

GUÍA RÁPIDA - PLANTILLAS

Manuales generales.

TÉRMINO	DESCRIPCIÓN
Introducción – Sintaxis	Creación y edición de plantillas
Introducción – Caché	Modo de preparación de las plantillas

Sintaxis de plantilla.

La sintaxis de las plantillas *Sigma* define un conjunto de elementos especiales:

- **Variables o sustituciones**, conocidos como *placeholders*, consisten en una serie de etiquetas que son sustituidas por el texto deseado en el momento de la *renderización*. Conceptualmente corresponden a huecos pendientes de rellenar. La sustitución se realiza mediante las funciones *setVariable()* y *setGlobalVariable()*. En caso de no realizar ninguna sustitución, el hueco o *placeholder* quedaría vacío.

Sintaxis: {[0-9A-Za-z_-]+}

- **Definición de bloques**, la utilización de bloques es uno de los principales mecanismos de las plantillas. Toda la información de la plantilla se estructura en bloques, siendo englobada en su totalidad por el bloque virtual “__global__”. Cada bloque puede *renderizarse* tantas veces como sea necesario, pudiendo no *renderizarse* ninguna vez y siendo, por tanto, eliminada. Los bloques pueden anidarse, con la única obligación de *renderizar* los bloques interiores antes que los exteriores. La función de *renderización* de los bloques es *parse()*.

```
<!-- BEGIN [0-9A-Za-z_-]+ -->
... block content ...
<!-- END [0-9A-Za-z_-]+ -->
```

- **Inclusión de ficheros externos**, es posible incluir dentro de una plantilla ficheros externos, que a su vez deberán ser plantillas. El proceso de inclusión de ficheros externos se realiza antes que las sustituciones y *renderizado*, por lo que no es posible sustituir el nombre de un fichero a incluir en la plantilla. Sin embargo, la función *addBlockfile()* puede permitirnos la inclusión de ficheros que no conocemos de antemano.

```
... some content ...
<!-- INCLUDE filename.html -->
... some more content ...
```

- **Funciones de plantillas**, *Sigma* permite la utilización de funciones dentro de las mismas plantillas. Dado que las plantillas no incluyen código ejecutable propio, estas funciones se emplean como *placeholders*, que deben enlazarse con funciones *PHP* existentes durante la *renderización* de la plantilla. Para asociar una función de plantilla con una función *PHP* se emplea el método *setCallbackFunction()* teniendo en cuenta que el número de parámetros de las funciones asociadas debe coincidir, y que el texto devuelto por la función *PHP* rellenará el hueco dejado por la función de la plantilla. Los argumentos introducidos en las funciones de plantilla pueden ser a su vez *placeholders*.

```
... some content ...
func_h1("embedded in h1")
... some more content ...
```

Funcionalidad: Variables y PlaceHolders.

MÉTODO	DESCRIPCIÓN	DEF.
clearVariables	Cancela la sustitución de todas las variables no globales del bloque.	Def
getPlaceholderList	Devuelve la lista de los placeholders contenidos en el bloque indicado.	Def
placeholderExists	Comprueba si existe un placeholder con el nombre indicado dentro de la plantilla o de un bloque en concreto.	Def
setGlobalVariable	Establece la sustitución de una variable global.	Def
setVariable	Establece la sustitución de un placeholder por un valor concreto.	Def

Funcionalidad: Acceso y renderización de los bloques.

MÉTODO	DESCRIPCIÓN	DEF.
blockExists	Comprueba si existe un bloque con el nombre indicado.	Def
getBlockList	Devuelve la lista de los bloques contenidos en la plantilla o en un bloque concreto.	Def
getCurrentBlock	Devuelve el nombre del bloque actual.	Def
parse	Completa la renderización del bloque indicado.	Def
parseCurrentBlock	Completa la renderización del bloque actual. Es importante tener en cuenta que Sigma no mantiene una pila con los bloques que están en uso por lo que invocar dos veces seguidas a <code>parseCurrentBlock</code> no tendrá sentido, ya que Sigma no considerará que halla cambiado el bloque a renderizar.	Def
setCurrentBlock	Establece el bloque activo, o bloque sobre el que se realiza la sustitución de variables.	Def
touchBlock	Impide que el bloque desaparezca pese a no haber sustituido ninguna variable contenida en el mismo. Por defecto, aquellos bloques en los que no se ha realizado ninguna sustitución de variables son eliminados al renderizarlos	Def
hideBlock	Opuesto a <code>touchBlock</code> . Este método oculta o elimina el bloque indicado.	Def

Funcionalidad: Sustituciones especiales.

MÉTODO	DESCRIPCIÓN	DEF.
addBlock	Sustituye un placeholder por un nuevo bloque.	Def
addBlockfile	Permite sustituir un placeholder por un nuevo bloque.	Def
replaceBlock	Reemplaza el contenido de un bloque por el texto suministrado en la invocación a la función	Def
replaceBlockfile	Reemplaza el interior de un bloque por el contenido del fichero indicado.	Def

Funcionalidad: funciones principales de renderización.

MÉTODO	DESCRIPCIÓN	DEF.
constructor HTML_Template_Sigma	Crea el objeto principal para la renderización de plantillas, además de establecer, opcionalmente, el directorio raíz de las plantillas, así como el directorio en que se realizarán las tareas de caché.	Def
loadTemplateFile	Cargar el contenido del fichero de la plantilla.	Def
setTemplate	Cargar la plantilla directamente a través de una cadena de texto. El resultado es el mismo que loadTemplateFile con la diferencia del origen de la plantilla.	Def
get	Devuelve una cadena con el resultado de la renderización. Puede emplearse para devolver en su lugar la renderización de un bloque concreto.	Def
show	Envía al usuario final el resultado de la renderización efectuada. Es equivalente a enviar al usuario el resultado de la función get. Al igual que get, puede usarse con un bloque en concreto.	Def

Funcionalidad: opciones y configuración.

MÉTODO	DESCRIPCIÓN	DEF.
errorMessage	Dado un código de error, devuelve el texto de error asociado.	Def
setOption	Permite establecer las opciones de renderización.	Def
setCacheRoot	Establece el directorio de trabajo de caché, también conocido como directorio de plantillas "preparadas". Puede definirse mediante el constructor de la clase.	Def
setRoot	Permite establece la carpeta raíz de las plantillas. Puede definirse mediante el constructor de la clase.	Def
setCallbackFunction	Establece una relación entre una función de plantilla y una función <i>PHP</i> , sustituyendo la primera por el texto devuelto por esta segunda.	Def