

[\[Indice\]](#)

## Qué son los frames

Los frames (en inglés *frame* = cuadro, bastidor o marco) es un elemento implementado por Netscape, que permite dividir la pantalla en varias áreas independientes unas de otras, y por tanto con contenidos distintos, aunque puedan estar relacionados. No hay límites para el contenido de cada una de estas áreas: tienen las mismas propiedades que la pantalla completa normal, tal y como la conocemos. No hay que confundir los frames con las tablas. Su apariencia, a veces, puede ser similar, pero mientras el contenido de la celda de una tabla es fijo, en un área de pantalla creado por el elemento **FRAME** se dispone de todos los recursos del HTML. Es una zona viva.

Las páginas que contienen una definición de **FRAME** **no pueden** contener el elemento **BODY** ni ninguno de los elementos que habitualmente aparecen en el **BODY** **antes** del elemento **FRAMESET** que es el que define la creación del **FRAME**. Si esto no se cumple, el **FRAME** será ignorado.

Algunos visualizadores no soportan los frames. Para que nuestra página con frames no resulte opaca a ellos, se utilizará el elemento **NOFRAMES** que permite ofrecer un texto alternativo en entorno normal.

Los frames permiten una flexibilidad de presentación extraordinaria, y para contenidos muy complejos, de difícil estructura por los medios convencionales del HTML, son casi insustituibles. De nuevo se presenta la duda de para qué visualizador es preferible programar nuestro WWW.

Por supuesto, los frames son parametrizables en cuanto a tamaño y número de áreas, si éstas se pueden redimensionar por el usuario o son fijas, si tienen o no barras de scrolling, se pueden anidar, relacionar sus contenidos, etc. Veamos algunos ejemplos prácticos y su sintaxis:

### Un frame básico. Sintaxis general

En general, todas las páginas que contengan definiciones de frames, se comportan como si fuesen llamadores o "lanzaderas", y deberán ser más o menos así:

```
<HTML>
<HEAD><TITLE> Mi titulo </TITLE></HEAD>

<FRAMESET>
  <NOFRAMES>
    <BODY>
      Su visualizador no soporta frames. Pulse
      <A HREF="indice.htm">aquí </A> para volver.
    </BODY>
  </NOFRAMES>

  <FRAME SRC="pagina1.htm" >
  <FRAME SRC="pagina2.htm" >

</FRAMESET>

</HTML>
```

Fíjate en que no aparece el elemento **<BODY>** en su posición habitual, sino dentro de un elemento especial que se activa sólo cuando el visualizador no soporta los frames: **NOFRAMES**. Si no declaras el área **NOFRAMES** y el visualizador no soporta este efecto, no se verá nada. Obviamente, los visualizadores que sí soportan frames ignorarán el contenido del área **NOFRAMES**.

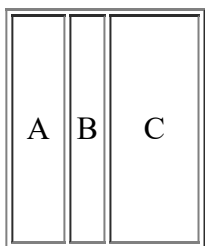
Hasta aquí ya podemos hacernos una idea de cómo funcionan los frames: Lo primero es crear una minipágina con la definición del frame, viene a ser algo así como una "lanzadera" y sólo contiene la definición del frame. Y por último, crear las páginas que constituirán el contenido de cada una de las áreas

definidas en la "lanzadera". En el ejemplo anterior son **pagina1.htm** y **pagina2.htm**, y aquí es donde escribirás tus cosas; es decir que son páginas completamente normales, y que también pueden ser utilizadas de la forma habitual. A su vez, la "lanzadera" puede ser invocada con un link desde cualquier página normal. Como puedes ver, no es obligado crear todo tu documento con frames: puedes utilizarlos solamente allí donde sea necesario, si quieres.

Es posible definir cualquier combinación de áreas verticales y horizontales. La clave está en combinar adecuadamente los anidamientos del elemento **<FRAMESET>** con sus atributos **COLS** o **ROWS** según interese. Como puedes ver, el más importante es el primer **<FRAMESET>**, ya que es el que define cómo va a ser "troceada" la página inicialmente, si en porciones verticales u horizontales, y sobre esta base se deberán definir todos los demás anidamientos.

## Un frame de 3 áreas verticales (COLS)

Se vería así:



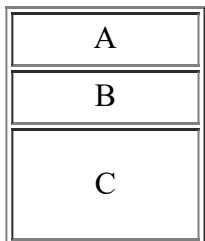
Y se escribiría; así:

```
<FRAMESET COLS=30%,20%,50%>
  <FRAME SRC="a.htm">
  <FRAME SRC="b.htm">
  <FRAME SRC="c.htm">
</FRAMESET>
```

[Ejemplo real](#)

## Un frame de 3 áreas horizontales (ROWS)

Se vería así:



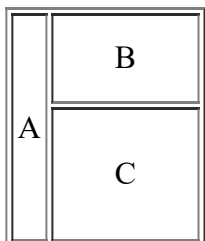
Y se escribiría así:

```
<FRAMESET ROWS=25%,25%,50%>
  <FRAME SRC="a.htm">
  <FRAME SRC="b.htm">
  <FRAME SRC="c.htm">
</FRAMESET>
```

[Ejemplo real](#)

## Un frame combinado de un área vertical y dos horizontales

Se vería así:



Y se escribiría así:

```
<FRAMESET COLS=20%,*>
  <FRAME SRC="a.htm">
  <FRAMESET ROWS=40%,*>
    <FRAME SRC="b.htm">
    <FRAME SRC="c.htm">
  </FRAMESET>
</FRAMESET>
```

[Ejemplo real](#)

## Un frame combinado de dos áreas horizontales, la de abajo dividida en dos verticales

Se vería así:



Y se escribiría así:



```
<FRAMESET ROWS=50%,*>
  <FRAME SRC="a.htm">
    <FRAMESET COLS=50%,*>
      <FRAME SRC="b.htm">
      <FRAME SRC="c.htm">
    </FRAMESET>
  </FRAMESET>
```

### [Ejemplo real](#)

Habrás podido comprobar que los parámetros que dimensionan los frames actúan al presentar la página inicialmente. Después, si el usuario lo desea, puede redimensionarlos como quiera. Esta es la opción por defecto. Si no se quiere permitir el redimensionado, se aplica el atributo **NORESIZE** al elemento FRAME. Así:

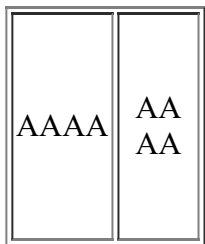
```
<FRAME NORESIZE SRC.....>
```

Algo parecido ocurre con las barras de scrolling. Están regidas por el atributo **SCROLLING**, que puede valer **YES NO** o **AUTO**. Por defecto es **AUTO**. Con esta opción el visualizador decide, en función del contenido, si son necesarias las barras o no. Con **YES** las pondrá siempre, aunque no sean necesarias, y con **NO** no las pondrá nunca, aunque sean necesarias.

```
<FRAME SCROLLING=YES ....>
```

## Un frame con dos áreas verticales. Una normal, la otra con márgenes forzados para el texto (MARGINWIDTH MARGINHEIGHT)

Se vería así:



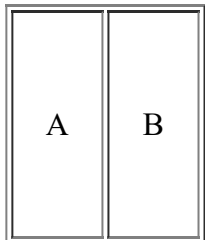
Y se escribiría así:

```
<FRAMESET COLS=50%,50%>
  <FRAME SRC="a.htm">
  <FRAME SRC="a.htm"
    MARGINWIDTH=50
    MARGINHEIGHT=50>
</FRAMESET>
```

### [Ejemplo real](#)

## Un frame con referencias cruzadas (NAME TARGET)

Se vería así:



Y se escribiría así:

```
<FRAMESET COLS=50%,50%>
  <FRAME SRC="aa.htm">
  <FRAME SRC="bb.htm" NAME="VENTANA">
</FRAMESET>
```

### [Ejemplo real](#)

Esto de los cruces es un poco complicado... fíjate en el contenido de cada página que entra en juego. Cuando actives **Ejemplo real** el fichero llamado es **cruzado.htm**. Pruébalo.

Fichero	Contenido	Acción
aa.htm	<A HREF="cc.htm" TARGET="VENTANA">	Pone CC en el frame derecho
	<A HREF="cc.htm">	Trae CC
bb.htm	Esto es "VENTANA"	No hace nada

<b>cc.htm</b>	<A HREF=" <b>bb.htm</b> " TARGET="VENTANA">	Pone BB en el frame derecho porque se llama "VENTANA"
	<A HREF=" <b>aa.htm</b> ">	Trae AA
<b>cruzado.htm</b>	<FRAMESET>	-
	<FRAME SRC=" <b>aa.htm</b> ">	Pone AA en el frame izqdo.
	<FRAME SRC=" <b>bb.htm</b> " NAME="VENTANA">	Pone BB en el frame derecho y le da como nombre "VENTANA"
	</FRAMESET>	-

Hasta aquí se ha utilizado el nombre de "VENTANA" como destino del atributo TARGET. Este nombre, como ya sabes, es el que hemos definido en el código de este ejemplo. Si en lugar de poner el nombre de destino que se ha definido en el FRAMESET, pones otro cualquiera, el visualizador lo que hace es abrir otra ventana y colocar allí el contenido de la página. Evidentemente, cuantas más ejecuciones del navegador haya en marcha, más memoria del ordenador se necesita, y abrir una nueva ventana, implica ejecutar otra vez el navegador. Recuerda cuando hagas esto, que algunos no andan muy sobrados de memoria....

La versión 3.0 de Netscape, implementa dos nuevas posibilidades de los frames: la de darle color a la barra de separación de los distintos frames o la de que no se vea dicha barra. Para ello se utilizan los nuevos atributos de FRAMESET y FRAME:

**FRAMEBORDER** se utiliza como atributo de FRAMESET y establece si serán visibles los bordes del frame o no. Puede tener dos valores: **YES** (por defecto) y **NO**. Si escribes:

```
<FRAMESET FRAMEBORDER=NO COLS=30%,30%,30%>
  <FRAME SRC="a.htm">
  <FRAME SRC="b.htm">
  <FRAME SRC="c.htm">
</FRAMESET>
```

Se obtiene un frame de tres columnas sin barras de separación entre ellas.

[Ejemplo real](#)

**BORDERCOLOR** se utiliza como atributo de FRAME y establece el color de los bordes visibles. Evidentemente, para que funcione, se tendrá que haber establecido **FRAMEBORDER=yes**. Si escribes:

```
<FRAMESET COLS=30%,30%,30%>
  <FRAME BORDERCOLOR="red" SRC="a.htm">
  <FRAME BORDERCOLOR="blue" SRC="b.htm">
  <FRAME SRC="c.htm">
</FRAMESET>
```

Se obtiene un frame de tres columnas con la primera barra de separación roja y la segunda azul. [Ejemplo real](#)

## Salir de una pantalla con frames

Para salir de una pantalla compuesta de frames es necesario definir algún link. En efecto, si estamos en una pantalla con, por ejemplo, dos frames, cualquier link invocado desde cualquiera de ellos, hace que la página llamada aparezca en ese mismo frame. Para volver a una pantalla "normal" sin frames, se puede escribir un link con la sintaxis siguiente:

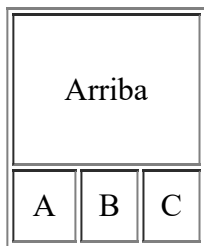
```
<A HREF="indice.htm" TARGET="_parent">Ver índice sin marcos </A>
```

El destino "**\_parent**" indica que presente la página llamada por el link en el navegador que se está ejecutando, sin marcos y sin abrir nuevas ocurrencias del navegador.

Existen otros valores predeterminados para el atributo TARGET: **\_blank** que abre una nueva ventana del navegador, **\_self** que llama el destino del enlace a la misma ventana en que se está (valor por defecto), y **\_top** que funciona igual que **\_parent**.

## Controlar el contenido de los frames con formularios

Hasta aquí hemos visto cómo hacer que los frames reciban sus contenidos siempre mediante links. No es la única forma; también pueden utilizarse formularios para esta tarea. Los formularios, además, nos pueden permitir definir ciertas selecciones previas para que el contenido invocado reúna determinados requisitos, así como elegir el frame de destino. Lo malo de esta técnica es que requiere escribir algo de JavaScript, lo que complica bastante la cosa. Veamos un ejemplo de página con cuatro frames. Por ejemplo, como este:



Y se escribiría así:

```
<FRAMESET ROWS=30%,*>
  <FRAME SRC="fraarriba.htm" NAME="arriba">
    <FRAMESET COLS=33%,33%,*>
      <FRAME SRC="a.htm" NAME="izquierda">
      <FRAME SRC="b.htm" NAME="centro">
      <FRAME SRC="c.htm" NAME="derecha">
    </FRAMESET>
</FRAMESET>
```

### [Ejemplo real](#)

Puesto que se va a poder elegir el frame de destino, es importante que todos ellos tengan nombre. Este es el código del frame **arriba**, que reside en la página que carga: **fraarriba.htm**

```
<HTML>
<HEAD><TITLE>Frames y formularios</TITLE></HEAD>
<BODY>
<CENTER><H2>FRAMES Y FORMULARIOS</H2></CENTER>

<TABLE BORDER=0 WIDTH="100%"><TR>

<TD ALIGN=CENTER>
<FORM>
<SELECT NAME="list">
<OPTION VALUE="a.htm" >Contenido A
<OPTION VALUE="b.htm" >Contenido B
<OPTION VALUE="c.htm" >Contenido C
</SELECT>
<BR>
<INPUT TYPE=button VALUE="Frame izquierdo" onClick=
  "parent.izquierda.location.href=
    this.form.list.options[this.form.list.selectedIndex].value">
</FORM>
</TD>

<TD ALIGN=CENTER>
<FORM>
<SELECT NAME="list">
<OPTION VALUE="a.htm" >Contenido A
<OPTION VALUE="b.htm" >Contenido B
<OPTION VALUE="c.htm" >Contenido C
</SELECT>
<BR>
<INPUT TYPE=button VALUE="Frame central" onClick=
  "parent.centro.location.href=
    this.form.list.options[this.form.list.selectedIndex].value">
</FORM>
</TD>
```

```

<TD ALIGN=Center>
<FORM>
<SELECT NAME="list">
<OPTION VALUE="a.htm" >Contenido A
<OPTION VALUE="b.htm" >Contenido B
<OPTION VALUE="c.htm" >Contenido C
</SELECT>
<BR>
<INPUT TYPE=button VALUE="Frame derecho" onClick=
    "parent.derecha.location.href=
    this.form.list.options[this.form.list.selectedIndex].value">
</FORM>
</TD></TR>
</TABLE>

</BODY>
</HTML>

```

Como puedes ver, todo reside en el código JavaScript que se ejecuta al pulsar el botón de cualquiera de los tres formularios definidos en el frame **arriba**. Mediante este código se captura un evento: **onClick** que se produce al pulsar con el puntero del ratón en el botón del formulario. La primera parte de la instrucción ejecutada: `parent.derecha.location.href=` es la que establece en qué frame se va a cargar la página seleccionada en la lista desplegable, en este caso el llamado **derecha**. La segunda parte de la instrucción: `this.form.list.options[this.form.list.selectedIndex].value` es la que indica qué página hay que cargar en el frame definido en la primera parte de la instrucción. Si... esto suena a muy complicado. No es fácil entender la compleja estructura de objetos del navegador. Para más detalle, puedes ver la sección sobre JavaScript, en particular, y toda la sección de programación en general, de esta guía.

Otra cosa que hay que tener en cuenta cuando se escribe JavaScript, es que distingue entre mayúsculas y minúsculas. En el ejemplo anterior, si los ficheros llamados por el formulario: a.htm b.htm c.htm se han guardado con el nombre en mayúsculas, como A.htm o B.htm, no funcionará si no se invocan escritos de la misma forma. Esto es especialmente importante cuando tengas tus páginas en servidores con sistemas operativos UNIX o LINUX, que son la mayoría de ellos.

## Frames incrustados

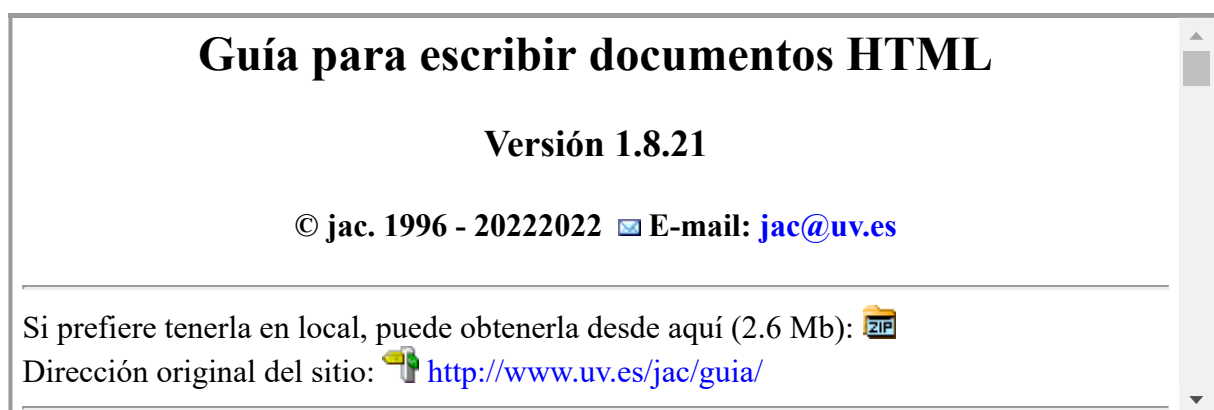
Se puede insertar un frame en cualquier punto de una página, ya sea ésta, a su vez, un frame o una página normal. Su apariencia recuerda a los gráficos y a los formularios, pero su contenido es el de una página independiente. Por ejemplo, si se escribe:

```

<IFRAME NAME="indice" SRC="indice.htm" WIDTH="300" HEIGHT="100" FRAMEBORDER="1"
MARGINWIDTH="5" MARGINHEIGHT="5" SCROLLING="Auto"></IFRAME>

```

Y este sería el resultado:



Analicemos sus atributos:

- **NAME** Como ya habrás supuesto, es el nombre del frame incrustado. Puede omitirse si no es necesario.
  - **SRC** Nombre de la página que se mostrará en el frame.
  - **WIDTH** Número de pixels que tendrá el frame de ancho. También puede indicarse en tanto por ciento respecto al ancho total de la ventana, como se hace en los frames convencionales.
  - **HEIGHT** Número de pixels que tendrá el frame de alto. También puede indicarse en tanto por ciento respecto al alto total de la ventana.
  - **FRAMEBORDER** Indica si se debe mostrar el borde del frame. **1** = si (por defecto) **0** = no.
  - **MARGINWIDTH** Indica el número de pixels que tendrá el ancho de los márgenes interiores izquierdo y derecho del frame respecto a su contenido.
  - **MARGINHEIGHT** Indica el número de pixels que tendrá el ancho de los márgenes interiores superior e inferior del frame respecto a su contenido.
  - **SCROLLING** Indica si hay que mostrar barras de scroll dentro del frame incrustado. Sus valores pueden ser: **Yes**, **No** y **Auto** (por defecto).
- 

[\[Indice\]](#)