

Primeros pasos con Smile

Traducción: Pescados Software

Satimage-software
www.satimage-software.com

COPYRIGHT © 2004 SATIMAGE (FRANCE)

26 de noviembre de 2004

Índice

| | |
|---|----------|
| 1. Introducción | 3 |
| 2. Escribir scripts en Smile | 3 |
| 2.1. El interpretador de Applescript | 3 |
| 2.2. El contexto persistente de Smile | 4 |
| 2.3. Monitorizando la ejecución de scripts y depurando código . . . | 5 |
| 3. Confort y productividad | 6 |
| 3.1. La Hoja de Trabajo | 6 |
| 3.2. Abrir diccionario. | 6 |
| 3.3. Encontrando una definición | 7 |
| 3.4. Insertar rápidamente un “file path” | 7 |
| 3.5. Búsqueda de texto | 8 |
| 3.6. Comparar documentos | 9 |
| 3.7. Escribiendo bibliotecas | 9 |
| 4. Applets y documentos de script | 9 |
| 4.1. Ventanas de script | 9 |
| 4.2. Guardar un script | 10 |
| 4.3. Guardar como applet o droplet | 11 |
| 4.4. Guardar un script sin el código | 11 |

1. Introducción

Smile es un entorno integrado de desarrollo de AppleScript. Smile incluye:

- un editor de scripts
- un constructor de interfaces
- una biblioteca de generación de gráficos
- un entorno interactivo de ploteo de datos numéricos (SmileLab)

El presente documento presenta las características de Smile como editor de scripts¹.

2. Escribir scripts en Smile

2.1. El interpretador de Applescript

La característica única de Smile #1, que mejorará su experiencia escribiendo código es su interpretador de AppleScript.

Cada ventana de texto² — aparte de funcionar como una ventana normal de edición de textos — es capaz de interpretar AppleScript. Pulsando *Enter* (no *Carriage Return* como en el Terminal) en una ventana de texto, compila y ejecuta en el acto el texto seleccionado.

Por defecto, Smile añade el resultado (si lo hay) del script a una ventana de texto especial, llamada `Console`³.

La otra opción es añadir el resultado a la misma ventana de texto, que es útil si vd. quiere reusar dicho resultado (como ocurre con una calculadora). Puede escoger una de las dos opciones en **Scripting > Output to Console**.

Las ventanas de texto ofrecen una experiencia distinta a la hora de escribir código, la interactividad. Esto le ayudará a probar y debuggear sus scripts. Una vez que el script haya sido suficientemente probado, lo puede transformar, por ejemplo, en un applet (esto se documenta en otro capítulo).

¹Este documento no presenta el lenguaje de AppleScript en sí mismo. Para documentarse sobre AppleScript, lea la documentación oficial de Apple.

²Las ventanas de Smile son blancas.

³Smile crea automáticamente una ventana `Console` cuando es necesario.

2.2. El contexto persistente de Smile

La característica única de Smile#2, que hace la programación intuitiva, es que Smile mantiene permanentemente un contexto global, y las instrucciones que se editan en una ventana de texto se ejecutan en ese ambiente.

Cualquier variable que vd. define en una ventana de texto es persistente — permanecerá definida hasta reiniciar Smile. Esto hace el interpretador de AppleScript mucho más valioso: puede vd. realmente editar y ejecutar un script línea por línea.

Por ejemplo, vd. modifica el valor de cualquier variable ejecutando texto desde cualquier ventana de texto, por ejemplo:

Ejemplo 1

```
set myVariable to 2
```

Y cuando le venga a vd. en gana, puede ver los contenidos de la variable en cualquier momento:

Ejemplo 2

```
myVariable
-- 2
```

Los handler que se compilan en ventanas de texto también son persistentes, y accesibles desde todas las ventanas de texto.

Ejemplo 3

```
on bip()
    beep 2
end bip
```

Cuando este bloque se selecciona y compila -como se ha descrito anteriormente-, el handler está disponible globalmente, y puede vd. invocarlo desde cualquier ventana de texto:

Ejemplo 4

```
bip() -- will beep twice
```

Éste es el mecanismo básico que le permite extender su entorno de programación (vea la sección 3.7).

Nota: para referirse a una variable definida previamente en un handler, debe declararse explícitamente como una variable global.

Ejemplo 5

```
on bip()
    beep myVariable
end bip
-- compilará bien, pero dará error al ejecutarse:
-- "The variable myVariable is not defined"
```

Ejemplo 6

```
on bip()
    global myVariable
    beep myVariable
end bip
-- correcto
```

2.3. Monitorizando la ejecución de scripts y depurando código

Depurar un script en Smile es distinto de otros programas —pero es más productivo. Smile aprovecha la ventaja de la naturaleza "interpretada" de AppleScript que le permite algo similar a "incrustar un script dentro del contexto propio de Smile. Esto significa dos cosas: 1. las variables y handlers de su script son permanentemente accesibles y modificables - 2. el diccionario de Smile y sus rutinas están siempre disponibles para ayudar en el proceso de depuración.

- para monitorizar un script, se inserta uno de los comandos que Smile propone. Las funciones más útiles son `quietmsg(s)` y `postit s`, que muestran `s` respectivamente al final de la ventana Console ⁴ y en una pequeña ventana flotante llamada `Message`.

Otra herramienta útil es `smilepause`, que funciona como un "break-point". Si inserta, por ejemplo, `smilepause 30`, el script parará por 30 segundos sin colgar la aplicación, de manera que puede vd. utilizar una ventana de texto para leer una cantidad, probar una expresión, cambiar una variable... Para reanudar la ejecución, presione la tecla *right arrow*. Para cancelar la ejecución, pulse la tecla *esc*.

⁴`quietmsg` es la versión "silenciosa" de `msg` que, además, trae la ventana Console al primer plano.

- para depurar un script o handler, cópielo en una ventana de texto y ejecútelo línea a línea, o bloque a bloque. En cualquier momento puede vd. leer cantidades, probar expresiones, etc. Las variables son persistentes, nunca tiene vd. que re-ejecutar el script desde el principio, como ocurre en el Editor de Scripts.

Para mostrar un valor que no puede mostrarse como texto de manera natural, Smile utiliza el comando `display`, que transforma cualquier elemento de AppleScript en texto.

Ejemplo 7

```
quietmsg(display thefiles)
repeat with f in thefiles
  try
    -- process f
    postit (f & ".°k")
  on error errstr
    quietmsg(f & ":"& errstr)
  end try
end repeat
```

3. Confort y productividad

3.1. La Hoja de Trabajo

Smile tiene una ventana de texto especial, llamada, **Hoja de Trabajo**. Por defecto, Smile abre la Hoja de Trabajo⁵ al iniciar Smile. Y al salir, Smile guarda los cambios automáticamente, como un archivo de texto en la carpeta `~/Library/Application Support/Smile/`.

3.2. Abrir diccionario...

Smile abre diccionarios en ventanas de texto con un menú popup **Index** en su barra de herramientas. Vd. puede manejar ventanas de diccionarios como ventanas de texto normales y corrientes.

El menú **Index** muestra la lista de "Suites" (negrita, subrayado), de comandos

⁵Por defecto, la Hoja de trabajo devuelve el resultado de los scripts al final de sí misma — no en la Consola.

(texto plano) y `classes`” (cursiva). Presionando la tecla *option* y escogiendo el menú, muestra los items en orden alfabético.

El submenú **Archivo > Abrir diccionario...** muestra una lista de items típicos para abrir sus diccionarios de forma rápida.

- **AppleScript**: muestra el diccionario básico de AppleScript
- **Adiciones de Script**, un submenú: muestra una lista de las adiciones de script instaladas
- la lista de aplicaciones scriptables que se están ejecutando
- **System Events**: una aplicación background que contiene muchos comandos útiles referentes a archivos, carpetas y volúmenes

Para abrir el diccionario de una aplicación que no esta abierta, arrastre su icono al icono de Smile en el Finder (o en el Dock), o utilice los otros items del submenú en **Abrir diccionario...**:

- **Explorer** presenta una lista de todas las aplicaciones scriptables (como el Editor de Scripts)
- **Otros...** abre los servicios de navegación habituales (abre más rápido)

3.3. Encontrando una definición

Para encontrar la definición de un término, seleccione el término y escoja **Encontrar definición** en el menú **Scripting**. El término puede ser un comando (p. ej. `write`) o un `class name`” (p. ej. `window`). Smile busca primero en su propio diccionario, después en el diccionario básico de AppleScript y en los de las Adiciones de Script instaladas.

Si Smile encuentra la definición para el término, abre el(los) diccionario(s) donde fue encontrado el término, y resalta las entradas relevantes.

3.4. Insertar rápidamente un “file path”

Puede vd. insertar fácilmente una referencia a un archivo (como un **alias**): arrastre su icono desde el Finder hasta una ventana de texto y aparecerá en el punto de inserción. Presionando la tecla *shift* al mismo tiempo inserta, en cambio, el “POSIX path” del archivo.

3.5. Búsqueda de texto

Para buscar y reemplazar texto, utilice el diálogo **Encontrar**. Aparte de búsquedas típicas en la presente ventana de texto o script, el diálogo **Encontrar** proporciona utilidades avanzadas para búsqueda de texto.

Buscando en carpetas

Utilizando el diálogo **Buscar**, vd. puede seleccionar un directorio y buscar en los documentos que contiene: active la casilla **Búsqueda carpeta**, y utilice el menú para seleccionar el directorio.

Smile buscará⁶ en todos los archivos de texto y scripts compilados (incluyendo applets) dentro de la carpeta seleccionada y dentro de todas sus subcarpetas. Una ventana llamada **Resultados de la búsqueda** mostrará el estado de la búsqueda en su barra de herramientas, y los resultados en su parte principal. Cada ítem encontrado produce dos líneas en la ventana: una que empieza con **mostrar** y otra que muestra la línea donde fue encontrado el término de búsqueda.

Ejemplo 8

```
show "Macintosh HD:Users:mike:Documents:Urgent"selection
1330, 1339
- purchase the license for smilelab
```

Para mostrar el resultado, ponga el cursor en la línea que empieza con **mostrar** y pulse *Enter*.

Expresiones regulares

El diálogo **Buscar** ofrece la opción “Regexp” (de “regular expressions”). Si activa vd. “Regexp”, el texto que introduce como parámetro de búsqueda, define un patrón de expresión regular: la mayoría de caracteres conservan su significado literal, pero otros de ellos se convierten en caracteres comodín, que permiten definir búsquedas avanzadas de texto. La opción **regexp** del diálogo **Buscar** funciona igual que los comandos **find text** y **change** de Satimage osax⁷. La utilización de Expresiones regulares queda cubierta en la documentación de Satimage osax.

⁶Para interrumpir la búsqueda, presione la tecla *esc*. Cerrar la ventana del resultado también finalizará la búsqueda.

⁷Sin embargo, algunas opciones sólo se pueden utilizar desde un script, y no desde el diálogo **Buscar**.

3.6. Comparar documentos

El menú Edición > Comparar (atajo de teclado *cmd-K*) compara el texto de las dos ventanas frontales, empezando en el sitio donde se encuentre el cursor de cada ventana, y selecciona el primer bloque de cada ventana donde hay diferencias –o el sitio donde falta texto. Pulsando otra vez *cmd-K* pasará a la siguiente diferencia.

3.7. Escribiendo bibliotecas

Smile utiliza un mecanismo para cargar bibliotecas al lanzarse: los documentos de texto que se guardan en la carpeta `Class scripts/Context additions` son compilados y cargados cuando se abre Smile. Puede vd. guardar sus propias bibliotecas en su carpeta de usuario, en:

```
~/Library/Application Support/Smile/Class scripts/Context  
additions/
```

4. Applets y documentos de script

Algunas veces, un script como simple texto no es suficiente. Es el caso, por ejemplo, cuando crea vd. una aplicación (“applet” o “droplet”). Hay también circunstancias en que tiene vd. que guardar un script como “script compilado”: el Finder necesita scripts compilados para ejecutar Acciones de carpeta, al igual que el Menú de Scripts de sistema y el propio  menú de scripts de Smile⁸.

Esa es la razón de que existan ventanas de script en Smile.

4.1. Ventanas de script

Smile es capaz de abrir scripts compilados, applets y droplets en ventanas de script, que tienen color de fondo. Las ventanas de script funcionan de modo muy similar a las ventanas del Editor de Scripts.

El menú Archivo > Nuevo Script crea una nueva ventana de script en blanco.

⁸El  menú muestra el contenido de la carpeta `User scripts` de Smile. Vd. puede crear una carpeta `User scripts` en `~/Library/Application Support/Smile/`.

El menú **Handlers** visible en la cabecera de la ventana muestra la lista de handlers⁹ en el script (*option-click* para mostrar las entradas en orden alfabético).

Para mostrar/ocultar el menú **Handlers** en ventanas de scripts, utilice el menú **Scripting > Listar handlers**. Las ventanas de texto también tienen esta característica.

Cuando vd. trabaja con una ventana de script, utiliza los mismos menús que si estuviera editando texto normal, y además puede utilizar los comandos específicos para scripts que se encuentran en el menú **Scripting**.

- **Ejecutar script** tratará de compilar y ejecutar el texto de la ventana activa. El resultado de la ejecución se mostrará en la **Consola**. Para interrumpir un script, presione *cmd-*. (punto) o la tecla *esc*.
- **Comprobar sintaxis** tratará de compilar el script de la ventana activa, y mostrará cualquier error que encuentre al hacerlo. En una ventana de script, el uso de la tecla *Enter* equivale a seleccionar el menú **Comprobar sintaxis**.

4.2. Guardar un script

Para guardar un script, utilice los ítems **Guardar** o **Guardar como...** del menú **Archivo**. Si está guardando un script nuevo, o si escoge el menú **Guardar como...**, el menú **Formato** del diálogo **Guardar** le permitirá escoger una de varias opciones. Si prefiere guardar el script como un documento normal **Documento** (por defecto), entonces puede guardarlo en cualquiera de los dos formatos disponibles:

- guardar el script como archivo compatible con sistemas pre-OS X, sin extensión,
- guardar el script como data-fork, sólo compatible con OS X, proporcionando explícitamente la extensión **.sct** al nombre del documento.

⁹El menú **Handlers** puede mostrar comentarios adicionales. Cualquier línea que comience con **-- mark** genera automáticamente una nueva entrada en el menú **Handlers**: el menú muestra la fracción de línea que sigue a **mark**.

4.3. Guardar como applet o droplet

Los applets y droplets son aplicaciones autoejecutables. Para guardar un script como applet, escoja el menú **Formato > Aplicación** en el diálogo **Guardar**¹⁰.

Al hacer doble-click sobre un applet, se ejecuta su handler **run** (o sin nombre), y luego se auto-quita. Para que el applet siga abierto después de ejecutar su handler **run** (que puede estar vacío), escoja el menú **Formato > Aplicación - Dejar abierta** al guardarlo. Implícitamente, los applets son en sí mismos scriptables, y cualquier script puede hacer llamadas a sus handlers en un bloque `tell ... end tell`.

4.4. Guardar un script sin el código

Los tres últimos ítems del menú **Formato** del diálogo **Guardar** son para guardar el script como **sólo-ejecución**, esto es, sin guardar el código del script. Tras guardar con esta opción, no se puede ver ni editar el script nunca más: sea cauto y guarde siempre una copia editable.

¹⁰Como en el Editor de Scripts, si el script incluye un handler **open**, Smile lo guarda como un droplet, en vez de applet. Un droplet ejecuta su handler **open** cuando se arrastran archivos/carpetas a su icono.