

# Kara Effector 3.2:

## Effector Book

### Vol. II [Tomo 30]

# Kara Effector 3.2:

En este **Tomo 30** continuaremos viendo las funciones de la **librería effector**. Esta librería contiene una serie de funciones interesantes que espero que con la ayuda de esta documentación, le puedan sacar el máximo provecho a la hora de llevar a cabo sus proyectos, no solo karaokes, sino también en la edición de los subtítulos.

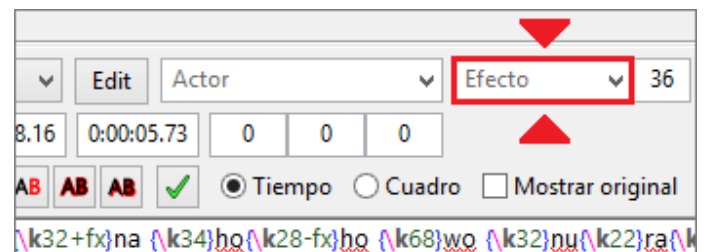
## Librería Effector [KE]:

1. **effector.pos**
2. **effector.knj**
3. **effector.offset\_pos**
4. **effector.import**
5. **effector.addfx**
6. **effector.savefx**
7. **effector.modify\_pos**
8. **effector.new\_pos**
9. **effector.default\_val**
10. **effector.effect\_offset**
11. **effector.decide**
12. **effector.print\_error**
13. **effector.run\_fx**
14. **effector.preprocesses\_styles**
15. **effector.preprocesses\_macro**
16. **effector.preprocesses\_lines**
17. **effector.macro\_fx**

De las anteriores funciones, las que están en negro son las funciones administrativas, y las que están en azul son las que de una u otra manera le podemos sacar provecho.

» **effector.effect\_offset( )**

Es una función interna del **KE** que retorna una **tabla** con a lo más 32 valores numéricos ingresados en la celda "efecto" del **Aegisub**:



Los valores se ingresan separados por coma (,) o por punto y coma (;), así, Ejemplo:



En las líneas se verá algo como esto:

Final	Estilo	Efecto	Texto
0:00:08.16	Romaji		*ko*do*ku *na *ho*ho *
0:00:13.19	Romaji		*yo*a*ke *no *ke*ha*ri *
0:00:18.39	Romaji	10, 24, 11.4, 86	*wa*ta*shi *wo *so*ra *
0:00:27.22	Romaji		*ki*bo*ku *ga *ka*na*ta *
0:00:34.30	Romaji		*ma*yo*ki *na*ga*ra *
0:00:39.31	Romaji		*su*re*chi*ga*ku *i*shi*ki *
0:00:45.79	Romaji		*tsu*ka*ma*te*ru *yo *shi*ka *
0:00:54.56	Romaji		*mo*to*me *a*ku *ko*ko*ro *
0:00:08.16	Hiragana		*こ*ど*く *な *ほ*ほ *
0:00:13.19	Hiragana		*よ*あ*け *の *け*は*り *

La **tabla** en la que se alojan estos valores se llama:

**effect\_val = { }**

Los valores por default de cada uno de los 32 valores numéricos de la **tabla effect\_val** es cero (0). Entonces los cuatro primeros valores, dependiendo de la línea, para este ejemplo, serían:

- Línea 1: **effect\_val{ 0, 0, 0, 0 }**
- Línea 2: **effect\_val{ 0, 0, 0, 0 }**
- Línea 3: **effect\_val{ 10, 24, 11.4, 86 }**
- Línea 5: **effect\_val{ 0, 0, 0, 0 }**
- Línea 6: **effect\_val{ 0, 0, 0, 0 }**

Y así sucesivamente para el resto de las líneas, es decir que si no ponemos nada en la celda “efecto” de una línea en el script, todos los 32 valores de la tabla **effect\_val** serán cero para esa línea.

Para ingresar diferentes valores por línea en la tabla, lo que debemos hacer es algo como esto:

Estilo	Efecto	Texto
Romaji		*ko*do*ku *na *ho*ho *wo *nu*ra *
Romaji	5, 25, 25	*yo*a*ke *no *ke*ha*ri *ga *shi*zu *
Romaji	10, 24, 11.4, 86	*wa*ta*shi *wo *so*ra *e *ma*ne *
Romaji	4, 8, 12, 16	*ki*bo*ku *ga *ka*na*ta *de *ma*ta *
Romaji		*ma*yo*ki *na*ga*ra *mo *ki*mi *w *
Romaji		*su*re*chi*ga*ku *i*shi*ki *te *ga *
Romaji		*tsu*ka*ma*te*ru *yo *shi*ka *ka *ri *
Romaji		*mo*to*me *a*ku *ko*ko*ro *so *re *
Hiragana		*こ*ど*く *な *ほ*ほ *を *ぬ*ら*す *

Para el ejemplo anterior, la **tabla** en la segunda línea tendrá los siguientes valores:

- **effect\_val[1] = 5**
- **effect\_val[2] = 25**
- **effect\_val[3] = 25**

o sea que podemos acceder a cualquier valor de la tabla tan solo colocando el nombre de la tabla (**effect\_val**) y el índice de su posición, desde 1 hasta el 32, sabiendo que el valor por default de cada uno de ellos es cero (0). Ejemplo:

- **effect\_val[24] = 0**

Solo para a los tres primeros valores de la **tabla**, podemos acceder a dichos valores con otro nombre opcional son los siguientes:

- **fx.offset\_x**
- **fx.offset\_y**
- **fx.offset\_z**

Estilo	Efecto	Texto
Romaji		*ko*do*ku *na *ho*ho *wo *nu*ra *
Romaji	5, 25, 25	*yo*a*ke *no *ke*ha*ri *ga *shi*zu *
Romaji	10, 24, 11.4, 86	*wa*ta*shi *wo *so*ra *e *ma*ne *
Romaji	4, 8, 12, 16	*ki*bo*ku *ga *ka*na*ta *de *ma*ta *
Romaji		
Romaji		
Romaji		
Romaji		
Hiragana		

fx.offset\_x = effect\_val[1]  
 fx.offset\_y = effect\_val[2]  
 fx.offset\_z = effect\_val[3]

Cualquiera de los dos nombres para estos tres primeros valores numéricos es válido para poder usarlos en un efecto en la celda de texto de la Ventana de Modificación del **KE**, que necesitamos.

#### Ejemplo:

Un ejemplo sencillo es adicionar tiempo a una o más líneas seleccionadas:

Final	Estilo	Efecto	Texto
0:00:08.16	Romaji		*ko*do*ku *na *ho*ho *wo *nu*ra *
0:00:13.19	Romaji	1000	*yo*a*ke *no *ke*ha*ri *ga *shi*zu *
0:00:18.39	Romaji		*wa*ta*shi *wo *so*ra *e *ma*ne *
0:00:27.22	Romaji		*ki*bo*ku *ga *ka*na*ta *de *ma*ta *
0:00:34.30	Romaji		*ma*yo*ki *na*ga*ra *mo *ki*mi *w *
0:00:39.31	Romaji		*su*re*chi*ga*ku *i*shi*ki *te *ga *
0:00:45.79	Romaji		*tsu*ka*ma*te*ru *yo *shi*ka *ka *ri *
0:00:54.56	Romaji		*mo*to*me *a*ku *ko*ko*ro *so *re *
0:00:08.16	Hiragana		*こ*ど*く *な *ほ*ほ *を *ぬ*ら*す *

En la imagen anterior, colocamos 1000 en la celda “efecto” de la segunda línea de karaoke, y lo que haremos en el efecto seleccionado será adicionar ese valor como tiempo extra en **Line End Time**:

Template Type [fx]: Syl

Line Start Time = l.start\_time

Line End Time = l.end\_time + effect\_val[1]

Entonces lo que la función hará será adicionar 1000 ms al tiempo final de la segunda línea y 0 ms al resto de ellas ya que ese es el valor por default de **effect\_val[1]**.

Lo que podemos concluir como la aplicación de esta función es el poder guardar valores diferentes e independientes en las líneas del script que selecciones y poderlos usar a nuestra conveniencia en los efectos a aplicar.

Como ya les había mencionado antes, el resto de las funciones de la librería **effector** son funciones consideradas como administrativas, ya que no podemos usarlas en nuestros efectos de manera directa, sino que lo hacen de forma interna y hacen que el KE funcione de la manera que se supone que tiene que hacerlo. A continuación daré una corta descripción de cada una de ellas:

- **effector.pos**: es la función que hace que los objetos karaokes queden posicionados en el vídeo luego de aplicar nuestros efectos.
- **effector.knj**: es la función que permite ubicar de forma vertical a nuestros subtítulos, en especial los kanjis, hiraganas y katakanas.
- **effector.modify\_pos**: esta función calcula cambios hechos en las celdas de posición de la Ventana de Modificación del KE y hace los ajustes necesarios para retornar el tag indicado. Las opciones de tags de posición que retornan las celdas de posición son:
  - \pos
  - \move
  - \moves3
  - \moves4
  - \mover
- **effector.new\_pos**: esta función está incluida en algunas funciones del KE para redefinir los valores de posición, ejemplo la función **shape.Rmove**
- **effector.default\_val**: esta función provee valores por default a las celdas de texto de la Ventana de Modificación del KE, con el fin de evitar un error en el caso de cometer un error sencillo.
- **effector.print\_error**: esta función está incluida en casi todas las funciones del KE y es una alerta temprana de qué puede estar mal en el caso de cometer un error en un efecto. La función hace que veamos un mensaje indicando el parámetro de la función en el que está el error y nos dice qué tipo de valor deber ir en dicho parámetro.
- **effector.run\_fx**: es la función que hace que se apliquen los efectos seleccionados.
- **effector.preprosses\_styles**: esta función hace que veamos en lista a todos los estilos disponibles para aplicar efectos de los que están en el script.
- **effector.preprosses\_macro**: esta función prepara la información que contienen las dos ventanas del KE y la organiza en las respectivas celdas.
- **effector.preprosses\_lines**: esta función extrae toda la información de las líneas del script, estén seleccionadas o no, y guarda la información de cada una de ellas en tablas independientes para que sea usada en cualquier momento de un efecto.
- **effector.macro\_fx**: es la función que hace posible la interfaz gráfica del KE, es la que diseña la posición y ubicación de todos los elementos en las dos ventanas de diálogos del Kara Effector.

## Librería Aegisub [KE]:

Es la librería del KE que contiene las funciones que hacen posible los diferentes modos de aplicar los efectos según las divisiones del texto en cada línea del script. Esta funciones hacen posible aplicar efectos tipo Syl, Word, Char y las demás opciones que hay **Template Type**.

La mayoría de las funciones de esta librería no son para hacer efectos directamente, son administrativas, y solo dos de ellas las podemos usar para obtener información de un string de texto asignado por nosotros mismos:

1. **aegisub.word( line\_text, dur, i )**
2. **aegisub.word2s( )**
3. **aegisub.wordsi( words )**
4. **aegisub.syls2c( )**
5. **aegisub.line2W( )**
6. **aegisub.line2S( )**
7. **aegisub.line2C( )**
8. **aegisub.word2S( )**
9. **aegisub.word2C( )**
10. **aegisub.word2c( )**
11. **aegisub.wordci( words )**
12. **aegisub.width( string\_txt )**
13. **aegisub.height( string\_txt )**

### » aegisub.width( string\_txt )

Esta función calcula en ancho medido en pixeles de un string de texto ingresado. Los cálculos que la función hace están basados en la información del estilo de la línea:

Fuente	
LilyUPC	62
Escala X%: 100	Escala Y%: 120
Espaciado: 1	

Los anteriores cinco datos del estilo de una línea, son los que determinan las dimensiones de un string de texto, lo que quiere decir estas dimensiones son relativas al estilo.

### » Ejemplo:

**aegisub.width( string.char( R(50, 97) )**

Entonces la función retorna la medida en pixeles del ancho del carácter al azar que retorne **string.char**

**aegisub.width( syl.text ) = syl.width**

**aegisub.width( word.text ) = word.width**

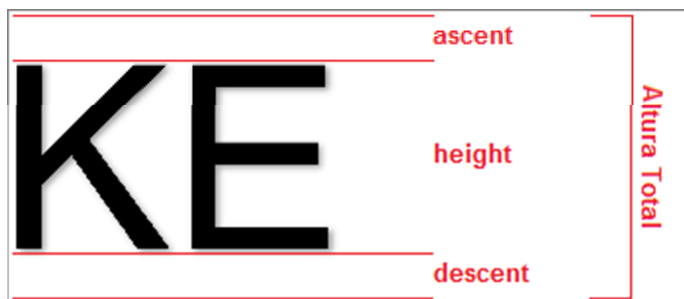
**aegisub.width( "KE" ) → relativo al estilo**

## » aegisub.height( string\_txt )

Esta función es similar a la anterior, pero con la diferencia de que retorna el alto de un string de texto en pixeles.

### > Ejemplo:

```
aegisub.height( syl.text ) = syl.height
aegisub.height( word.text ) = word.height
aegisub.height( "KE" ) → relativo al estilo
```



## Recursos [KE]:

He denominado al próximo tema como **Recursos del KE**, y son una serie de funciones, variables y múltiples valores y herramientas extras, con las que también podemos contar para ayudarnos en nuestros proyectos.

### » Recursos [KE]:

función

### » set\_temp( ref, val )

Esta función es una creación del maestro **Pyointa**, cuya finalidad principal era el poder almacenar el valor de una variable definida con valor aleatorio, para hacer efectos con templates en **Automation Auto-4**. La función almacena a una variable en una **tabla** específica, para ser usada posteriormente. La **tabla** en donde son almacenadas las variables se llama:

```
temp = { }
```

El parámetro "**ref**" es el nombre de la variable que vamos a crear y el parámetro "**val**" es el valor que le daremos a dicha variable. El argumento "**ref**" debe ir entre comillas, ya sean dobles o sencillas.

### > Ejemplo:

```
set_temp( "sizexy", R(80,120) )
→ temp.sizexy = R(80,120)
→ temp = { sizexy = R(80,120) }
```

### > Ejemplo:

Add Tags Language: » Automation Auto-4

```
!fscx!set_temp( "xy", R(80,120) )!fscy!temp.xy!
```

### > Ejemplo:

Add Tags Language: » Lua

```
format("\\fscx%s\\fscy%s", set_temp("xy", R(80,120)), temp.xy)
```

Esta función es equivalente a declarar una función en la celda de texto "**Variable**" y luego llamarla la cantidad de veces que sea necesaria. Este hecho hace que la función **set\_temp** sea prácticamente obsoleta en el **Kara Effector**, pero aun así está incluida dentro de él, porque hay muchos que la usan y ya están familiarizados con ella.

### » Recursos [KE]:

función

### » remember( ref, val )

Es la función equivalente a la función **set\_temp** que **Aegisub** estrenó en su versión **2.1.9** y la finalidad de esta es exactamente la misma, con la ligera diferencia de que ésta almacena las variables en una **tabla** llamada:

```
recall = { }
```

### > Ejemplo:

```
remember( "angle", R(360) )
→ recall.angle = R(360)
→ recall = { angle = R(360) }
```

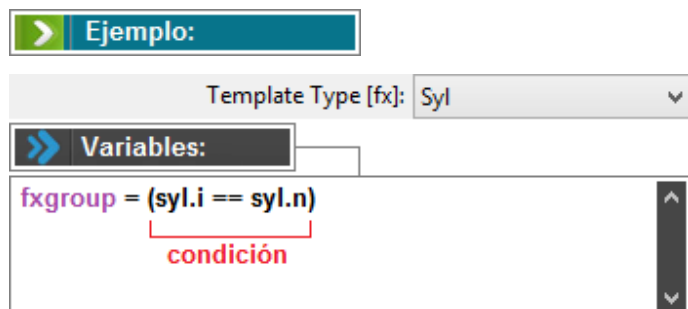
### » Recursos [KE]:

variable

```
fxgroup = false or true
```

Es una variable booleana a la que solo le podemos asignar los valores de falso o verdadero (false o true), usada en la celda de texto "**Variables**".

Esta variable es una función original del **Aegisub** que ha sido adaptada al **KE**, que restringe por medio de una o más condiciones dadas, los objetos a los cuales se les aplicará un efecto seleccionado.



La condición con la que definimos a la variable **fxgroup** es de igualdad, que es una condición de comparación, y solo es verdadera cuando **syl.i** es igual a la última sílaba de cada línea seleccionada para aplicar el efecto.

Entonces, cualquiera que haya sido el efecto aplicado, éste solo se aplicará a la última sílaba y no tendrá en cuenta al resto de ellas.

Recordemos las condiciones de comparación:

- == → igual a
- ~= → diferente a
- > → mayor que
- < → menor que
- >= → mayor o igual que
- <= → menor o igual que

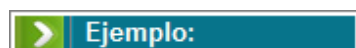
También tenemos a la disyunción (**or**) y la conjunción (**and**), que también arrojan un valor booleano entre falso (false) o verdadero (true), al comparar el valor de dos condiciones:

Disyunción (**or**):

- true or true → true
- true or false → true
- false or true → true
- false or false → false

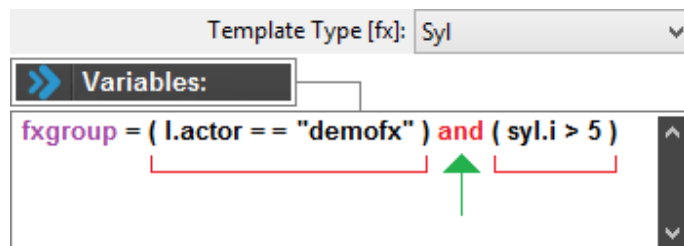
Conjunción (**and**):

- true and true → true
- true and false → false
- false and true → false
- false and false → false



Estilo	Actor	Texto
Romaji		*ko*do*ku *na *ho*ho *wo *nu*ra*su *nu
Romaji	demofx	*yo*a*ke *no *ke*ha*ni *ga *shi*zu*ka *ni
Romaji		*wa*ta*shi *wo *so*ra *e *ma*ne*ku *yo
Romaji		*ki*bo*ku *ga *ka*na*ta *de *ma*te*te*ru
Romaji		*ma*yo*ki *na*ga*ra *mo *ki*mi *wo *sa*ga
Romaji		*su*re*chi*ga*ku *ni *shi*ki *te *ga *fu*re*

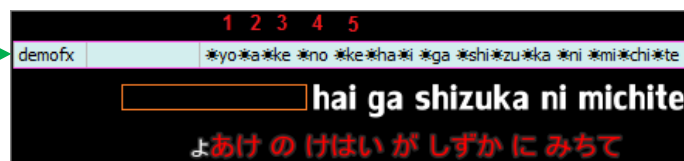
Para este ejemplo, ponemos un nombre cualquiera a una o más líneas del script (no todas para que se note el fxgroup) en la celda "actor" del **Aegisub**, como se nota en la imagen anterior.



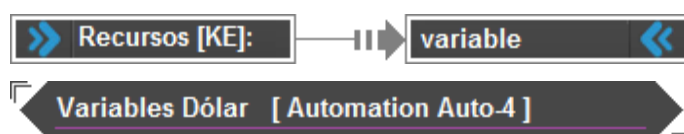
La primera condición es verdadera (true) solo para aquellas líneas que en la celda "actor" ponga: demofx

La segunda condición es verdadera (true) solo cuando las sílabas de las líneas son mayores que 5.

Y como las dos condiciones están relacionadas con una Conjunción (**and**), la variable **fxgroup** será verdadera solo cuando las dos condiciones sean verdadera. Es decir que al aplicar un efecto, éste solo tomará en cuenta a las sílabas mayores que 5 que pertenezcan a las líneas que tengan como actor a "demofx":

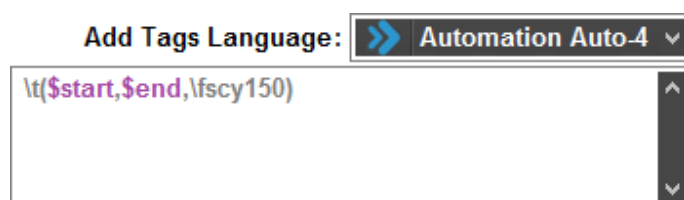


El rectángulo naranja resalta el hecho que el efecto aplicado no tuvo en cuenta a las sílabas "yo", "a", "ke", "no" y "ke", porque a pesar de que éstas pertenecen a una línea con actor "demofx", no cumplen con la segunda condición de que el número de su posición sea mayor que 5.



Son una serie de variables inspiradas en las variables Dólar del **Automation Auto-4** del **Aegisub**, y sirven para acceder a valores concernientes a las líneas y a las sílabas de las líneas de texto, pero acá en el **KE** he querido ampliar ese listado de variables para que abarquen a todos los modos de efectos dependiendo del **Template Type**.

La forma en la que el **KE** reconoce a dichas variables es cambiando el lenguaje de **Add Tags**. Ejemplo:



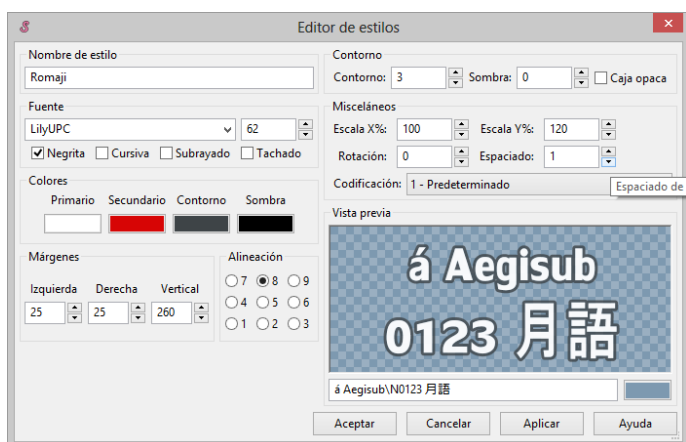


El primer grupo de variables dólar son las referentes a los estilos de cada una de las líneas del script:

#### variables Dólar de Estilos

\$layer	\$margin_b	\$fontsize	\$color4
\$style	\$scale_x	\$fontname	\$alpha1
\$actor	\$scale_y	\$spacing	\$alpha2
\$margin_l	\$angle	\$color1	\$alpha3
\$margin_r	\$outline	\$color2	\$alpha4
\$margin_v	\$shadow	\$color3	\$align
\$margin_t			

**\$layer**, **\$style** y **\$actor** son variables de las líneas del script y el resto son las variables de todos los valores de los estilos de las líneas:



Luego tenemos las variables de tiempo:

#### variables Dólar de Tiempo

\$lstart	\$wstart	\$sstart	\$fstart
\$cstart	\$fxstart	\$start	
\$lend	\$wend	\$send	\$fend
\$cend	\$fxend	\$end	
\$lmid	\$wmid	\$smid	\$fmid
\$cmid	\$fxmid	\$mid	
\$lkdur	\$wkdur	\$skdur	\$fkdur
\$ckdur	\$fxkdur	\$kdur	
\$ldur	\$wdur	\$sdur	\$fdur
\$cdur	\$fxdur	\$dur	

- **\$lstart**: tiempo de inicio de la línea medido en ms, relativo al cero absoluto del vídeo.
- **\$wstart**: tiempo de inicio de cada palabra medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$sstart**: tiempo de inicio de cada sílaba medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.

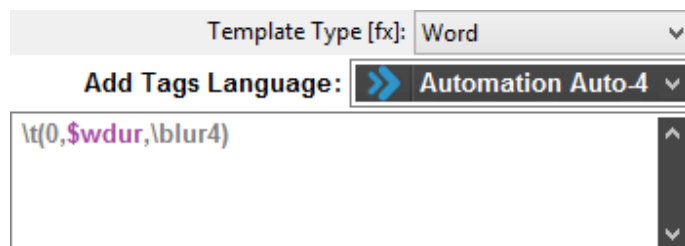
- **\$fstart**: tiempo de inicio de cada **furigana** medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$cstart**: tiempo de inicio de cada carácter medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$fxstart**: es el tiempo de inicio del objeto karaoke por default dependiendo del **Template Type**. O sea que si aplicamos un karaoke con un modo Word, la variable **\$fxstart** tomará el valor de **\$wstart**.
- **\$start**: esta variable tiene el mismo valor de **\$fxstart**, es decir que ambas variables son la forma general de obtener el tiempo de inicio del objeto karaoke, dependiendo del **Template Type**.

Estas dos últimas variables las vemos en un recuadro rojo en la tabla de las variables, este recuadro indica las variables generales de cada categoría de variables.

En esta misma categoría de variables dólar de tiempo están las de tiempo final y tiempo medio y de duración total de los objetos karaokes:

- **\$lend**: tiempo final de la línea. Es equivalente a **l.end\_time**
- **\$wmid**: tiempo medio de la palabra. Es equivalente a **word.mid\_time**
- **\$skdur**: duración de la sílaba en centésimas de segundos. Es equivalente a **syl.dur / 10**
- **\$cdur**: duración del carácter medido en ms. Es equivalente a **char.dur**

#### Ejemplo:



#### variables Dólar de Contadores

\$li	\$wi	\$si	\$fi
\$ci	\$fxi	\$i	
\$linen	\$wordn	\$slyn	\$furin
\$charn	\$fxn	\$n	

Las variables de contadores nos dan dos valores, el índice de la posición y la cantidad de objetos karaoke.

- **\$li**: es el contador de las líneas seleccionadas para aplicarle un efecto, equivalente a **line.i**
- **\$wi**: es el contador de las palabras por línea, y es equivalente a **word.i**



- **\$si:** es el contador de las sílabas por línea, y es equivalente a **syl.i**
- **\$fi:** es el contador de los furiganas por línea, y es equivalente a **furi.i**
- **\$ci:** es el contador de los caracteres por línea, y es equivalente a **char.i**
- **\$fxi:** es el contador de los objetos karaokes en un efecto, dependiendo del **Template Type**.
- **\$i:** es lo mismo que **\$fxi**

El resto de las variables de esa categoría indican la cantidad de objetos karaokes, en el caso de **\$linen**, ésta indica la cantidad total de líneas seleccionadas para aplicar un efecto, y las demás indican la cantidad de dicho objeto karaoke en cada línea seleccionada. Ejemplo:

- **\$slyn:** indica la cantidad total de sílabas en cada línea, de las seleccionadas para un efecto.

#### variables Dólar de Distancias Notables

\$lleft	\$wleft	\$sleft	\$fleft
\$cleft	\$fxleft	\$left	
\$lright	\$wright	\$sright	\$fright
\$cright	\$fxright	\$right	
\$ltop	\$wtop	\$stop	\$ftop
\$ctop	\$fxtop	\$top	
\$lbottom	\$wbottom	\$sbottom	\$fbottom
\$cbottom	\$fxbottom	\$bottom	
\$lcenter	\$wcenter	\$scenter	\$fcenter
\$ccenter	\$fxcenter	\$center	
\$lx	\$wx	\$sx	\$fx
\$cx	\$fxx	\$x	
\$lmiddle	\$wmiddle	\$smiddle	\$fmiddle
\$cmiddle	\$fxmiddle	\$middle	
\$ly	\$wy	\$sy	\$fy
\$cy	\$fxy	\$y	

Esta categoría es la de las variables de distancias notables de los objetos karaokes. Son seis diferentes:

- **left (izquierda)**
- **center (centro)**
- **right (derecha)**
- **top (superior)**
- **middle (medio)**
- **bottom (inferior)**

También tenemos dos equivalencias respectivas para las variables de centro y medio, que son \$x y \$y para cada uno de los objetos karaokes. Ejemplo

- **\$wcenter = \$wx**

#### variables Dólar de Medidas Notables

\$lwidth	\$wwidth	\$swidth	\$fwidth
\$cwidth	\$fxwidth	\$width	
\$lheight	\$wheight	\$sheight	\$fheight
\$cheight	\$fxheight	\$height	

Esta última categoría de variables dólar, es de las medidas notables de los objetos karaoke. Dichos valores vienen medidos en pixeles y equivalen al ancho y al alto.

- **\$lwidth = l.width**
- **\$wwidth = word.width**
- **\$swidth = syl.width**
- **\$fwidth = furi.width**
- **\$cwidth = char.width**
- **\$fxwidth = \$width**
- **\$lheight = l. height**
- **\$wheight = word. height**
- **\$sheight = syl. height**
- **\$fheight = furi. height**
- **\$cheight = char. height**
- **\$fxheight = \$height**

Podemos concluir que no hay una variable dólar de la cual no hayamos conocido su valor con anterioridad, pero como algunos de los usuarios del **KE** antes hacían efectos con los templates del **Aegisub**, entonces están más familiarizados con este tipo de notación de las variables y se les facilita seguir creando efectos en el **Kara Effector** con el lenguaje **Automation Auto-4**

Estas variables son algunas de las herramientas habilitadas para servirnos de apoyo a la hora de hacer efectos en lenguaje **Automation Auto-4**, pero no son las únicas, en el siguiente Tomo veremos unas cuantas más.

Es todo por ahora para el **Tomo 30** del **KE**. Intenten poner en práctica todos los ejemplos vistos y no olviden descargar la última actualización disponible del **Kara Effector 3.2** y visitarnos en el **Blog Oficial**, lo mismo que en los canales de **YouTube** para descargar los nuevos Efectos o dejar algún comentario. Pueden visitarnos y dejar su comentario en nuestra página de **Facebook**:

- [www.karaeffector.blogspot.com](http://www.karaeffector.blogspot.com)
- [www.facebook.com/karaeffector](http://www.facebook.com/karaeffector)
- [www.youtube.com/user/victor8607](http://www.youtube.com/user/victor8607)
- [www.youtube.com/user/NatsuoKE](http://www.youtube.com/user/NatsuoKE)
- [www.youtube.com/user/karalaura2012](http://www.youtube.com/user/karalaura2012)