

Librería Effector [KE]

Esta librería contiene un listado de funciones especiales, las cuales la mayoría son “**funciones administrativas**”, ya que son funciones internas del **KE** que no generan efectos, sino que hacen que él realice su tarea de forma correcta.

No vale la pena el entrar en detalle con las anteriormente mencionadas funciones administrativas, pero al igual las mencionaré. Por otra parte, hay unas cuantas funciones en esta librería que sí nos pueden ser de gran utilidad.

El listado de funciones que contiene la **librería effector** es el siguiente:

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.offset_pos() :

Es una función interna del **KE** que hace que las líneas de efecto generadas queden posicionadas de forma relativa a una posición previa hecha a las líneas del script en el vídeo.

Ejemplo:

Lo que debemos hacer es seleccionar la o las líneas que queremos reposicionar en el vídeo, para ello usaremos la herramienta para arrastrar los subtítulos del **Aegisub** que está en la esquina superior izquierda, al lado del vídeo:



El procedimiento es simple, ya que una vez que pulsamos el botón para arrastrar los subtítulos, a cada elemento visible en el vídeo le sale un cuadrado con una cruz al centro que indica la posición y alineación del mismo, y con el clic derecho del mouse en él elegimos la nueva posición:



Una vez que hayamos desplazado la línea seleccionada a una nueva posición, la función **effector.offset_pos** hará internamente que esas coordenadas sean el punto de referencia al aplicar un efecto. Ejemplo:



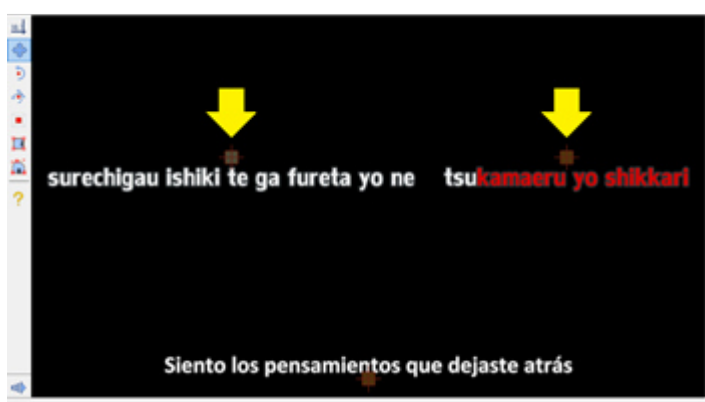
Esta habilidad interna del **KE** es ideal, por ejemplo, para aquellas líneas que coinciden en tiempo y se superponen en el vídeo:

Ejemplo:

Las líneas tienen el mismo centro, y al coincidir en tiempo, se superponen:



Entonces podemos mover las líneas de forma que ya no coincidan en su posición:



Y una vez que ya le hemos dado nuevas posiciones a las líneas en conflicto, ya podemos aplicar un efecto sin el temor de que las líneas de fx vayan a quedar superpuestas:



En resumen hasta acá, esta función hace que el **KE** redefina los centros que por default ya tiene cada línea del script dados por los valores de posición y alineación en el estilo de cada línea.

Otra habilidad que tiene esta función interna del **KE** es poder reposicionar a las líneas de karaoke previamente con la herramienta de rotación en los estilos de las mismas:

Ejemplo:



Y en vídeo veremos algo como esto:



Y al aplicar los efectos, el texto completo adoptará el ángulo de inclinación puesto en los estilos:



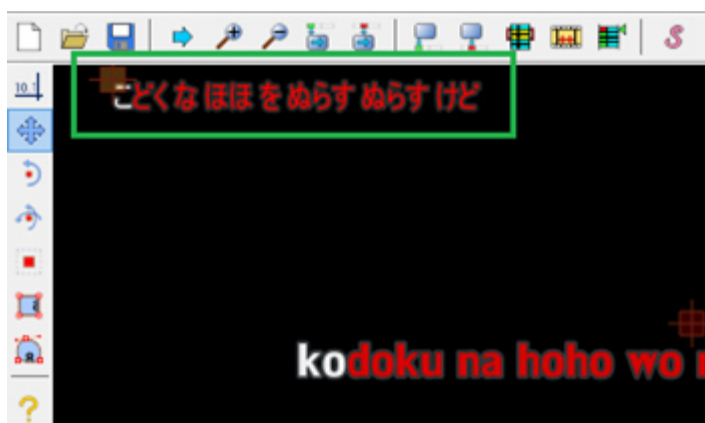
Una vez seleccionado el ángulo de inclinación de las líneas, podemos también modificar la posición en el vídeo con la herramienta de arrastre de subtítulos:



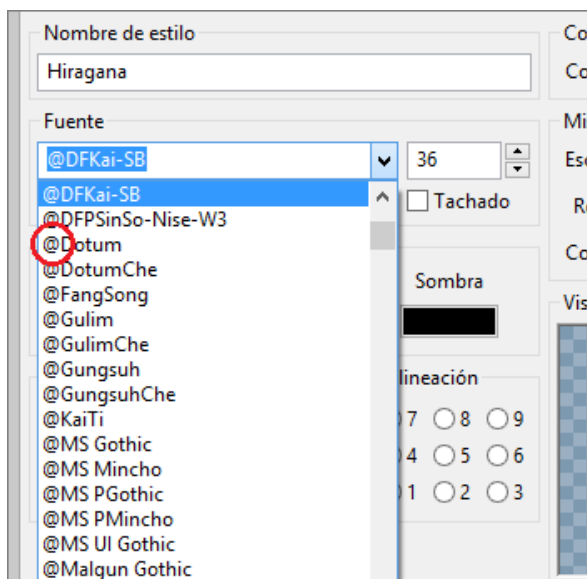
Y al aplicar los efectos seleccionados, la función reubicará automáticamente al texto para que las posiciones y ángulos de inclinación sean los mismos:



Otra de las utilidades de esta habilidad se usa para aplicar efectos a los **kanjis**, pero de forma vertical y utilizando las **fonts** que ya vienen prediseñadas para que el **kanji** se vea vertical con solo modificar la rotación del texto:



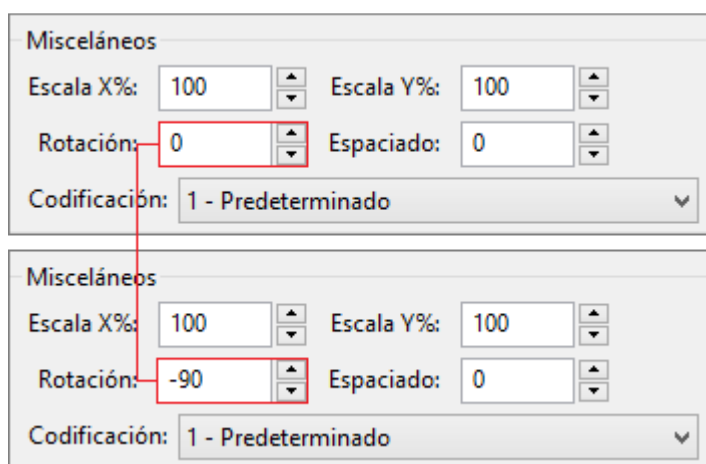
Las **fonts** que vienen prediseñadas para los **kanjis** en vertical, son la que en su nombre pone el símbolo de arroba (@) al inicio de su nombre:



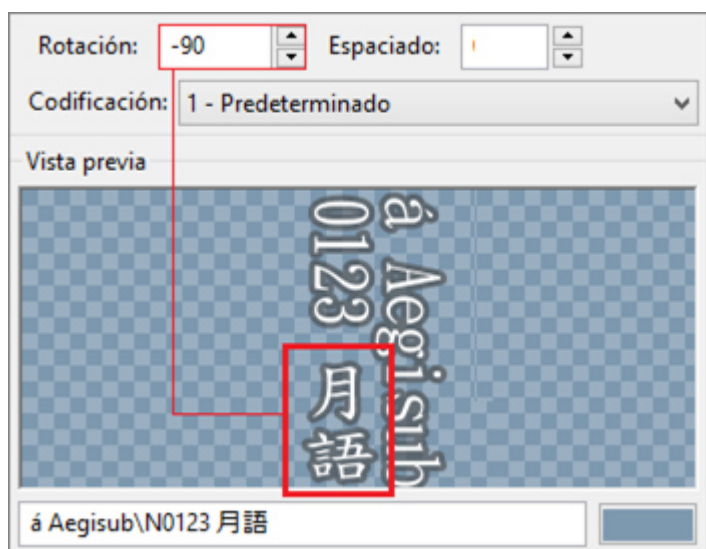
Seleccionamos una de esas **fonts** que más nos guste y se ajuste a nuestro proyecto, y vemos cómo el texto **kanji** está rotado en un ángulo de 90°:



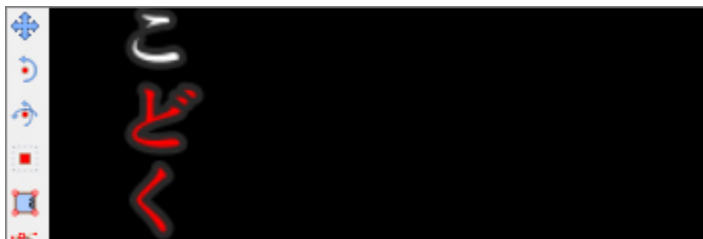
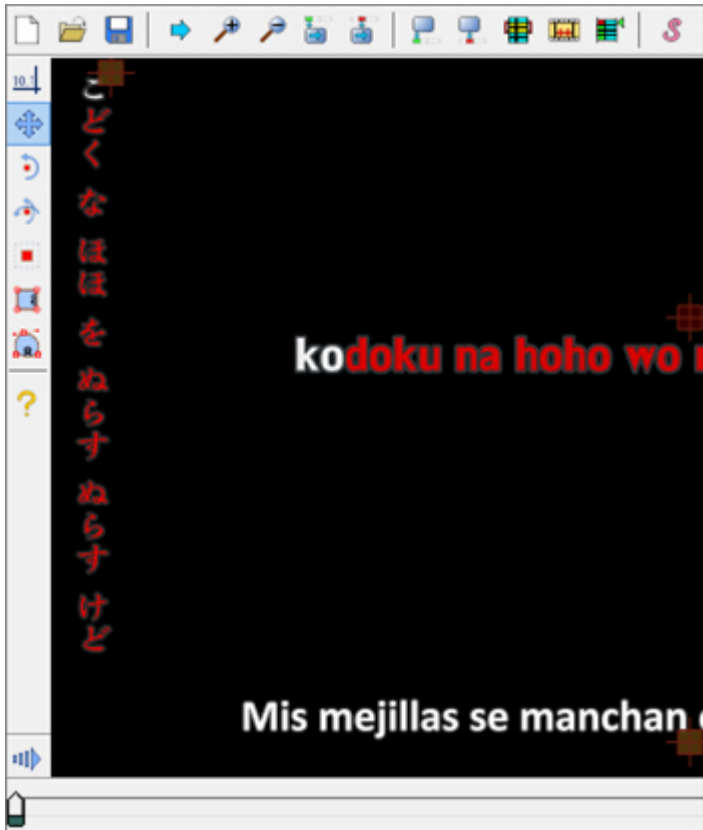
Entonces modificamos la rotación del texto así:



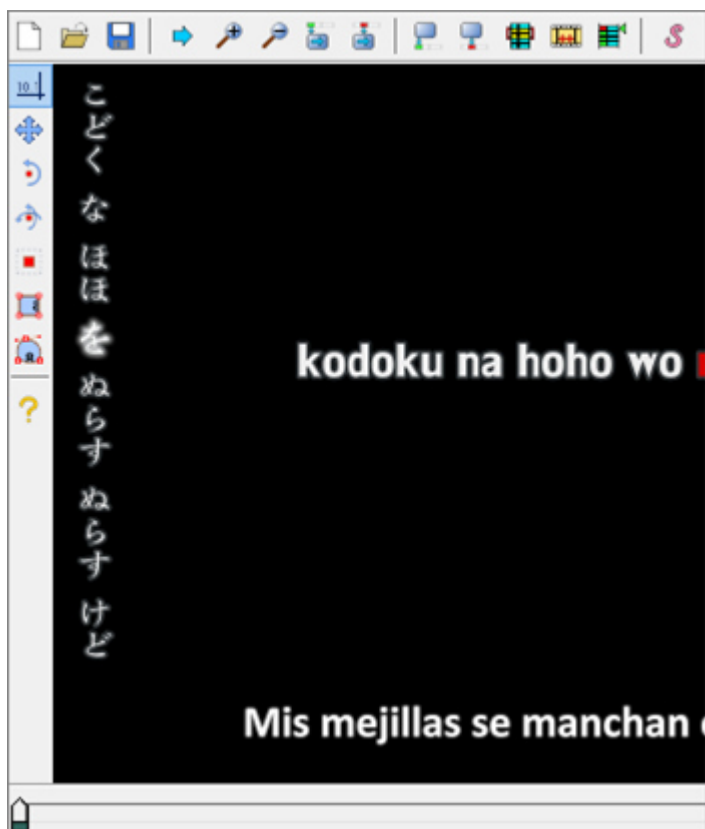
Ahora el **kanji** ya se ve bien en vertical aunque el resto del texto esté rotado -90°. Este hecho en particular no importa siempre y cuando en las líneas de texto solo haya **kanji**:



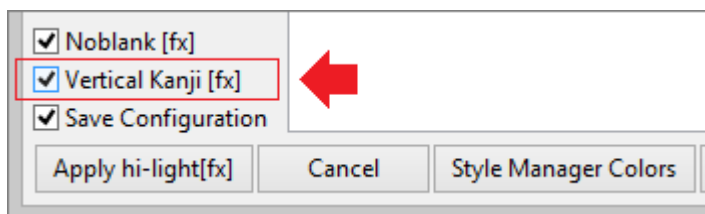
Hecho esto ya podemos ver en el vídeo cómo se ve el estilo del **kanji** en vertical sin aún haber aplicado ningún efecto:



Aplicando todos los efectos, la función hace los ajustes necesarios para que el texto **kanji** quede de forma vertical usando las **fonts** prediseñadas para ello:



La ventaja de esta habilidad es que hay un listado grande de **fonts** para **kanji** en vertical, y que de otro modo no podemos hacer con las **fonts** que son diseñadas para texto horizontal. Recordemos que para aquellas fonts de texto horizontal usamos la opción de “**Vertical Kanji [fx]**” del **KE** y obtenemos los mismos resultados:



effector.import(file_fx) :

Esta función nos permite importar un archivo **.lua** al **Kara Effector** para poder usar las funciones y variables que vienen en dicho archivo, al hacer uno o más efectos.

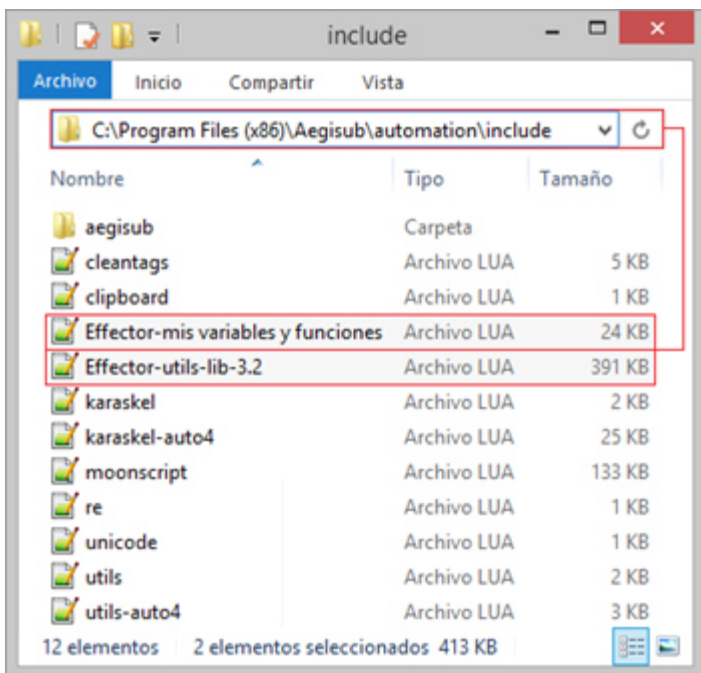
Esta función se usa en la celda de texto “**Variables**” y tiene una forma abreviada de llamarla:

```
import( file_lua )
```

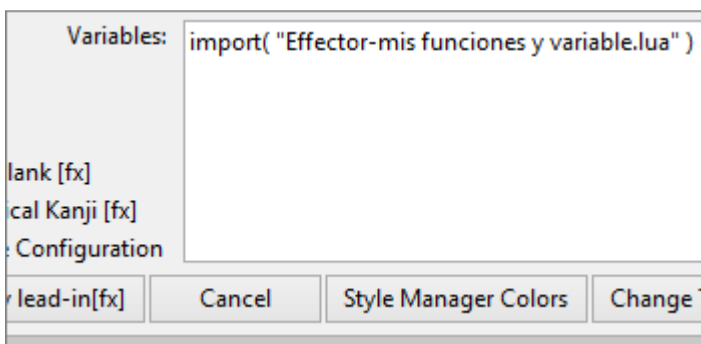
Hay dos formas de importar un archivo **.lua** al **KE**, una de ellas es guardar el archivo que queremos importar en la misma carpeta en donde esté guardado el archivo de la librería principal del **KE**: **Effector-utils-lib-3.2.lua**

- **Método 1:** El archivo **Effector-utils-lib-3.2.lua** del KE, la mayoría lo tenemos en la carpeta “**include**” del **Aegisub**, y es ahí en donde debemos guardar el archivo **.lua** que queremos importar al **Kara Effector**.

Ejemplo:

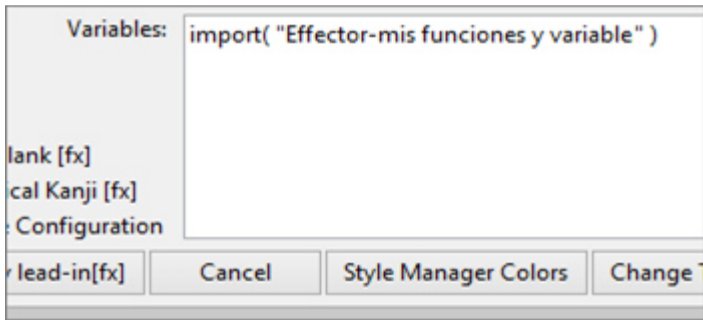


Y entonces, en la celda de textos “**Variables**” debemos poner así:



Obviamente no es necesario un nombre tan largo para el archivo **.lua** que vamos a importar, es solo un ejemplo.

Podemos omitir la extensión **.lua** del nombre del archivo y aun así poderlo importar satisfactoriamente:



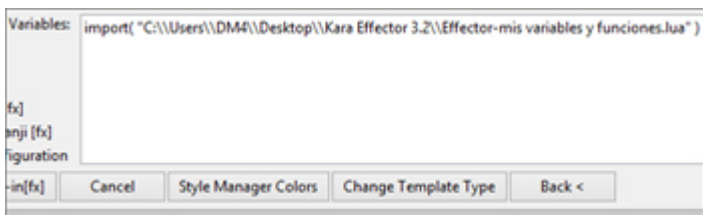
- **Método 2:** Con este método, no hay necesidad de guardar el archivo en la misma carpeta de la librería principal del **KE**, pero se debe poner la ruta completa de la posición del archivo dentro de la función:

Recordemos que las rutas deben ir con doble “\\”.

Ejemplo:

import(“C:\\Users\\DM4\\Desktop\\Kara Effector 3.2\\Effector-mis variables y funciones.lua”)

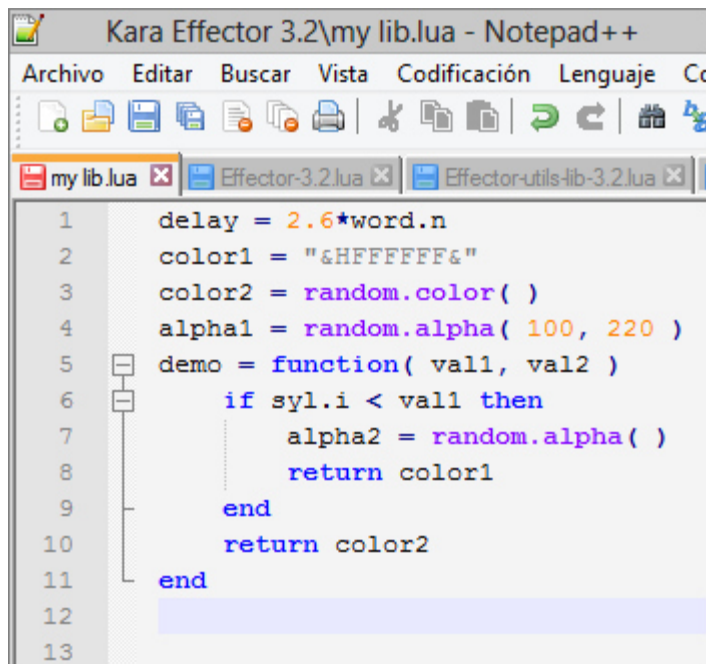
Y todo lo anterior debe ir en la celda de texto “**Variables**”:



Una vez comprendido la forma de importar un archivo **.lua** al **KE**, ya podemos ver un ejemplo de cómo puede ser dicho archivo y las ventajas que tiene el poder importarlo.

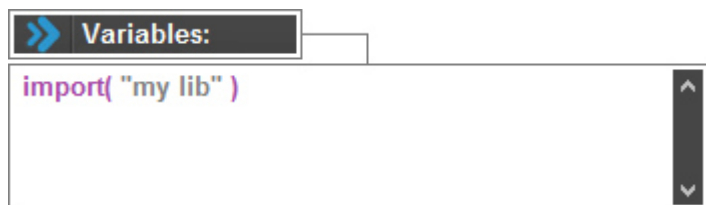
Ejemplo:

He creado un archivo llamado “**my lib.lua**” que contiene una serie de variables y una función sencilla, que he guardado en la misma carpeta en donde se encuentra la librería principal del KE:

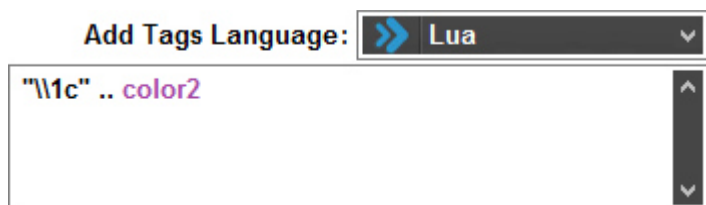


Obviamente al ser un archivo **.lua**, el lenguaje a usar para declarar las variables y las funciones debe ser el **LUA**, lo que hasta este punto ya no debería ser un obstáculo tan difícil de superar como lo era al principio del aprendizaje.

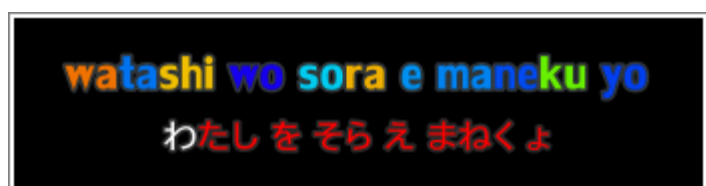
Guardado el archivo, lo importamos al **KE** en la celda de texto “**Variables**” como acabamos de aprender:



Una vez importado el archivo **.lua**, ya podemos disponer de cualquiera de sus variables o funciones declaradas en él, en cualquier celda de texto de la Ventana de Modificación del **KE**, incluso en la misma celda “**Variables**”. Ejemplo:



Y al aplicar el efecto, el **KE** reconocerá la variable “**color2**” que está en el archivo **.lua** creado, porque previamente lo ha importado:



El poder declarar variables y funciones en un archivo **.lua** independiente del **KE** nos da la ventaja de tener más espacio y comodidad para hacerlo. Otra de las ventajas es la siguiente:

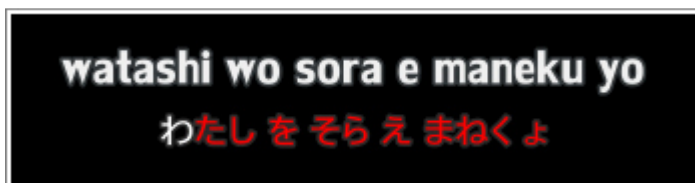
- Se puede modificar el archivo **.lua**, y la modificación se asume en tiempo real en el efecto en que importemos el archivo, sin la necesidad de tener que recargar el **KE**.

Ejemplo:

Modificamos la variable “**color2**”:

```
1  delay = 2.6*word.n
2  color1 = "&HFFFFFF&"
3  color2 = "&HFFFFFF&"
4  alpha1 = random.alpha( 100, 220 )
5  demo = function( val1, val2 )
6      if syl.i < val1 then
7          alpha2 = random.alpha( )
8          return color1
9      end
10     return color2
11 end
```

Y sin tener que recargar el **KE**, aplicamos el efecto como en el ejemplo anterior, y la modificación se verá reflejada de forma inmediata:



Saber y/o dominar el lenguaje de programación **LUA** no es un requisito para poder hacer efectos, ya que la razón del ser del **KE** es hacer sencillo el proceso de creación, edición y aplicación de los mismos. Para los que estén interesados en ampliar sus conocimientos en este lenguaje, recomiendo la siguiente web en español:

www.lua.org



O pueden descargar alguna de las ediciones en **PDF** del manual completo **LUA** en inglés:

- [Programando en LUA 2da Edición](#)
- [Programando en LUA 3ra Edición](#)



Al igual les dejaré ambos links en caso de que no funcionen los hipervínculos anteriores:

- 2da Edición: www.mediafire.com/?9aajra8ro6oukfa
- 3ra Edición: www.mediafire.com/?qf54ed3642rrn8o

Como les comentaba, este material es para aquellos que quieren profundizar en el lenguaje de programación **LUA** y aplicarlo en la creación de efectos, pero tengo pensado más adelante mostrarle unos cortos consejos de programación en **LUA** para hacer efectos en formato .ass

effector.addfx(library_fx, name_fx) :

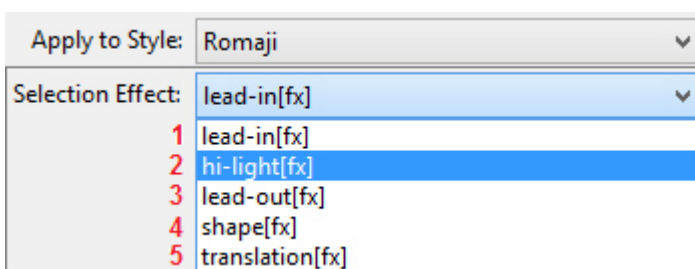
Esta función nos permite aplicar dos o más efectos al mismo tiempo, con la posibilidad de poder usar las variables y funciones declaradas en el primer efecto, en los demás adicionados. Su forma abreviada es:

addfx(library_fx, name_fx)

El parámetro **library_fx** es el nombre de alguna de las cinco librerías de efectos del **KE**:

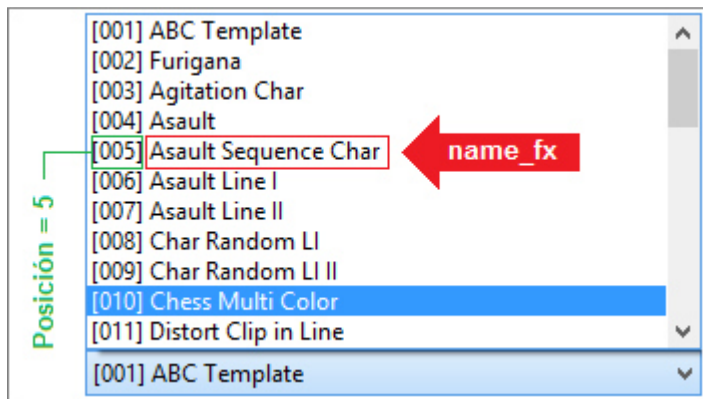
1. “lead-in”
2. “hi-light”
3. “lead-out”
4. “shape”
5. “translation”

Estas cinco librerías de efectos están en el mismo orden y hacen referencia a las opciones de la Ventana de Inicio del **Kara Effector**:



El parámetro **library_fx** puede ser el string del nombre de la librería de efectos o el número que indique su posición.

El parámetro **name_fx** es el string del nombre del efecto que queremos adicionar. Ejemplo:



Y de forma similar al parámetro **library_fx**, podemos poner el string del nombre del efecto a agregar o el número de la oposición que ocupa en su respectiva librería. Entonces podemos utilizar la función con cualquiera de las siguientes combinaciones:

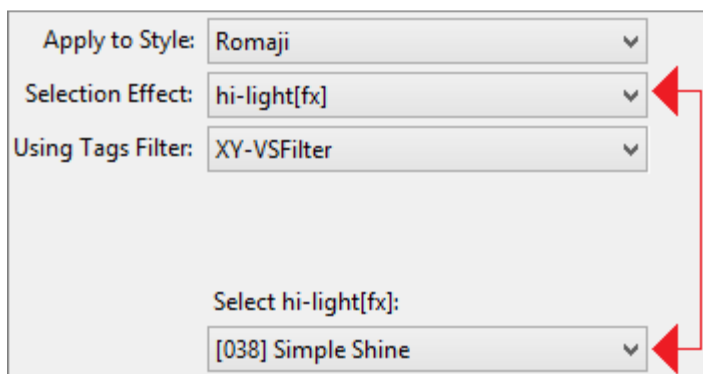
- **addfx(nombre_librería, nombre_efecto)**
- **addfx(nombre_librería, posición_efecto)**
- **addfx(posición_librería, nombre_efecto)**
- **addfx(posición_librería, posición_efecto)**

Esta función se usa en la celda de texto “**Variables**” y puede ser usada la cantidad de veces que así lo dispongamos.

Ejemplo:

Seleccionamos el primer efecto de la librería **lead-in[fx]** que es el efecto: **[001] ABC Template**

Y supongamos que queremos añadir a este lead-in el efecto “**Simple Shine**” de la librería **hi-light[fx]**:



Entonces tenemos la siguiente información del efecto que vamos a añadir:

- Librería: “**hi-light**”; Posición: 2
- Efecto: “**Simple Shine**”; Posición: 38

Y para añadirlo al lead-in[fx] “**ABC Template**“, podemos hacerlo con algunas de las siguientes cuatro opciones en la celda de texto “**Variables**“:

- `addfx(“hi-light”, “Simple Shine”)`
- `addfx(“hi-light”, 38)`
- `addfx(2, “Simple Shine”)`
- `addfx(2, 38)`

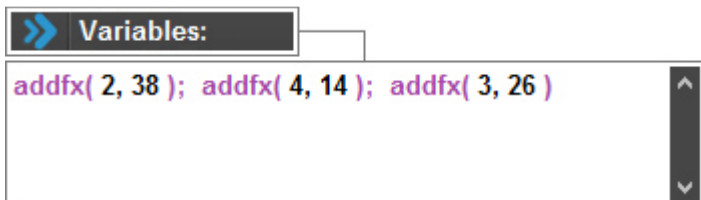
Ya sabemos que no importa cuál de las cuatro anteriores formas de hacerlo, la función añadirá el efecto al que ya hemos elegido, pero para el ejemplo usaré la primera, que es poniendo en nombre, tanto como el de la librería, como el del efecto:



Y al aplicar el lead-in, se aplicarán los dos efectos de forma simultánea, uno seguido del otro. Siempre se aplica primero el efecto de base, y luego todos los demás en el mismo orden en que los hayamos añadido:



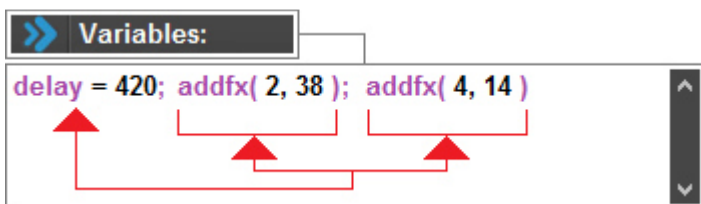
Si queremos añadir más de un efecto a la vez, los debemos poner de la siguiente forma. Ejemplo:



Lo que añadirá 3 efectos diferentes al que seleccionamos previamente en la Ventana de Inicio del **KE**.

El método para que una o más variables, al igual que las funciones declaradas en el primer efecto seleccionado en la Ventana de Inicio del **KE**, las podamos usar en los efectos a agregar, sin necesidad de tener que declararlas en ellos también, es el siguiente:

Ejemplo:



Al declarar la variable “**delay**” en este ejemplo, al inicio de la celda de texto “**Variables**“, nos aseguramos de que esta también pueda ser usada en todos los efectos añadidos que así lo dispongamos.

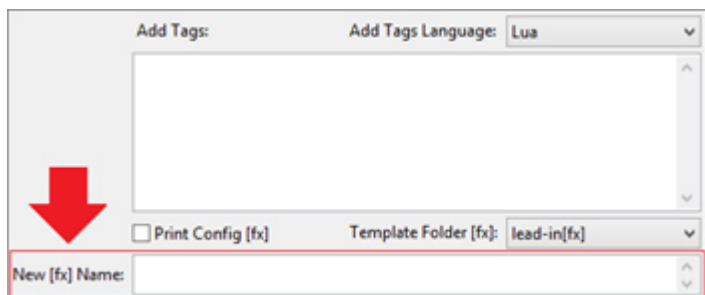
El aplicar más de un efecto al mismo tiempo es una ventaja cuando apenas estamos desarrollando el efecto total de un karaoke, cartel o logo; ya que no es evitará estar aplicando uno a uno a cada uno de los efectos individuales que lo componen.

effector.savefx(mode) :

Es una función interna del **KE** que exporta una serie de información, tanto del script .ass, como de los efectos a aplicar. Consta de 12 modos:

1. “**export config**”
2. “**export times ms**”
3. “**export times**”
4. “**export lines**”
5. “**export text**”
6. “**export hira**”
7. “**export kata**”
8. “**export roma**”
9. “**export text_stripped**”
10. “**export hira_stripped**”
11. “**export kata_stripped**”
12. “**export roma_stripped**”

Toda la información que exporta esta función, lo hace en un archivo nuevo en formato .txt, y los modos de la función se deben ingresar en la celda de texto “**New [fx] Name**”:



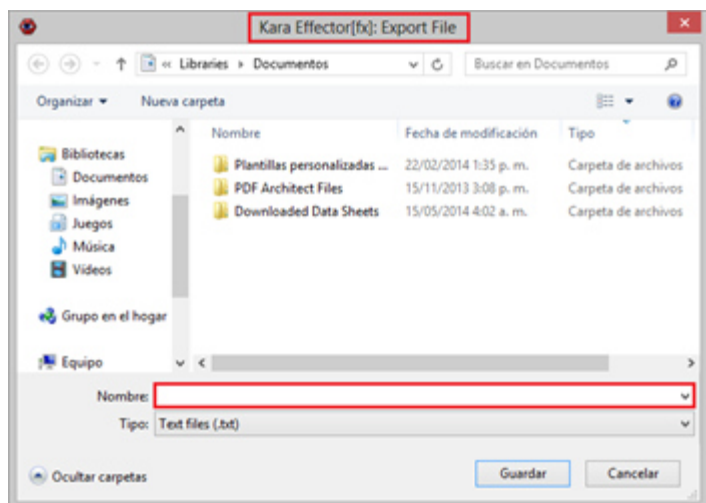
Lo que debemos hacer es escribir el modo en esta celda de texto, sin las comillas, y aplicar el efecto como siempre lo hacemos de forma normal.



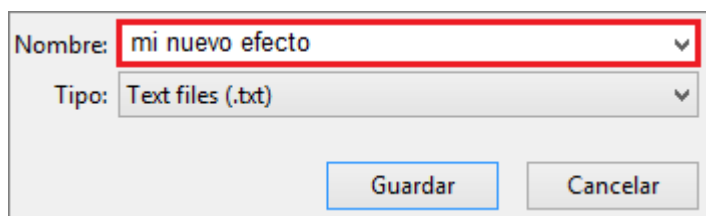
Exporta en un archivo .txt a todas las configuraciones del efecto, que posteriormente puede ser compartido con otros usuarios del **KE**.

Ejemplo:

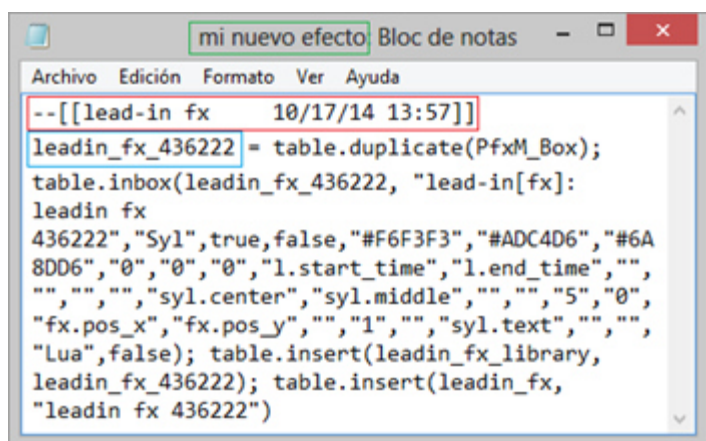
Al usar cualquiera de los doce modos de la función, nos debe salir un cuadro de dialogo como el de la siguiente imagen, en donde decidiremos el lugar de destino del archivo y el nombre del mismo:



Este es el método rápido de exportar las configuraciones del efecto en un archivo .txt, y una vez hallamos decidido en lugar de destino, le ponemos un nombre a dicho archivo:



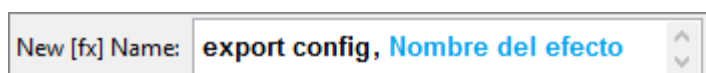
Al abrir el archivo exportado, veremos algo como esto:



De la anterior imagen del archivo exportado se resaltan tres rectángulos de diferentes colores:

- En verde, el nombre del archivo .txt
- En rojo vemos 2 datos, primero la librería de efectos en la que se guardará el efecto una vez que hallamos pegado estas configuraciones en el **KE**, en el archivo **Effector-newfx-3.2.lua**. y segundo, la fecha y hora de creación del archivo. La fecha en formato Mes/Día/Año y la hora en HH:MM
- En azul, el nombre que por default el KE le asigna a las configuraciones de un efecto al cual no le hallamos puesto uno.

El método para asignarle un nombre al efecto exportado en un archivo .txt es el siguiente:

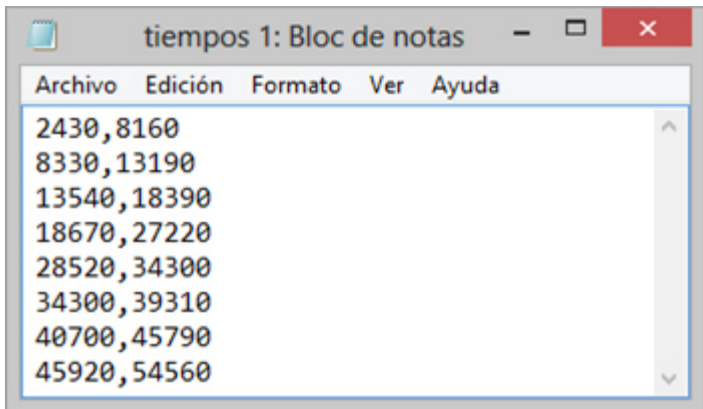


O sea, escribimos el modo “**export config**” y el nombre que le pondremos al efecto, separados por una coma (,). Ambos sin comillas.

Modo 2: <

New [fx] Name: export times ms

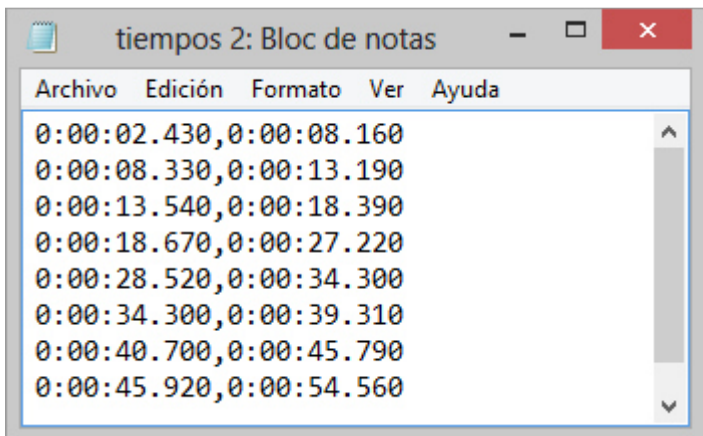
Exporta un archivo .txt con los tiempos de inicio y final en ms de las líneas seleccionadas para aplicar un efecto:



Modo 3: <

New [fx] Name: export times

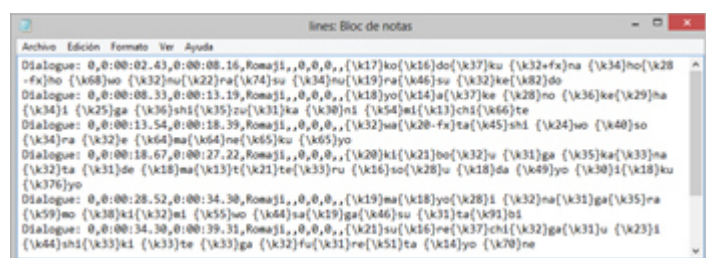
Exporta un archivo .txt con los tiempos de inicio y final en formato HH:MM:SS.ms de las líneas seleccionadas para aplicar un efecto:



Modo 4: <

New [fx] Name: export lines

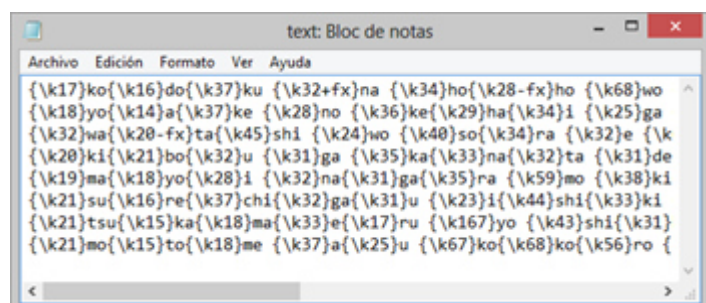
Exporta un archivo .txt con la información completa de las líneas seleccionadas para aplicar un efecto, con el mismo formato de un script .ass:



Modo 5:

New [fx] Name: export text

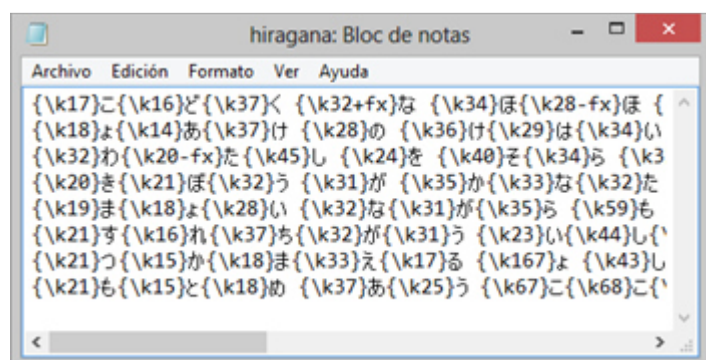
Exporta un archivo .txt con el contenido completo de las líneas seleccionadas para aplicar un efecto, incluidos los tags (etiquetas):



Modo 6:

New [fx] Name: export hira

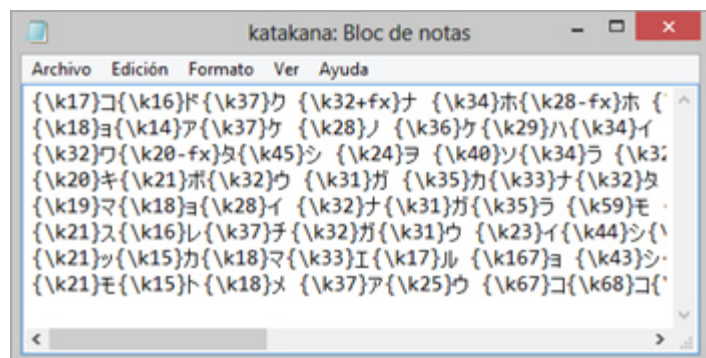
Exporta un archivo .txt con el **romaji** o el **katakana** de las líneas karaoke seleccionas, convertidos en **hiraganas**:



Modo 7:

New [fx] Name: export kata

Exporta un archivo .txt con el **romaji** o el **hiragana** de las líneas karaoke seleccionas, convertidos en **katakanas**:



Modo 8:

New [fx] Name: **export roma**

Exporta un archivo .txt con el **hiragana** o el **katakana** de las líneas karaoke seleccionas, convertidos en **romajis**:



Modo 9:

New [fx] Name: **export text_stripped**

Este modo es similar al modo 5, pero exporta el contenido de las líneas de texto seleccionadas sin los tags (etiquetas).

Modo 10:

New [fx] Name: **export hira_stripped**

Este modo es similar al modo 6, exporta el contenido de las líneas de texto seleccionadas convertidas a **hiraganas**, pero sin los tags (etiquetas).

Modo 11:

New [fx] Name: **export kata_stripped**

Este modo es similar al modo 6, exporta el contenido de las líneas de texto seleccionadas convertidas a **katakanas**, pero sin los tags (etiquetas).

Modo 12: <

New [fx] Name: export roma_stripped

Este modo es similar al modo 6, exporta el contenido de las líneas de texto seleccionadas convertidas a **romajis**, pero sin los tags (etiquetas).

No está demás aclarar que los modos que exportan a los **romajis**, **katanas** y **hiraganas** están pensados para las líneas de karaoke que tienen las sílabas separadas, es decir que no aplican a las líneas de traducción normales.

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**:

▼ Edit Actor ▼ Efecto ▼ 36

8.16 0:00:05.73 0 0 0

AB AB AB ✓ ☒ Tiempo ☐ Cuadro ☐ Mostrar original

\k32+fx)na (\k34)ho(\k28-fx)ho (\k68}wo (\k32}nu(\k22}ra(\k

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

Edit Actor ▼ 10, 24, 11.4, 86 ▼ 36

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*
0:00:18.39	Romaji	10, 24, 11.4, 86	*wa*ta*shi *wo *so*ra *
0:00:27.22	Romaji		*ki*bo*u *ga *ka*na*ta
0:00:34.30	Romaji		*ma*yo*na*ga*ra *
0:00:39.31	Romaji		*su*re*chi*ga*u *i*shi*
0:00:45.79	Romaji		*tsu*ka*ma*ru *yo *
0:00:54.56	Romaji		*mo*to*me *a*u *ko*ko
0:00:08.16	Hiragana		*こ*ど*く *な *ほ*ほ *
0:00:13.19	Hiragana		*よ*あ*の *け*は*

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* *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*u *ga *ka*na*ta *de *ma*t*
Romaji		*ma*yo* *na*ga*ra *mo *ki*mi *w*
Romaji		*su*re*chi*ga*u *i*shi*ki *te *ga *
Romaji		*tsu*ka*ma*e*ru *yo *shi*ka*ka*ri*
Romaji		*mo*to*me *a*u *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*ki *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*u *ga *ka*na*ta *de *ma*t
Romaji	<div> <div></div> <div>fx.offset_x = effect_val[1]</div> <div>fx.offset_y = effect_val[2]</div> <div>fx.offset_z = effect_val[3]</div> </div>	
Romaji		
Romaji		
Romaji		
Hiragana		

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*
0:00:13.19	Romaji	1000	*yo*a*ke *no *ke*ha*ki *ga *shi*
0:00:18.39	Romaji		*wa*ta*shi *wo *so*ra *e *ma*
0:00:27.22	Romaji		*ki*bo*u *ga *ka*na*ta *de *ma*
0:00:34.30	Romaji		*ma*yo*ki *na*ga*ra *mo *ki*mi
0:00:39.31	Romaji		*su*re*chi*ga*u *ki*shi*ki *te *
0:00:45.79	Romaji		*tsu*ka*ma*te*ru *yo *shi*k*ka
0:00:54.56	Romaji		*mo*to*me *a*u *ko*ko*ro *so*
0:00:08.16	Hiragana		*こ*ど*く *な *ほ*ほ *を *ぬ*ら*

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.Remove**
- **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.preprocesses_styles**: esta función hace que veamos en lista a todos los estilos disponibles para aplicar efectos de los que están en el script.
- **effector.preprocesses_macro**: esta función prepara la información que contienen las dos ventanas del **KE** y la organiza en las respectivas celdas.
- **effector.preprocesses_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**.