **interpolate(pct, min, max)**
- **interpolate_color(pct, first, last)**
- **interpolate_alpha(pct, first, last)**

La explicación de estas y otras funciones del **Aegisub** están en su **Web Oficial**:

http://docs.aegisub.org/manual/Automation_4_utils.lua

Algunas de las anteriores funciones, me imagino que las han usado o las han visto en algún Efecto, es por eso que no profundizaré en detalles con ellas, o no al menos por ahora.

Hay otras funciones que también pertenecen al **Aegisub** y se pueden usar en el **Kara Effector**:

- **maxloop(new_loop)**
- **relayer(new_layer)**
- **retime(modo, add_start, add_end)**

maxloop(n): esta función determina el cantidad de veces que se repetirá un Línea fx. Ejemplo:

Add Tags: Add Tags Language: Lua

```
maxloop(3), "\\fad(200,0)'
```

Entonces cada Línea fx se triplica:

23	0:00:40.70	0:00:45.79	English			Me aferra
24	0:00:45.92	0:00:54.56	English			Dos cora
25	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*Ko
26	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*Ko
27	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*Ko
28	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*do
29	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*do
30	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*do
31	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*ku

Otro Ejemplo, en Lenguaje **Automation Auto-4**:

Add Tags: Add Tags Language: Automation Auto-4

```
!maxloop($sdur/10)!!retime("syl",0,0){\an5\b1}
```

Es decir que maxloop cumple una función muy similar a la celda de texto “**loop**“, pero con la diferencia de que la función **maxloop** se puede usar dentro de las nuevas funciones que se pueden hacer en la celda de texto “**Variables**“.

relayer(n): esta función determina el número de la capa de cada una de las líneas generadas por un efecto. El entero “**n**” determina el número de la capa. Un ejemplo sencillo en lenguaje **LUA** podría ser:

Add Tags: Add Tags Language: Lua

```
relayer(syl.i), "\\blur2'
```

Y este sería el resultado:

número de la capa



23	0	0:00:40.70	0:00:45.79	English			Me aferraré a
24	0	0:00:45.92	0:00:54.56	English			Dos corazones
25	1	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*Ko
26	2	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*do
27	3	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*ku
28	4	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*na
29	5	0:00:02.43	0:00:08.16	Romaji	lead-in	Effector [Fx]	*ho

O sea que el número de la capa de cada línea de efecto generada equivale a la posición de la sílaba dentro de la línea karaoke, esto es **syl.i**

Recordemos que el número de la capa (**layer**) determina la visibilidad de dos o más objetos que coinciden parcial o totalmente en su posición en el vídeo. Un objeto en el vídeo (ya sea una letra, una sílaba, una línea, una shape o lo que sea que pongamos en nuestras líneas en el **Aegisub**), se verá en frente del otro, sí o solo sí su número de capa es mayor a la del otro objeto con el que coincide en su posición o, si tienen el mismo número de capa y su número de línea en el script del **Aegisub** es mayor.

Y para usar esta función en lenguaje **Automation Auto-4** lo único que debemos hacer es escribirla dentro de los ya conocidos signos de admiración:

Add Tags: Add Tags Language: Automation Auto-4 ▼

!relayer(syl.i)\blur2

Es decir que **relayer** cumple una función muy similar a la celda de texto “**layer**“, pero con la diferencia de que la función relayer se puede usar dentro de las nuevas funciones que se pueden hacer en la celda de texto “**variables**“.

retime(modos, add_start, add_end): es una de las funciones más importantes y más usadas del lenguaje **Automation Auto-4**, ya que “**re-timea**” los tiempos de las líneas de efecto generadas y es la función que hace posible que categoricemos nuestros efectos en entradas, efectos de sílaba activa, efectos de salida y demás tipos de efectos dependiendo del **modo** que usemos en la función.

Esta función se puede usar también en el **Kara Effector** exactamente de la misma manera que se usa en el **Aegisub**, con la significativa diferencia de que en el **Kara Effector**, esta función tiene muchos más **modos** que la versión original en **Automation Auto-4**.

Los **modos** de la función **retime** que se pueden usar en **Automation Auto-4** son 10:

1. preline
2. line
3. postline
4. presyl
5. start2syl
6. syl
7. syl2end
8. postsyl
9. sylpct
10. set or abs

Estos 10 **modos** se pueden usar en el **Kara Effector** de la misma manera que se hace en el **Aegisub**, ya sea en lenguaje **LUA** o en **Automation Auto-4**.

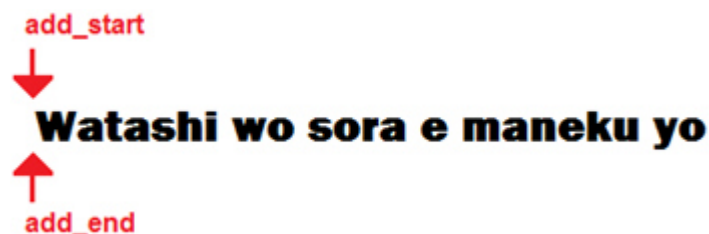
Los modos de la función **retime** que se pueden aplicar en el **Kara Effector** son 29:

1. preline
2. line
3. postline
4. preword
5. start2word
6. word
7. word2end
8. postword
9. presyl
10. start2syl
11. syl
12. syl2end
13. postsyl
14. prefuri
15. start2furi
16. furi
17. furi2end
18. postfuri
19. prechar
20. start2char
21. char
22. char2end
23. postchar
24. linepct
25. wordpct
26. sylpct
27. furipct
28. charpct
29. set o abs

retime(“preline”, add_start, add_end): el prefijo “pre” es la abreviatura de “previous” (previo, anterior, antes de) y “line” es “línea” en español. O sea que “preline” significa: antes de la línea.

El inicio y final del modo “**preline**” equivalen al tiempo de inicio de la línea karaoke (line.start_time). En este mismo orden de ideas, **add_start** es el tiempo en milisegundos que se agregará o sustraerá al inicio, y **add_end** es el tiempo en milisegundos que se agregará o sustraerá al tiempo final, que como ya lo había mencionado, es el mismo que el tiempo de inicio para este modo. Para sustraer tiempo, ya sea al inicio, al final o a ambos, ponemos valores negativos dentro de la función **retime**.

La siguiente gráfica muestra en dónde se encuentran por default el **add_start** y el **add_end** en el modo “**preline**”:



Lo que equivale a: **retime**("preline", 0, 0)

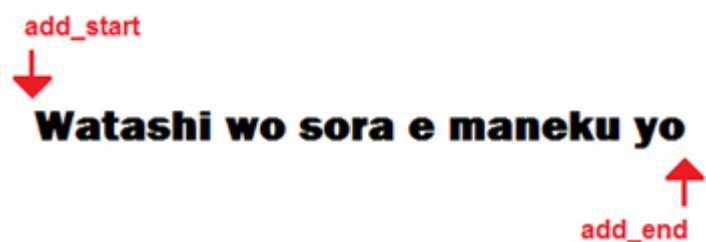
El modo “**preline**” puede ser usado en todo tipo de efecto o **Template Type**. Los recordamos:

- **Line**
- **Syl**
- **Furi**
- **Char**
- **Translation Line**
- **Translation Word**
- **Translation Char**

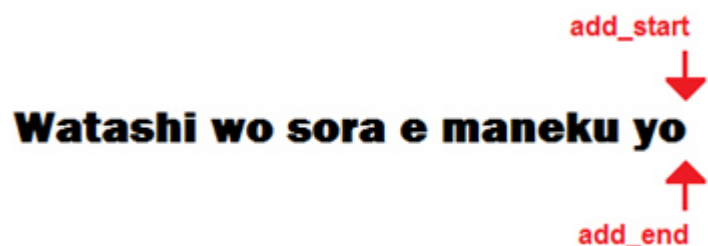
Algunos ejemplos de la función **retime** en modo “**preline**”:

- **retime**("preline", -200, 0)
- **retime**("preline", -300, 200)
- **retime**("preline", 400, 1000)
- **retime**("preline", 0, 2000)
- **retime**("preline", -800, -100)

retime("line", add_start, add_end): el tiempo de inicio equivale al tiempo de inicio de la línea karaoke (line.start_time) y el tiempo final, al tiempo final de cada una de las líneas de karaoke (line.end_time). Este **modo** está habilitado para todos los **Template Type**.

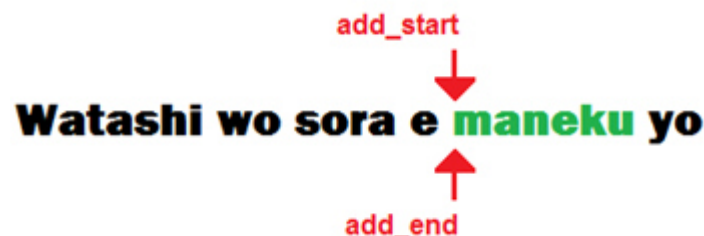


retime(“postline”, add_start, add_end): “post” de posterior, o sea, el tiempo después de la línea. El tiempo de inicio equivale al tiempo final de la línea karaoke (line.end_time) y el tiempo final, también al tiempo final de cada una de las líneas de karaoke (line.end_time). Este **modo** está habilitado para todos los **Template Type**.



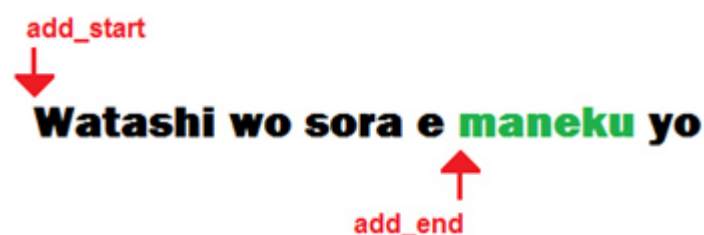
retime(“preword”, add_start, add_end): literalmente “antes de la **palabra**“. El tiempo de inicio equivale al tiempo de inicio de cada una de las palabras en cada línea karaoke (line.start_time + word.start_time) y el tiempo final, también al tiempo de inicio de cada una de las palabras en cada línea. Este **modo** está habilitado para los siguientes **Template Type**: Syl, Char, Translation Word y Translation Char.

Template Type [fx]:	Syl	
	Line	
	Syl	■
	Furi	
	Char	■
	Translation Line	
	Translation Word	■
	Translation Char	■



retime(“start2word”, add_start, add_end):

literalmente “desde el inicio hasta la **palabra**“. El tiempo de inicio equivale al tiempo de inicio de cada una de las líneas (line.start_time) y el tiempo final, al tiempo de inicio de cada **palabra** (line.start_time + word.start_time). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Char, Translation Word y Translation Char.

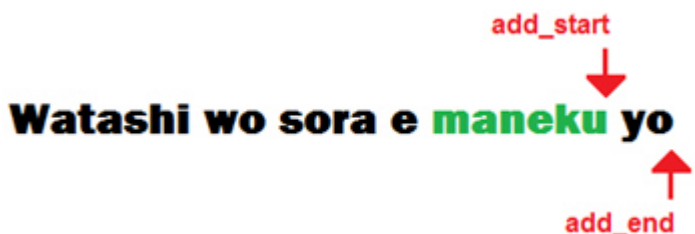


retime("word", add_start, add_end): literalmente "**palabra**". El tiempo de inicio equivale al tiempo de inicio de cada una de las palabras en cada línea karaoke ($\text{line.start_time} + \text{word.start_time}$) y el tiempo final, al tiempo final de cada una de las palabras ($\text{line.start_time} + \text{word.end_time}$). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Char, Translation Word y Translation Char.



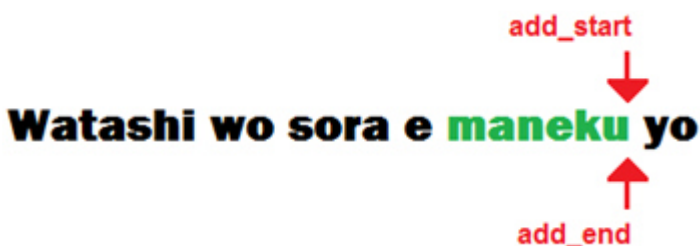
retime("word2end", add_start, add_end):

literalmente "desde la **palabra** hasta el final". El tiempo de inicio equivale al tiempo final de cada una de las palabras de cada línea karaoke ($\text{line.start_time} + \text{word.end_time}$) y el tiempo final, al tiempo final de cada una de las líneas karaoke (line.end_time). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Char, Translation Word y Translation Char.

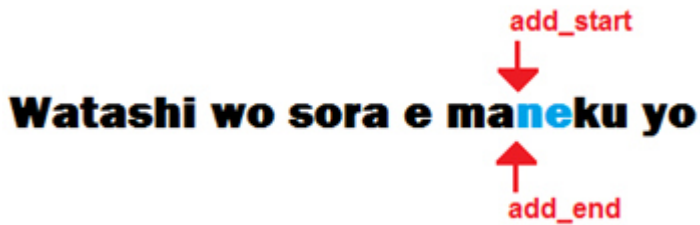


retime("postword", add_start, add_end):

literalmente "después de la **palabra**". El tiempo de inicio equivale al tiempo final de cada una de las palabras en cada línea karaoke ($\text{line.start_time} + \text{word.end_time}$) y el tiempo final, también al tiempo final de cada una de las palabras ($\text{line.start_time} + \text{word.end_time}$). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Char, Translation Word y Translation Char.

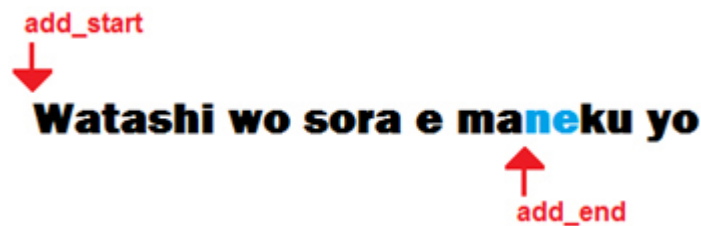


retime("presyl", add_start, add_end): literalmente "antes de la **sílaba**". El tiempo de inicio equivale al tiempo de inicio de cada una de las sílabas en cada línea karaoke ($\text{line.start_time} + \text{syl.start_time}$) y el tiempo final, también al tiempo de inicio de cada una de las sílabas en cada línea. Este **modo** está habilitado para los siguientes **Template Type**: Syl, Furi, Char y Translation Char.

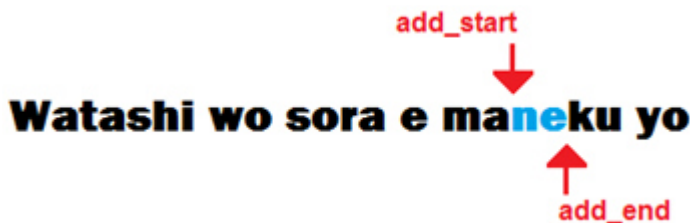


`retime("start2syl", add_start, add_end):`

literalmente “desde el inicio hasta la **sílaba**“. El tiempo de inicio equivale al tiempo de inicio de cada una de las líneas (`line.start_time`) y el tiempo final, al tiempo de inicio de cada **sílaba** (`line.start_time + syl.start_time`). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Furi, Char y Translation Char.

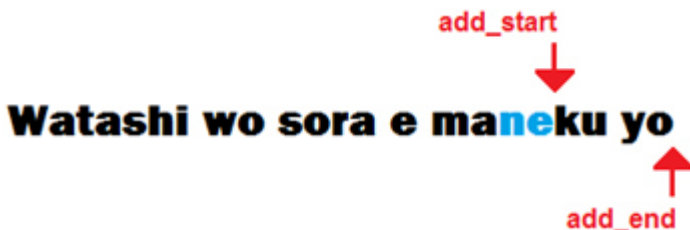


`retime("syl", add_start, add_end):` literalmente “**sílaba**“. El tiempo de inicio equivale al tiempo de inicio de cada una de las sílabas en cada línea karaoke (`line.start_time + syl.start_time`) y el tiempo final, al tiempo final de cada una de las sílabas (`line.start_time + syl.end_time`). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Furi, Char y Translation Char.



`retime("syl2end", add_start, add_end):`

literalmente “desde la **sílaba** hasta el final“. El tiempo de inicio equivale al tiempo final de cada una de las sílabas de cada línea karaoke (`line.start_time + syl.end_time`) y el tiempo final, al tiempo final de cada una de las líneas karaoke (`line.end_time`). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Furi, Char y Translation Char.



retime(“postsyl”, add_start, add_end):

literalmente “después de la **sílaba**“. El tiempo de inicio equivale al tiempo final de cada una de las sílabas en cada línea karaoke ($\text{line.start_time} + \text{syl.end_time}$) y el tiempo final, también al tiempo final de cada una de las sílabas ($\text{line.start_time} + \text{syl.end_time}$). Este **modo** está habilitado para los siguientes **Template Type**: Syl, Furi, Char y Translation Char.

Watashi wo sora e maneku yo



The diagram shows the word "Watashi wo sora e maneku yo" in a bold, black, sans-serif font. The word "maneku" is highlighted in blue. Above the word, a red arrow labeled "add_start" points down to the beginning of "maneku". Below the word, a red arrow labeled "add_end" points up to the end of "maneku".

retime(“prefuri”, add_start, add_end): literalmente “antes del **furigana**“. El tiempo de inicio equivale al tiempo de inicio de cada uno de los furiganas en cada línea karaoke ($\text{line.start_time} + \text{furi.start_time}$) y el tiempo final, también al tiempo de inicio de cada una de los furiganas en cada línea.

にほん なつ あつ
日本 の 夏 は 暑い です

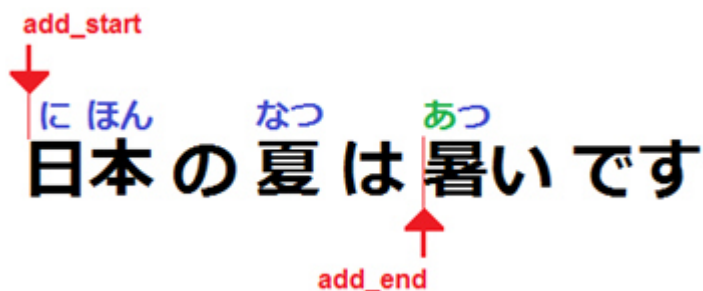


The diagram shows the Japanese sentence "日本 の 夏 は 暑い です" in a bold, black, sans-serif font. Above the sentence, the furigana "にほん", "なつ", and "あつ" are written in blue, green, and green respectively. Below the sentence, a red arrow labeled "add_start" points down to the beginning of "あつ". Below the sentence, a red arrow labeled "add_end" points up to the beginning of "あつ".

retime(“start2furi”, add_start, add_end):

literalmente “desde el inicio hasta el **furigana**“. El tiempo de inicio equivale al tiempo de inicio de cada una de las líneas (line.start_time) y el tiempo final, al tiempo de inicio de cada **furigana** ($\text{line.start_time} + \text{furi.start_time}$).

add_start
↓
にほん なつ あつ
日本 の 夏 は 暑い です
↑
add_end



The diagram shows the Japanese sentence "日本 の 夏 は 暑い です" in a bold, black, sans-serif font. Above the sentence, the furigana "にほん", "なつ", and "あつ" are written in blue, green, and green respectively. Below the sentence, a red arrow labeled "add_start" points down to the beginning of the first line. Below the sentence, a red arrow labeled "add_end" points up to the beginning of "あつ".

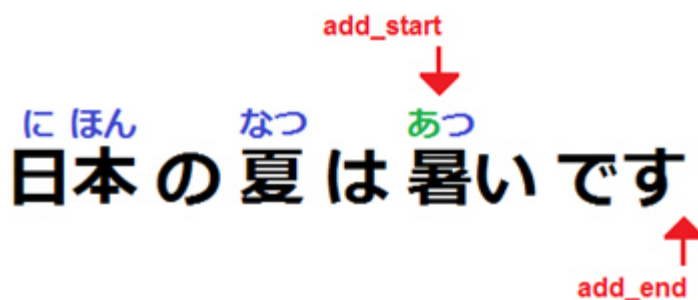
`retime("furi", add_start, add_end):`

literalmente “**furigana**“. El tiempo de inicio equivale al tiempo de inicio de cada uno de los furiganas en cada línea karaoke ($\text{line.start_time} + \text{furi.start_time}$) y el tiempo final, al tiempo final de cada uno de los furigana ($\text{line.start_time} + \text{furi.end_time}$).



`retime("furi2end", add_start, add_end):`

literalmente “desde el **furigana** hasta el final“. El tiempo de inicio equivale al tiempo final de cada uno de los furiganas de cada línea karaoke ($\text{line.start_time} + \text{furi.end_time}$) y el tiempo final, al tiempo final de cada una de las líneas karaoke (line.end_time).



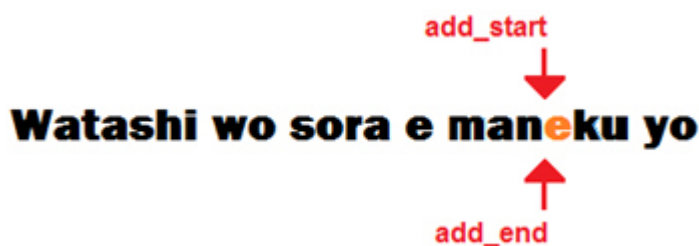
`retime("postfuri", add_start, add_end):`

literalmente “después del **furigana**“. El tiempo de inicio equivale al tiempo final de cada uno de los furiganas en cada línea karaoke ($\text{line.start_time} + \text{furi.end_time}$) y el tiempo final, también al tiempo final de cada uno de los furiganas ($\text{line.start_time} + \text{furi.end_time}$).



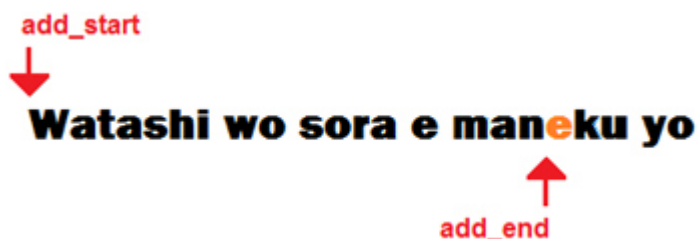
Olvidaba mencionar que los 5 modos furiganas solo se pueden aplicar al **Template Type: Furi**, ya que no tiene mucho sentido aplicarle este tipo de modos a los demás **Template Type**.

retime("prechar", add_start, add_end): literalmente "antes del **caracter**". El tiempo de inicio equivale al tiempo de inicio de cada uno de los caracteres en cada línea karaoke ($\text{line.start_time} + \text{char.start_time}$) y el tiempo final, también al tiempo de inicio de cada uno de los caracteres en cada línea. Este **modo** está habilitado para los siguientes **Template Type**: Char y Translation Char.

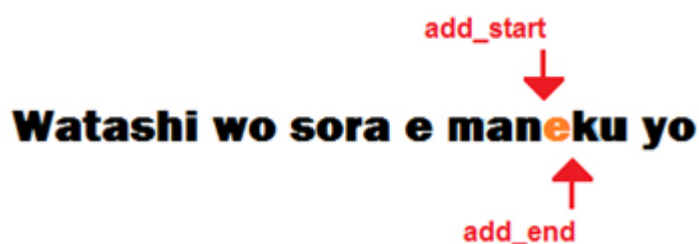


retime("start2char", add_start, add_end):

literalmente "desde el inicio hasta el **caracter**". El tiempo de inicio equivale al tiempo de inicio de cada una de las líneas (line.start_time) y el tiempo final, al tiempo de inicio de cada **caracter** ($\text{line.start_time} + \text{char.start_time}$). Este **modo** está habilitado para los siguientes **Template Type**: Char y Translation Char.

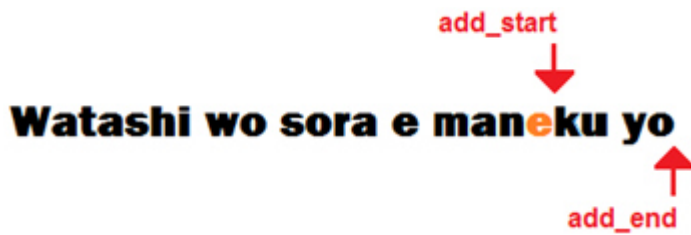


retime("char", add_start, add_end): literalmente "**caracter**". El tiempo de inicio equivale al tiempo de inicio de cada uno de los caracteres en cada línea karaoke ($\text{line.start_time} + \text{char.start_time}$) y el tiempo final, al tiempo final de cada uno de los caracteres ($\text{line.start_time} + \text{furi.end_time}$). Este **modo** está habilitado para los siguientes **Template Type**: Char y Translation Char.



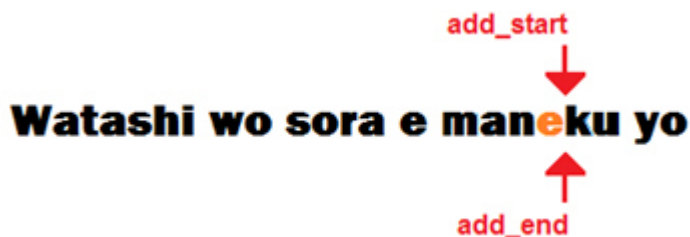
retime("furi2end", add_start, add_end):

literalmente “desde el **furigana** hasta el final”. El tiempo de inicio equivale al tiempo final de cada uno de los furiganas de cada línea karaoke (line.start_time + furi.end_time) y el tiempo final, al tiempo final de cada una de las líneas karaoke (line.end_time). Este **modo** está habilitado para los siguientes **Template Type**: Char y Translation Char.



retime("postchar", add_start, add_end):

literalmente “después del **furigana**“. El tiempo de inicio equivale al tiempo final de cada uno de los furiganas en cada línea karaoke (line.start_time + furi.end_time) y el tiempo final, también al tiempo final de cada uno de los furiganas (line.start_time + furi.end_time). Este **modo** está habilitado para los siguientes **Template Type**: Char y Translation Char.

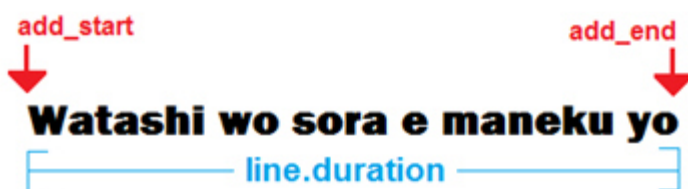


A continuación veremos los 5 modos “pct” de la función **retime**, que son: “linepct”, “wordpct”, “sylvpct”, “furipct” y “charpct”.

“pct” es la sigla de “percent” (porcentaje en español). El 0% equivale al inicio del modo y el 100% al tiempo final, o sea, al total de la duración.

retime("linepct", 0, 100):

usada con estos dos valores (0 y 100) es lo mismo que el modo “line”.



La diferencia real es cuando se usan otros valores para que se adicionen o resten del tiempo inicial de la línea.

Ejemplo:

- `line.start_time = 1200`
- `line.duration = 3600 ms`
- `line.end_time = 4800`
- `retime("linepct", 0, 50)`

0% de $3600 = 0$

50% de $3600 = 1800$, entonces:

$1200 + 0 = 1200 \leftarrow$ sería el tiempo de inicio del re-timeo

$1200 + 1800 = 3000 \leftarrow$ sería el tiempo final del re-timeo

De manera muy similar funcionan los otros 4 modos "pct".

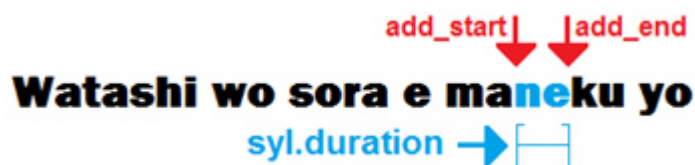
`retime("wordpct", 0, 100):`

usada con estos dos valores (0 y 100) es lo mismo que el modo "word".



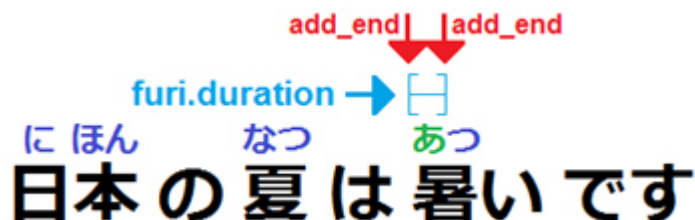
`retime("sylpct", 0, 100):`

usada con estos dos valores (0 y 100) es lo mismo que el modo "syl".



`retime("furipct", 0, 100):`

usada con estos dos valores (0 y 100) es lo mismo que el modo "furi".



retime("charpct", 0, 100):

usada con estos dos valores (0 y 100) es lo mismo que el modo "char".

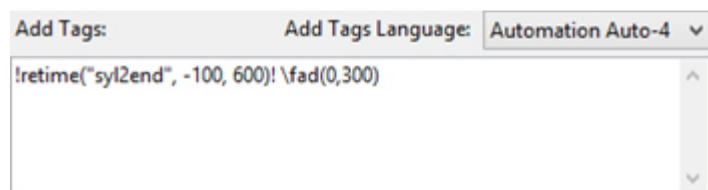
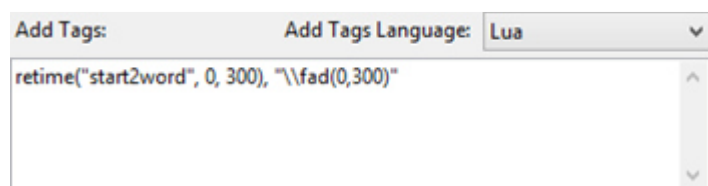


Y por último, está el modo "set" o "abs", que es lo mismo. Y hace referencia al tiempo absoluto con respecto al cero del vídeo, ejemplo:

retime("set", 1200, 23100): entonces el tiempo de inicio del re-timeo será 1200 ms y el tiempo final será 23100 ms.

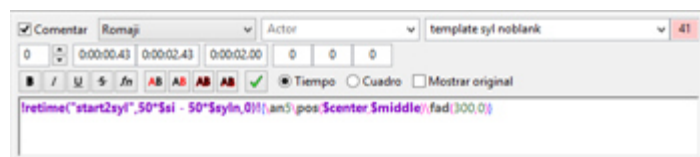
No pareciera ser sencillo memorizar los **29 modos** de la función **retime**, pero es solo cuestión de práctica y de un poco de sentido común dado el nombre de cada uno de los modos. Y como ya lo había mencionado, esta función es de gran utilidad y más para aquellos que ya la han usado en **Automation Auto-4** y están familiarizados con los 10 modos por default del **Aegisub**.

Estos son dos cortos ejemplos de cómo usar la función **retime** dependiendo del lenguaje que más nos guste, **LUA** o **Automation Auto-4**:



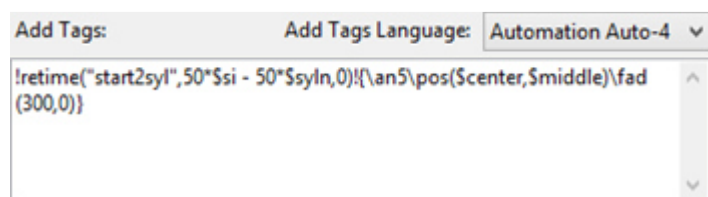
Y a manera de práctica, pueden usar el resto de los modos para que sepan qué uso darle acada uno de ellos y crear efectos de entrada (**lead-in**), de sílaba activa (**hi-light**), de salida (**lead-out**) o la combinación que deseen. Las posibilidades son muchas y los efectos también.

El **Kara Effector** también nos da la posibilidad de pegar en él algún **Template** copiado desde el **Aegisub**. Ejemplo:



Inicio	Final	Estilo	Efecto	Texto
0:00:00.43	0:00:02.43	Romaji	template syl noblank	!retime("start2syl", 50*\$si - 50*\$eyln, 0)!*
0:00:02.43	0:00:08.16	Romaji		*Ko*do*ku *na *ho*ho *ko *nu*ra*su *nu*ra*su *ke*do
0:00:08.33	0:00:13.19	Romaji		*Yo*ka*ke *no *ke*ka*ki *ga *shi*zu*ka *ni *mi*chi*te

En el caso de que nos guste este Template en particular, que de casualidad es de un Efecto que nos gusta, entonces lo copiamos tal cual y lo pegamos en el **Kara Effector** usando el lenguaje **Automation Auto-4**, así:



Y al aplicar el Efecto, el resultado será el mismo que en el **Aegisub**, lo que implica una gran ventaja desde donde se lo mire.

Con este artículo se despide con la recomendación de poner en práctica los ejemplos vistos en él. No olviden descargar la más reciente actualización disponible del **Kara Effector** y visitarnos en la web principal lo mismo que en los canales de **YouTube** para descargar los nuevos Efectos o dejar algún comentario, exponer alguna duda o hacer alguna sugerencia. Muchas gracias por seguirnos.

En este artículo “[Función retime – Nuevos Modos \[fx\]](#)” se describe una actualización de 4 nuevos modos de la función retime. Para un total de 33 modos disponibles en Kara Effector.