

## UT2. HTML BÁSICO

---

1. Qué es HTML .....	2
2. Una página básica .....	2
3. Dando forma al texto .....	7
Listas .....	15
Comentarios no visibles en la pantalla.....	23
4. Enlaces con otras páginas.....	25
Estructura de los enlaces.....	25
Tipos de enlaces .....	25
Enlaces dentro de la misma página .....	25
Enlaces con otra página nuestra.....	29
Enlaces con una página fuera de nuestro sistema.....	31
Enlaces con una dirección de email .....	32
Atributos de los enlaces.....	34
5. Imágenes .....	36
Atributos en las imagenes.....	43
6. Caracteres especiales.....	43
7. Bibliografía .....	47
8. Curiosidades .....	47

### 1. Qué es HTML

El HTML (**Hyper Text Markup Language**) es el lenguaje con el que se escriben las **páginas web**. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por **etiquetas**, que marcan el inicio y el fin de cada elemento del documento.

Un documento hipertexto no sólo se compone de texto, puede contener imágenes, sonido, vídeos, etc., por lo que el resultado puede considerarse como un documento multimedia.

Los documentos HTML deben tener la extensión **html** o **htm**, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web).

Los **navegadores** se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado.

### 2. Una página básica

El principio esencial del lenguaje HTML (HyperText Markup Language) es el uso de las etiquetas (tags). Funcionan de la siguiente manera:

**<XXX>** Este es el inicio de una etiqueta.

**</XXX>** Este es el cierre de una etiqueta.

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas, indiferentemente. Por claridad, se usarán en este manual solamente las mayúsculas.

## UT2. HTML BÁSICO

---

...Nosotros vamos a optar en el bloc de notas por las minúsculas por ser lo que se está imponiendo de acuerdo con las últimas normas.

Lo que haya entre ambas etiquetas estará influenciada por ellas. Por ejemplo, todo el documento HTML debe estar entre las etiquetas `<HTML>` y `</HTML>`:

`<HTML> [Todo el documento] </HTML>`

El documento en sí está dividido en dos zonas principales:

El encabezado, comprendido entre las etiquetas `<HEAD>` y `</HEAD>`. La cabecera de la página se utiliza para agrupar información sobre ella, como puede ser el título. La etiqueta `<head>` va justo debajo de la etiqueta `<html>`.

El cuerpo, comprendido entre las etiquetas `<BODY>` y `</BODY>`

Dentro del encabezado hay información del documento, que no se ve en la pantalla principal, principalmente el título del documento, comprendido entre las etiquetas `<TITLE>` y `</TITLE>`. El título debe ser breve y descriptivo de su contenido, pues será lo que vean los demás cuando añadan nuestra página a su bookmark (o agenda de direcciones).

Dentro del cuerpo está todo lo que queremos que aparezca en la pantalla principal (texto, imágenes, etc.)

Por tanto, la estructura queda de esta manera:

`<HTML>`

`<HEAD>`

`<TITLE> Título de la página </TITLE>`

`</HEAD>`

`<BODY>`

`[Aquí van las etiquetas que visualizan la página]`

`</BODY>`

`</HTML>`

Antes de crear nuestra primera página, unas consideraciones sobre el texto: Cuando escribimos en el documento el texto que queremos que aparezca en la pantalla, veremos que éste se acomoda a ella, sin que tengamos que pulsar el retorno del carro. Si queremos separar el texto en distintos párrafos debemos usar la etiqueta `<P>`, (que no tiene su correspondiente etiqueta de cierre `</P>`)

El texto puede tener unas cabeceras, comprendidas entre las etiquetas `<H1>` y `</H1>`, `<H2>` y `</H2>`, etc. (hasta el número 6), siendo el número indicativo del tamaño. El tamaño mayor es el correspondiente al número 1. Puedes experimentar en el ejemplo que sigue, cambiando el número para comprobar el efecto que se logra.

Una etiqueta muy interesante es la de centrado `<CENTER>` y `</CENTER>` (no la soportan todos los navegadores, aunque sí la mayoría de ellos). Nos centra todo lo que esté dentro de ella, ya sea texto, imágenes, etc.

También tenemos los separadores (horizontal rules), que se consiguen con la etiqueta `<HR>` (no existe la correspondiente de cierre). Con ella se obtiene una raya horizontal tan ancha como la pantalla, y con la apariencia de estar embutida sobre el fondo, como se puede observar a continuación:

### Ejemplo1:

En el procesador de texto copiamos lo siguiente:

`<HTML>`

`<HEAD>`

`<TITLE> Mi Web </TITLE>`

</HEAD>

<BODY>

<H1> <CENTER> Primera pagina </CENTER> </H1>

<HR>

Esta es mi primera página, aunque todavía es muy sencilla. Como el lenguaje HTML no es difícil, pronto estaré en condiciones de hacer cosas más interesantes.

<P> Aquí va un segundo párrafo.

</BODY>

</HTML>

Ejemplo 2: Página con dos párrafos

<HTML>

<HEAD>

<TITLE>El primer documento HTML</TITLE>

</HEAD>

<BODY>

<p align=left>El lenguaje HTML es tan sencillo > que prácticamente se entiende sin estudiar el significado de sus etiquetas principales.

<p align=right>Ademas de textos en negrita, tambien se pueden poner en cursiva o tachados.

</BODY>

</HTML>

### Ejemplo 3: Página con dos párrafos

```
<HTML>  
  
<HEAD>  
  
<TITLE>El primer documento HTML</TITLE>  
  
</HEAD>  
  
<BODY>  
  
<p align=center>El lenguaje HTML es tan sencillo > que prácticamente se  
entiende sin estudiar el significado de sus etiquetas principales.  
  
<hr align=center width="75%" size="3" color="red">  
  
  
  
<p align=justify>Ademas de textos en negrita, tambien se pueden poner en  
cursiva o tachados.  
  
</BODY>  
  
</HTML>
```

Las líneas en blanco y las indentaciones del texto se han puesto para mayor claridad, pero no son necesarias. De hecho, podría estar todo en una sola línea. Lo importante es el orden correcto de las etiquetas. Por cierto, una etiqueta puede

estar anidada dentro de otra. Véase en el ejemplo cómo lo está la etiqueta `<CENTER>` dentro de la etiqueta `<H1>`. Es muy importante, en estos casos, que las etiquetas de inicio y de cierre vayan en el orden correcto, pues de lo contrario se producirían errores.

Obsérvese además la falta de los acentos. Se hablará más adelante sobre el motivo de ello.

Guardamos el fichero en el procesador de textos con el nombre de `mipag1.html` y lo cargamos en el navegador.

Habrà un fichero distinto para cada capítulo; conviene crear una carpeta específica e irlos guardando en ella, para poder repasar lo aprendido, aparte de que pueden ser necesarios para ejecutar otros ejemplos prácticos.

### 3. Dando forma al texto

Como hemos visto en el ejemplo del capítulo anterior, cuando queremos poner un texto sin ninguna característica especial, lo ponemos directamente. Únicamente, la separación entre párrafos (dejando una línea en blanco) la conseguimos con la etiqueta `<P>`.

Si queremos separar los párrafos, o cualquier otra cosa, pero sin dejar una línea en blanco, usamos una etiqueta parecida `<BR>` (break, o romper). Tampoco tiene etiqueta de cierre.

Si queremos obtener múltiples líneas en blanco no basta con repetir la etiqueta `<P>`, sino que hay que combinarla con la etiqueta `<BR>`. Así por ejemplo, si queremos obtener cuatro líneas en blanco, pondríamos:

`<P>`

`<BR>`

<BR>

<BR>

<BR>

### ejemplo

<HTML>

<HEAD>

<TITLE>El primer documento HTML</TITLE>

</HEAD>

<BODY>

<p>El lenguaje HTML es tan sencillo > que prácticamente se entiende sin estudiar el significado de sus etiquetas principales.

<br>

<br>

<br>

<p>Ademas de textos en negrita, tambien se pueden poner en cursiva o tachados.

</BODY>

</HTML>

Al escribir el texto, si ponemos más de un espacio en blanco entre dos palabras observamos que el navegador sólo reconoce uno de ellos. Si queremos forzarle a que lo haga, debemos poner el código "&nbsp;" (non-breaking space).

Para destacar alguna parte del texto se pueden usar:



## UT2. HTML BÁSICO

---

**<B> y </B>** para poner algo en negrita (bold).

**<I> y </I>** para poner algo en cursiva (italic).

**<U> y </U>** para subrayar texto

**<DEL> y </DEL>** para tachar texto

Ejemplo:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>El primer documento HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<p>El lenguaje HTML es tan sencillo que prácticamente se entiende sin  
estudiar el significado de sus etiquetas principales.
```

```
<p>Ademas de textos <b>en negrita</b>, tambien se pueden poner en  
<i>cursiva</i> <u>subrayados</u> o <del>tachados</del>.
```

```
</BODY>
```

```
</HTML>
```

Otra etiqueta interesante es **<PRE> y </PRE>**. El texto que se encuentre entre ella estará preformateado, es decir que aparecerá como si hubiera sido escrito con una máquina de escribir, con una fuente de espaciado fijo (tipo Courier). Además se respetarán los espacios en blanco y retornos del carro, tal como estaban en

nuestro documento HTML (lo cual no ocurre normalmente, como hemos visto anteriormente). Es muy apropiada para confeccionar tablas y otros documentos similares.

### Ejemplo:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>El primer documento HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Texto preformateado</H1>
```

Diferencia entre un texto normal y un texto preformateado. En el texto preformateado, se respetan los caracteres y los espacios.<br>

iiiiii<br>

wwwwww<br>

```
<PRE>
```

Diferencia entre un texto normal y un texto preformateado. En el texto preformateado, se respetan los caracteres y los espacios.<br>

iiiiii<br>

wwwwww<br>

```
</PRE>
```

```
</BODY>
```

```
</HTML>
```

Con la etiqueta `<TT>` y `</TT>` conseguimos también que el texto tenga un tamaño menor y la apariencia de los caracteres de una máquina de escribir (typewriter). La diferencia con la anterior es que no preformatea el texto, sino que únicamente cambia su apariencia. Conseguimos un espaciado simple. No sirve la etiqueta `br`

**Ejemplo:**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>El primer documento HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Texto preformateado</H1>
```

```
<tt>Diferencia entre un texto normal y  
un texto preformateado. En el texto  
preformateado, se respetan los  
caracteres y los espacios.<br>
```

```
iiii<br>
```

```
wwwww<br> </tt>
```

```
<PRE>
```

```
Diferencia entre un texto normal y  
un texto preformateado. En el texto  
preformateado, se respetan los  
caracteres y los espacios.<br>
```

```
iiii<br>
```

```
wwwww<br>
```

```
</PRE>
```

```
</BODY>
</HTML>
```

La etiqueta **<BLOCKQUOTE>** y **</BLOCKQUOTE>** sirve para poder dar formato con sangría a un párrafo de un documento HTML. Con esta etiqueta se definen citas largas. Este párrafo está escrito entre ambas etiquetas.

Ejemplo:

```
<HTML>
<HEAD>
<TITLE>El primer documento HTML</TITLE>
</HEAD>
<BODY>
<H1>Texto preformateado</H1>
<tt>Diferencia entre un texto normal y
un texto preformateado. En el texto
preformateado, se respetan los
car&aacute;cteres y los espacios.<br>
iiiiii<br>
wwwwww<br> </tt>
<PRE>
Diferencia entre un texto normal y
un texto preformateado. En el texto
preformateado, se respetan los
car&aacute;cteres y los espacios.<br>
iiiiii<br>
wwwwww<br>
</PRE>
<p><BLOCKQUOTE>La noche pasó lentamente. Salió el sol. Los hobbits se
levantaron bastante tarde y la mañana prosiguió. Se solicitó el concurso de
gente, que recibió orden de despejar los pabellones y quitar mesas, sillas,
cucharas, cuchillos, botellas, platos, linternas, macetas de arbustos en flor,
migajas, papeles, carteras, pañuelos y guantes olvidados, y alimentos no
consumidos, que eran muy pocos.</BLOCKQUOTE>
</BODY>
</HTML>
```

**Ejercicio:** Ejercicio: crear una página con tres párrafos teniendo cada párrafo una cabecera de distinto tamaño y justificación. Dejando distintos espacios entre los párrafos y las cabeceras y los párrafos. Con frases en negrita y cursiva subrayadas y tachadas y con texto preformateado y sangría.

En las fórmulas matemáticas puede interesar poder escribir índices y subíndices, que se consiguen con las etiquetas `<SUP>` `</SUP>` y `<SUB>` `</SUB>` respectivamente.

Así, por ejemplo:

$m^2$  se consigue de la siguiente manera: `m<SUP>2</SUP>`

$V_2$  se consigue con: `V<SUB>2</SUB>`

**Ejercicio:** Realiza una página web donde se muestre la información que

aparece a continuación, con las mismas características: textos en cursiva y negrita, subrayados, superíndices y subíndices, tabulaciones al principio de párrafo, tamaños de la letra, etc.

### Sistemas de Ecuaciones

Podemos definir las **ecuaciones** como igualdades entre *expresiones algebraicas* (sucesión de términos constituidos por números y letras, cada término separado del otro por un signo "+"ó "-"), en la que intervienen una o más letras, llamadas *incógnitas*, cuyo valor hay que averiguar. Las expresiones que están a ambos lados del signo igual son los miembros de la ecuación: *primer miembro* el de la izquierda, *segundo miembro* el de la derecha. Se denomina *solución* de una ecuación a un valor o conjunto de valores de la incógnita ( $x$ ), para los cuales se verifica la igualdad. Por ejemplo:

- a)  $x^2 + y^2 + 5 = 0$ , es una ecuación con dos incógnitas sin solución.
- b)  $3x^3 + 5x^2 + 3x + 2 = 0$ , es una ecuación polinómica.
- c)  $(x - 5)^2 + 3 = x^2 - 1$ , es una ecuación lineal, es decir, de primer grado.
- d) Las ecuaciones polinómicas de segundo grado que responden a la estructura:  $ax^2 + bx + c = 0$ , se las denomina cuadráticas. Son ecuaciones de este tipo:  $x^2 + 5x + 3 = 0$ , ó  $(x - 2)^2 + 7x = 5 + x$ . Generalmente las ecuaciones cuadráticas se presentan de la forma polinómica:  $f(x) = ax^2 + bx + c$ .

### Logaritmos

El *logaritmo* en base  $a$  de un número  $n$ , es otro número  $b$ , tal que cumple esta ecuación:  $a^b = n$ . Dicho matemáticamente:  $\log_a n = b \implies a^b = n$ . Supongamos que el logaritmo en base  $a$  de un número  $n_1$  sea  $b_1$  ( $\log_a n_1 = b_1$ ). Entonces  $a^{b_1} = n_1$ . Supongamos que el logaritmo en base  $a$  de un número  $n_2$  sea  $b_2$  ( $\log_a n_2 = b_2$ ). Entonces  $a^{b_2} = n_2$ . Supongamos que nos piden que calculemos el logaritmo del producto  $n_1 \cdot n_2$ , y digamos que es  $b$ . Si tenemos en cuenta las igualdades anteriores nos queda:

$$\begin{aligned} \log_a n_1 \cdot n_2 &= \log_a a^{b_1} \cdot a^{b_2} = b \\ a^b &= a^{b_1} \cdot a^{b_2} = a^{b_1 + b_2} \end{aligned}$$

### Listas

A menudo nos interesará presentar las cosas en forma de listas. Podemos escoger entre tres tipos distintos:

1. Listas desordenadas (no numeradas)
2. Listas ordenadas (numeradas)
3. Listas de definición.

**Las listas desordenadas** (unordered lists) sirven para presentar cosas que, por no tener un orden determinado, no necesitan ir precedidas por un número. Su estructura es la siguiente:

<pre>&lt;UL&gt;   &lt;LI&gt; Una cosa &lt;/LI&gt;   &lt;LI&gt; Otra cosa &lt;/LI&gt;   &lt;LI&gt; Otra más &lt;/LI&gt;   &lt;LI&gt; Etc. &lt;/LI&gt; &lt;/UL&gt;</pre>	Es decir, toda la lista está dentro de la etiqueta <code>&lt;UL&gt;</code> y <code>&lt;/UL&gt;</code> , y luego cada cosa va precedida de la etiqueta <code>&lt;LI&gt;</code> <code>&lt;/LI&gt;</code> (list item).
--	---

Ejemplo1:

```
<html>
<head>
</head>
<body>
<UL>
<LI> Mamíferos</LI>
<LI> Peces</LI>

<LI> Aves</LI>
</UL>
</body>
</html>
```

### Ejemplo 2:

```
<html>

<head>

</head>

<body>

<UL type=circle>
<LI> Mamíferos</LI>
<LI> Peces</LI>

<LI> Aves</LI>
</UL>

</body>

</html>
```

### Ejemplo 3:

```
<html>

<head>

</head>

<body>

<UL type=square>
<LI> Mamíferos</LI>
<LI> Peces</LI>
<LI> Aves</LI>
</UL>

</body>

</html>
```



## UT2. HTML BÁSICO

---

Se puede anidar una lista dentro de otra. Por ejemplo:

```
<html>

<head>

</head>

<body>

<UL>
<LI> Mamíferos</LI>
<LI> Peces</LI>
    <UL>
        <LI> Sardina</LI>
        <LI> Bacalao</LI>
    </UL>
<LI> Aves</LI>
</UL>
</body>

</html>
```

**Las listas ordenadas** (ordered lists) sirven para presentar cosas en un orden determinado. Su estructura es muy similar a la anterior. La diferencia estriba en que en el resultado aparecerá automáticamente un número correlativo para cada cosa.

Ejemplo:

```
<html>

<head>

</head>

<body>

<ol>

<li>Rodriguez Pablo</li>

<li>Gonzalez Raul</li>

<li>Lopez Hector</li>

</ol>

</body>

</html>
```

Ejemplo 2:

```
<html>

<head>

</head>

<body>

<ol type= A>

<li>Rodriguez Pablo</li>

<li>Gonzalez Raul</li>

<li>Lopez Hector</li>

</ol>

</body>

</html>
```

### Ejemplo 3:

```
<html>

<head>

</head>

<body>

<ol type= a>

<li>Rodriguez Pablo</li>

<li>Gonzalez Raul</li>

<li>Lopez Hector</li>

</ol>

</body>

</html>
```

### Ejemplo 4:

```
<html>

<head>

</head>

<body>

<ol type= I>

<li>Rodriguez Pablo</li>

<li>Gonzalez Raul</li>

<li>Lopez Hector</li>

</ol>

</body>

</html>
```

### Ejemplo 5:

```
<html>

<head>

</head>

<body>

<ol type= I>

<li>Rodriguez Pablo</li>

<li>Gonzalez Raul</li>

<li>Lopez Hector</li>

</ol>

</body>

</html>
```

Al igual que las listas desordenadas, también se pueden anidar las listas ordenadas.

### Ejemplo:

```
<html>

<head>

</head>

<body>

<ol>

<li>Argentina</li>

  <ol>

    <li> La Nación</li>

    <li>Clarín</li>

    <li>Página 12</li>
```

</ol>

<li>España</li>

<ol>

<li>Abc</li>

<li>El pais</li>

</ol>

</ol>

</html>

El tercer tipo lo forman las listas de definición. Como su nombre indica, son apropiadas para glosarios (o definiciones de términos). Toda la lista debe ir englobada entre las etiquetas <DL>y </DL>. Y a diferencia de las dos que hemos visto, cada renglón de la lista tiene dos partes:

- 1) El nombre de la cosa a definir , que se consigue con la etiqueta <DT> (definition term) y 2) la definición de dicha cosa, que se consigue con la etiqueta <DD> (definition definition).

<DL>

<DT> Una cosa a definir</DT>

<DD> La definición de esta cosa</DD>

<DT> Otra cosa a definir</DT>

<DD> La definición de esta otra cosa</DD>

</DL>

Ejemplo:

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<dl>
```

```
<dt>C++</dt>
```

```
<dd>Es un lenguaje de programación, diseñado a mediados de  
los años 1980, por Bjarne Stroustrup, como extensión del lenguaje  
de programación C.</dd>
```

```
<dt>Java</dt>
```

```
<dd>Es un lenguaje de programación orientado a objetos desarrollado  
por Sun Microsystems a principios de los 90.</dd>
```

```
<dt>JavaScript</dt>
```

```
<dd>Es un lenguaje interpretado, es decir, que no requiere compilación,  
utilizado principalmente en páginas web, con una sintaxis semejante a la  
del lenguaje C.</dd>
```

```
</dl>
```

```
</body>
```

```
</html>
```

### *Comentarios no visibles en la pantalla*

A veces es muy útil escribir comentarios en el documento HTML sobre el código que escribimos, que nos pueden servir para recordar posteriormente sobre lo que hicimos, y que no queremos que se vean en pantalla.

Esto se consigue encerrando dichos comentarios entre estos dos símbolos:

`<!-- y -->`

Ejemplo:

`<!-- Esto es un comentario al código que no se verá en pantalla -->`

Ejemplo práctico

En el procesador de textos copiamos:

`<HTML>`

`<HEAD>`

`<TITLE> Mi pagina del Web - 2 </TITLE>`

`</HEAD>`

`<BODY>`

`<CENTER>`

`<H1> Mis aficiones </H1>`

`</CENTER>`

`<!-- Esto es un comentario al código que no se verá en pantalla -->`

`<HR>`

Sin un orden particular, mis `<B>` aficiones `</B>` son las siguientes:

<UL>

<LI> El cine</LI>

<LI> El deporte</LI>

<UL>

<LI> Natacion</LI>

<LI> Baloncesto</LI>

</UL>

<LI> La musica</LI>

</UL>

La música que más me gusta es <I> (en orden de preferencia): </I>

<OL>

<LI> El rock</LI>

<LI> El jazz</LI>

<LI> La música clásica</LI>

</OL>

</BODY>

</HTML>

Guardamos el fichero de texto con el nombre mipag2.html y lo cargamos en el navegador.



### 4. Enlaces con otras páginas

La característica que más ha influido en el espectacular éxito del Web (o tela de araña) ha sido, aparte la de su carácter multimedia, la posibilidad de unir los distintos documentos repartidos por todo el mundo por medio de enlaces hipertexto.

#### *Estructura de los enlaces*

En general, los enlaces tienen la siguiente estructura:

`<A HREF="xxx"> yyy </A>`

donde **xxx** es el destino del enlace (Obsérvese las comillas).

**yyy** es el texto indicativo en la pantalla del enlace (con un color especial y generalmente subrayado).

#### *Tipos de enlaces*

Vamos a distinguir cuatro tipos de enlaces:

1. Enlaces dentro de la misma página
2. Enlaces con otra página nuestra
3. Enlaces con una página fuera de nuestro sistema
4. Enlaces con una dirección de email

#### **Enlaces dentro de la misma página**

A veces, en el caso de documentos (o páginas) muy extensos, nos puede interesar dar un salto desde una posición a otra determinada.

## UT2. HTML BÁSICO

---

En este caso, lo que antes hemos llamado XXX, es decir, el destino del enlace, en este caso el sitio dentro de la página a donde queremos saltar, se sustituye por #marca (la palabra marca puede ser cualquier palabra que queramos). Lo que hemos llamado antes YYY es la palabra (o palabras) que aparecerán en la pantalla en color (en forma de hipertexto). Su estructura es, entonces:

```
<A HREF="#marca"> YYY </A>
```

Y en el sitio exacto a donde queremos saltar, debemos poner la siguiente etiqueta:

```
<A NAME="marca"> </A>
```

Por ejemplo, si quiero saltar desde aquí a la pantalla final, pongo la siguiente etiqueta:

```
<A HREF="#final"> Pulsa para ir al final</A>
```

Que resulta como: Pulsa para ir al final (Puedes comprobar cómo salta a la pantalla final)

Y en el final del documento he puesto esta otra etiqueta:

```
<A NAME="final"> </A>
```

**Ejemplo:**

```
<html>
```

```
<head>
```

```
</head>
```

```
body>
```

```
<h1>Tutorial de MySQL</h1>
```

```
<a href="#introduccion">Introducción</a><br>
```

```
<a href="#mostrarbasedatos">show databases</a><br>
```

`<a href="#creaciontabla">Creación de una tabla y mostrar sus campos</a><br>`

`<a href="#cargarregistros">Carga de registros a una tabla y su recuperación</a><br>`

`<a name="#introduccion"></a>`

`<h2>Introducción</h2>`

`<p>`

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje de programación para trabajar con base de datos relacionales como MySQL, Oracle, etc.<br>

MySQL es un interpretador de SQL, es un servidor de base de datos.<br>

MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.<br>

Ingresando instrucciones en la línea de comandos o embebidas en un lenguaje como PHP nos comunicamos con el servidor. Cada sentencia debe acabar con punto y coma (;).<br>

La sensibilidad a mayúsculas y minúsculas, es decir, si hace diferencia entre ellas, depende del sistema operativo, Windows no es sensible, pero Linux sí. Por ejemplo Windows interpreta igualmente las siguientes sentencias:<br>

`create database administracion;<br>`

`Create DataBase administracion;<br>`

Pero Linux interpretará como un error la segunda.<br>

Se recomienda usar siempre minúsculas. Es más el sitio `mysqlia.com.ar` está instalado sobre un servidor Linux por lo que todos los ejercicios deberán respetarse mayúsculas y minúsculas.

`<a name="mostrarbasedatos"></a>`

`<h2>show databases</h2>`

`<p>`

Una base de datos es un conjunto de tablas.`<br>`

Una base de datos tiene un nombre con el cual accederemos a ella.`<br>`

Vamos a trabajar en una base de datos ya creada en el sitio, llamada "administracion".`<br>`

Para que el servidor nos muestre las bases de datos existentes, se lo solicitamos enviando la instrucción:`<br>`

`show databases;<br>`

Nos mostrará los nombres de las bases de datos, debe aparecer en este sitio "administracion".`<br>`

`<a name="creaciontabla"></a>`

`<h2>Creación de una tabla y mostrar sus campos</h2>`

`<p>`

Una base de datos almacena sus datos en tablas.`<br>`

Una tabla es una estructura de datos que organiza los datos en columnas y filas; cada columna es un campo (o atributo) y cada fila, un registro. La intersección de una columna con una fila, contiene un dato específico, un solo valor.`<br>`

Cada registro contiene un dato por cada columna de la tabla.`<br>`

Cada campo (columna) debe tener un nombre. El nombre del campo hace referencia a la información que almacenará.<br>

Cada campo (columna) también debe definir el tipo de dato que almacenará.<br>

```
<a name="cargarregistros"></a>
```

```
<h2>Carga de registros a una tabla y su recuperación</h2>
```

```
<p>
```

Usamos "insert into". Especificamos los nombres de los campos entre paréntesis y separados por comas y luego los valores para cada campo, también entre paréntesis y separados por comas.<br>

Es importante ingresar los valores en el mismo orden en que se nombran los campos, si ingresamos los datos en otro orden, no aparece un mensaje de error y los datos se guardan de modo incorrecto.<br>

Note que los datos ingresados, como corresponden a campos de cadenas de caracteres se colocan entre comillas simples. Las comillas simples son OBLIGATORIAS.

```
</body>
```

```
</html>
```

### Enlaces con otra página nuestra

Puede ser que tengamos una sola página. Pero lo más frecuente es que tengamos varias páginas, una inicial (o principal) y otras conectadas a ella, e incluso entre ellas mismas.

Supongamos que queremos enlazar con la página creada en el ejemplo del capítulo anterior, que la hemos llamado mipag2.html. En este caso, simplemente sustituimos lo que hemos llamado

XXX (el destino del enlace) por el nombre del fichero:

**<A HREF="mipag2.html"> Ejemplo del capítulo 2 </A>**

Que resulta como: Ejemplo del capítulo 2 (Puedes comprobar cómo carga esa página)

Si queremos que vaya a un sitio concreto de otra página nuestra en vez de ir al principio de la página, adonde va por defecto, en ese sitio tenemos que colocar una marca (véase la sección anterior), y completar el enlace con la referencia a esa marca.

Lo veremos con el siguiente ejemplo: Quiero poner un enlace desde aquí al capítulo 4, pero a un sitio concreto, donde he puesto la marca **<A NAME="alinear"></A>**. Entonces la etiqueta

tiene que ser: **<A HREF="cap4.html#alinear"> Capitulo 4 </A>** Que resulta: Capitulo 4 (Obsérvese cómo al pulsar el enlace salta al capítulo 4, pero no al comienzo, sino a donde yo quiero que lo haga) .

Una observación importante: Estoy suponiendo que la página en la que estoy escribiendo esta etiqueta y la otra página a la que quiero saltar están en el mismo directorio. Porque pudiera ocurrir que he organizado mi sitio del Web con un directorio principal, y otros subdirectorios auxiliares. Si la página a la que quiero saltar está, p. ej. en el subdirectorio subdir, entonces en la etiqueta tendría que haber puesto **"subdir/mipag2.html"**.

Y a la inversa, si quiero saltar desde una página a otra que está en un directorio anterior, en la etiqueta tendría que haber puesto **"../mipag2.html"**. Esos dos puntos hace que se dirija al directorio anterior. Obsérvese que se debe utilizar el

símbolo / para indicar los subdirectorios, y no este otro \, que es propio únicamente de Windows.

Si nos queremos evitar todas estas complicaciones, podemos tener todo junto en un único directorio, pero esto tiene el inconveniente de que esté todo más desordenado, y sean más difíciles de hacer las futuras modificaciones.

### Enlaces con una página fuera de nuestro sistema

Si queremos enlazar con una página que esté fuera de nuestro sistema (es decir, que esté en un servidor distinto al que soporta nuestra página), es necesario conocer su dirección completa, o URL (Uniform Resource Locator). [localizador uniforme de recursos]. El URL podría ser, además de la dirección de una página del Web, una dirección de ftp, etc.

Una vez conocida la dirección (o URL), lo colocamos en vez de lo que hemos llamado anteriormente xxx (el destino del enlace). Si queremos enlazar con la página de Google (cuyo URL es: <http://www.google.es/>), la etiqueta sería:

```
<A HREF="http://www.google.es"> Página inicial de Google </A>
```

**Ejemplo:**

```
<html>
```

```
<head>
```

```
<title>Un vínculo que se abre en otro página del navegador</title>
```

```
</head>
```

```
<body>
```

```
<a href="http://www.google.es/" target="_blank">Google</a>
```

```
</body>
```

```
</html>
```

Que daría como resultado: Página inicial de Google

### Ejemplo 2

```
<html>

<head>

<title>Un vínculo que se abre en otro página del navegador</title>

</head>

<body link= "#31B404" vlink="#FE2E2E" alink="#013ADF">

<a href="http://www.google.es/" target="_blank">Google</a>

</body>

</html>
```

Daria el resultado anterior pero cambiando el color del link

### Enlaces con una dirección de email

En este caso, sustituimos lo que se ha llamado antes xxx (el destino del enlace) por mailto:seguido de la dirección de email. La estructura de la etiqueta es:

```
<A HREF="mailto: dirección de email"> Texto del enlace </A>
```

Un ejemplo de esto:

Comentarios a <A HREF="mailto: redes.mjvillar@gmail.com">MJose</A>

Pero hay algunos navegadores que no reconocen este tipo de enlace, y lo verían de esta manera:



Comentarios a MJosé

y no tendrían ningún medio de conocer la dirección de email. Por lo tanto, es más seguro poner algo así como: Comentarios a María José en `<A HREF="mailto:`

`redes.mjvillar@gmail.com"> redes.mjvillar@gmail.com </A>` Que resulta:

Comentarios a María José en [redes.mjvillar@gmail.com](mailto:redes.mjvillar@gmail.com)

Es decir, es conveniente, por la razón dicha anteriormente, poner también en el texto del enlace la dirección de email.

### Ejemplo práctico

En el procesador de textos copiamos:

```
<HTML>
<HEAD>
<TITLE> Mi pagina del Web - 3 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Mis paginas favoritas </H1>
</CENTER>
<HR>
```

Estas son mis paginas favoritas:

```
<P><A HREF="http://www.ubuntu-es.org/"> Ubuntu </A>
<BR> <A HREF="http://www.microsoft.com"> Microsoft </A>
<BR> <A HREF="http://www.yahoo.com"> Yahoo! </A>
</BODY>
</HTML>
```

Guardamos el fichero de texto con el nombre mipag3.html y lo cargamos en el navegador.

### Atributos de los enlaces

- **name:** sirve para nombrar un ancla o marcador que será el destino del enlace.
- **href:** sirve para indicar la URI destino del hipervínculo.
- **accesskey:** asigna una tecla única de acceso al hipervínculo.

Browser	Windows	Linux	Mac
Internet Explorer	[Alt] + <i>accesskey</i>	N/A	
Chrome	[Alt] + <i>accesskey</i>	[Alt] + <i>accesskey</i>	[Control] [Alt] + <i>accesskey</i>
Firefox	[Alt] [Shift] + <i>accesskey</i>	[Alt] [Shift] + <i>accesskey</i>	[Control] [Alt] + <i>accesskey</i>
Safari	[Alt] + <i>accesskey</i>	N/A	[Control] [Alt] + <i>accesskey</i>
Opera	Opera 15 or newer: [Alt] + <i>accesskey</i> Opera 12.1 or older: [Shift] [Esc] + <i>accesskey</i>		

`<a href="https://www.w3schools.com/html5" accesskey="h">HTML5</a><br>`

`<a href="https://www.w3schools.com/css3" accesskey="c">CSS3</a>`

- **shape:** establece la forma de la imagen que contiene el hipervínculo.
- **coords:** establece la posición en pantalla de la imagen.

`<object data="planets.jpg" alt="Planets" type="image/jpg" usemap="#Map1">`

```
<map name="Map1">
```

```
<a href="sun.htm" shape="rect" coords="0,0,82,126">The Sun</a>
```

```
<a href="mercur.htm" shape="circle" coords="90,58,3">Mercury</a>
```

```
<a href="venus.htm" shape="circle" coords="124,58,8">Venus</a>
```

```
</map>
```

```
</object>
```

-----

```
<!-- funciona mejor -->
```

```

```

```
<map name="Map1">
```

```
<area href="sun.htm" shape="rect" coords="0,0,82,126" alt="The Sun" />
```

```
<area href="mercur.htm" shape="circle" coords="90,58,3" alt="Mercury" />
```

```
<area href="venus.htm" shape="circle" coords="124,58,8" alt="Venus" />
```

```
</map>
```

- **target:** establece el marco de destino (por defecto la ventana actual).  
Valores posibles: `_blank`, `_self`, `_parent`, `_top`.
- **tabindex:** establece un número de orden. Usando la tecla "TAB" marcamos los enlaces en el orden establecido.

```
<a href="https://www.w3schools.com/" tabindex="2">W3Schools</a><br>
```

```
<a href="http://www.google.com/" tabindex="1">Google</a><br>
```

```
<a href="http://www.microsoft.com/" tabindex="3">Microsoft</a><br>
```

### 5. Imágenes

La etiqueta que nos sirve para incluir imágenes en nuestras páginas del Web es muy similar a la de enlaces a otras páginas, que hemos visto en el capítulo anterior. La única diferencia es que, en lugar de indicar al programa navegador el nombre y la localización de un documento de texto HTML para que lo cargue, se le indica el nombre y la localización de un fichero que contiene una imagen.

La estructura de la etiqueta es:

**<IMG SRC="imagen.jpg">**

Con el comando IMG SRC (image source, fuente de la imagen) se indica que se quiere cargar una imagen llamada imagen.jpg (o el nombre que tenga).

Dentro de la etiqueta se pueden añadir otros comandos, tal como **ALT**

**<IMG SRC="imagen.jpg" ALT="descripción">**

Con el comando ALT se introduce una descripción (una palabra o una frase breve) indicativa de la imagen. Este comando, que en principio se puede omitir, es en beneficio de los que accedan a nuestra página con un programa navegador en forma de texto sólo. Ya que no son capaces de ver la imagen, por lo menos pueden hacerse una idea sobre ella. Pero no es sólo por esto. Hay casos, como veremos más adelante, en los que se utiliza una imagen como enlace a otra página.

Si se omitiera este comando, los que utilizan dichos navegadores no podrían de ninguna manera acceder a esas páginas.

Con respecto a la localización del fichero de esa imagen, se puede decir aquí lo mismo que en el capítulo anterior referente a los enlaces. Si no se indica nada especial, como en el caso que se ha expuesto, quiere decir que el fichero imagen.jpg está en el mismo directorio que el documento HTML que estamos escribiendo. Si no es así, se siguen los mismos criterios que los indicados para los enlaces.

Al igual que una página con la que queremos enlazar puede estar fuera de nuestro sistema, (en cuyo caso había que indicar su URL o dirección completa), podemos cargar una imagen que no esté en nuestro sistema siguiendo el mismo método, es decir, indicar en la etiqueta el URL completo de la imagen. Aunque esto no es muy aconsejable, pues alargaría innecesariamente el tiempo de carga de nuestra página.

Un aspecto muy importante a tener en cuenta es el tamaño de las imágenes, pues una imagen grande supone un fichero grande, y esto puede resultar en un tiempo excesivo de carga, con el consiguiente riesgo de que quien esté intentando cargar nuestra página se canse de esperar, y desista de ello.

Para elegir la posición de la imagen con respecto al texto hay distintas posibilidades. La más sencilla es colocarla entre dos párrafos, con un titular a un lado. Los navegadores permiten que el texto pueda rodear a la imagen.

De momento nos vamos a limitar a escoger la posición del titular con respecto a la imagen (si es que queremos ponerle un titular, claro está). Se puede poner arriba, en medio o abajo del lado de la imagen. Para ello se añade el comando `ALIGN` (desaconsejado) a la etiqueta, de la siguiente manera:

Titular alineado arriba

**<IMG SRC="isla.jpg" ALIGN=TOP> Titular alineado arriba**

**Ejemplo:**

**<HTML>**

**<HEAD>**

**<TITLE> practica 3: Imagenes texto </TITLE>**

**</HEAD>**

<BODY>

<h2 align=left>Imagen alineada verticalmente con aling="top"</h2>

<IMG SRC="G:\multimedia\imagenes\dock.jpg" ALIGN=TOP WIDTH=150  
HEIGHT=150 >esto es un texto metodo en un párrafo. esto es un texto  
metodo en un párrafo. esto es un texto metodo en un párrafo. esto es un  
texto metodo en un párrafo.

</BODY>

</HTML>.

Titular alineado en medio

<IMG SRC="isla.jpg" ALIGN=MIDDLE> Titular alineado en medio

Ejemplo:

<HTML>

<HEAD>

<TITLE> practica 3: Imagenes texto </TITLE>

</HEAD>

<BODY>

<h2 align=left>Imagen alineada verticalmente con aling="middle"</h2>

<IMG SRC="G:\multimedia\imagenes\dock.jpg" ALIGN=MIDDLE WIDTH=150  
HEIGHT=150 >esto es un texto metodo en un párrafo. esto es un texto  
metodo en un párrafo. esto es un texto metodo en un párrafo. esto es un  
texto metodo en un párrafo.

</BODY>

</HTML>.

Titular alineado abajo

<IMG SRC=" G:\multimedia\imagenes\isla.jpg" ALIGN=BOTTOM> Titular  
alineado abajo

Ejemplo:

<HTML>

<HEAD>

<TITLE> practica 3: Imagenes texto </TITLE>

</HEAD>

<BODY>

<h2 align=left>Imagen alineada verticalmente con aling="bottom"</h2>

<IMG SRC="G:\multimedia\imagenes\dock.jpg" ALIGN=BOTTOM  
WIDTH=150 HEIGHT=150 >esto es un texto metodo en un párrafo. esto es  
un texto metodo en un párrafo. esto es un texto metodo en un párrafo.  
esto es un texto metodo en un párrafo.

</BODY>

</HTML>.

Otra posibilidad muy interesante es la de utilizar una imagen como enlace a otra página. Para estos casos se utilizan generalmente imágenes pequeñas (iconos), aunque se puede usar cualquier tipo de imagen.

Según vimos en el capítulo anterior, la estructura general de un enlace es:

```
<A HREF="xxx"> yyy </A>
```

donde xxx era el destino del enlace e yyy el texto del enlace (o más generalmente hablando, lo que aparece en la pantalla como el enlace; en el capítulo anterior era un texto, y en éste va a ser una imagen). En este caso sustituimos xxx por el nombre del fichero de la página a la que queremos acceder. Y en lugar de yyy ponemos la etiqueta completa de la imagen (que queda así englobada dentro de la etiqueta del enlace)

Como ejemplo vamos a utilizar la imagen (hombre.jpg) para acceder al ejemplo práctico del capítulo 2 (mipag2.html):

```
<A HREF="mipag2.html"><IMG SRC="hombre.jpg"></A>
```

Pulsando la imagen comprobamos cómo efectivamente enlaza con la página deseada. Obsérvese además que la imagen está rodeada de un rectángulo del color normal en los enlaces. Si no se desea que aparezca ese rectángulo, hay que incluir dentro de la etiqueta de la imagen el atributo **BORDER=0**, es decir:

```
<A HREF="mipag2.html"><IMG SRC="hombre.jpg" BORDER=0></A>
```

Posicionando el cursor sobre esta última imagen, comprobamos que actúa también como enlace aunque carezca del rectángulo de color. Esto puede resultar más estético, pero se corre el riesgo de que el usuario no se dé cuenta de que la imagen sirve de enlace.

También podemos utilizar una imagen para enlazar con otra imagen. En este caso sustituimos xxx (el destino del enlace) con el nombre del fichero de la imagen a la que queremos acceder e yyy (lo que aparece en pantalla como el enlace) por la



etiqueta completa de la imagen que queremos que aparezca en la pantalla como el enlace de la otra.

Supongamos que queremos enlazar con la imagen isla.jpg por medio de esta otra imagen: (casa.jpg):

```
<A HREF="isla.jpg"><IMG SRC="casa.jpg"></A>
```

Que da como resultado:

Por último, otra posibilidad es la de utilizar un texto para enlazar con una imagen. En este caso sustituimos xxx (el destino del enlace) con el nombre del fichero de la imagen a la que queremos acceder e yyy (lo que aparece en pantalla como el enlace) por el texto.

Supongamos que queremos enlazar con la imagen isla.jpg por medio del texto "un paraíso tropical":

```
<A HREF="isla.jpg"> un paraíso tropical </A>
```

Se pueden capturar las imágenes que aparecen en pantalla, con objeto de guardarlas permanentemente en nuestro disco duro. En los navegadores más actuales se hace pulsando sobre la imagen con la tecla derecha del ratón, con lo que se obtiene un menú en el que está la posibilidad de guardar la imagen.

Un tipo de imágenes del que se hace abundante uso y que sirven para mejorar la presentación de la página son los iconos, botones, barras separadoras, etc. A pesar de su tamaño o forma, son imágenes como cualquier otra.

### Ejemplo práctico

Primeramente es necesario capturar tres imágenes:

isla.jpg, hombre.jpg y casa.jpg y guardarlas en el mismo directorio en el que se

## UT2. HTML BÁSICO

---

guardará el fichero de texto que se va a crear a continuación, junto con los de los capítulos anteriores.

En el procesador de textos copiamos:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Mi Web - 4 </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<CENTER>
```

```
<H1> Mi página Web </H1>
```

```
</CENTER>
```

```
<HR>
```

Esta es mi página Web. No es muy extensa, pero tiene todos los elementos básicos.

Espero que os guste. Poco a poco le iré añadiendo más cosas interesantes.

```
<P> <A HREF="mipag2.html"> <IMG SRC="hombre.jpg"> </A> Mis aficiones
```

```
<P> <A HREF="mipag3.html"> <IMG SRC="casa.jpg"> </A> Mis páginas favoritas
```

```
<CENTER>
```

```
<H3> Un lugar ideal para mis vacaciones </H3>
```

```
<IMG SRC="isla.jpg" ALT="isla">
```

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

Guardamos el fichero de texto con el nombre mipag4.html y lo cargamos en el navegador.

### Atributos en las imagenes

- **src**: indica la URI de acceso a la imagen.
- **alt**: texto con breve descripción de la imagen, que se presenta en el caso de no poder visualizar la imagen.
- **longdesc**: indica URI de una descripción larga de la imagen, en caso de tener asociado un mapa de imágenes, debe describir su contenido.
- **name**: identifica la imagen para su posterior manejo en hoja de estilo o script.
- **height, width**: establecen la altura y el ancho.
- **usemap**: sirve para asociar a la imagen un mapa de imagenes.
- **ismap**: sirve para definir un mapa de imagenes.

## 6. Caracteres especiales

Existen algunas limitaciones para escribir el texto. Una de ellas es debido a que las etiquetas se forman como un comando escrito entre los símbolos "<" y ">". Por tanto, si se quisieran escribir estos caracteres como parte normal del texto, daría esto lugar a una ambigüedad, ya que el programa navegador podría interpretarlos como el comienzo o final de una etiqueta, en vez de un carácter más del texto.

Para resolver este problema, existen unos códigos para poder escribir estos caracteres y otros relacionados con las etiquetas.

**&lt;** para < (less than, menor que)

**&gt;** para > (greater than, mayor que)

`&amp;` para `&` (ampersand)

`&quot;` para `"` (double quotation)

Como se ve, estos códigos empiezan siempre con el signo `&` y acaban siempre con `;`

De una manera similar, existen códigos para escribir letras específicas de distintos idiomas.

Hay muchos de ellos, pero, lógicamente, los que más nos interesan son los propios del castellano (las vocales acentuadas, la ñ y los signos `é` y `í` )

Los códigos de las vocales acentuadas se forman comenzando con `&`, seguido de la vocal en cuestión, seguido de la palabra acute (aguda) y terminando con el signo `;`

`&aacute;` para la `á`

`&eacute;` para la `é`

`&iacute;` para la `í`

`&oacute;` para la `ó`

`&uacute;` para la `ú`

`&Aacute;` para la `Á`

`&Eacute;` para la `É`

`&Iacute;` para la `Í`

`&Oacute;` para la `Ó`

`&Uacute;` para la `Ú`

El resto de los códigos son:

`&ntilde;` para la ñ

`&Ntilde;` para la Ñ

`&uuml;` para la ü

`&Uuml;` para la Ü

`&#191;` para ¿

`&#161;` para ¡

Todo esto, que como se ve es muy laborioso, puede parecer inútil ya que si escribimos nuestro texto sin hacer ningún caso de estas convenciones, escribiendo las letras acentuadas y demás signos directamente, es muy posible que el resultado lo veamos correctamente en nuestro navegador, pero nunca podremos estar seguros que les ocurra lo mismo a todos los que accedan a nuestras páginas con otros navegadores distintos.

En la práctica, el problema no es tan grave, pues los programas editores de HTML suelen tener la posibilidad de escribir los códigos automáticamente.

Incluso si se escribe en un procesador de textos de forma manual, se puede escribir el texto en una primera etapa de forma normal y luego aplicarle algún de los programas que hacen la conversión adecuada.

### Ejemplo práctico

Como ejercicio de este capítulo vamos a sustituir en el ejemplo práctico del capítulo anterior (mipag4.html) las vocales acentuadas por sus correspondientes códigos.

Quedará de esta manera:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Mi página del Web - 4 </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<CENTER>
```

```
<H1> Mi página del Web </H1>
```

```
</CENTER>
```

```
<HR>
```

Esta es mi página del Web. No es muy extensa, pero tiene todos los elementos básicos. Espero que os guste. Poco a poco le iré añadiendo más cosas interesantes.

```
<P> <A HREF="mipag2.html"> <IMG SRC="hombre.jpg"> </A> Mis aficiones
```

```
<P> <A HREF="mipag3.html"> <IMG SRC="casa.jpg"> </A> Mis  
páginas
```

favoritas

```
<CENTER>
```

```
<H3> Un lugar ideal para mis vacaciones </H3>
```

```
<IMG SRC="isla.jpg" ALT="isla">
```

```
</CENTER>
```

</BODY>

</HTML>

Guardamos el fichero de texto con el nombre mipag5.html y lo cargamos en el navegador.

## 7. Bibliografía

Manual HTML de Francisco Arocena

## 8. Curiosidades

<http://rendur.com/>

Rendur es un editor de HTML que funciona en tiempo real, con lo cual los cambios que vamos imprimiéndole al código se van viendo en tiempo real sin necesidad de llevar a cabo ninguna acción. Pero no sólo es un editor de código HTML, también podemos editar CSS cambiando a la pestaña adecuada.

Apenas entramos al sitio vemos a la izquierda el documento real y a la derecha un popup flotante donde vemos el código editable, ésta ventana la podemos redimensionar y desplazar a gusto para que no nos moleste. A medida que vamos editando el código se puede ver a la izquierda como va quedando.

## UT2. HTML BÁSICO

---

Ejercicio: crear una página con tres párrafos teniendo cada párrafo una cabecera de distinto tamaño y justificación. Dejando distintos espacios entre los párrafos y las cabeceras y los párrafos. Con frases en negrita y cursiva subrayadas y tachadas. Poniendo una línea entre párrafo y párrafo de distinto color anchura y altura así como justificación y con texto preformateado

**Ejercicio:** Ejercicio: crear una página con tres párrafos teniendo cada párrafo una cabecera de distinto tamaño y justificación. Dejando distintos espacios entre los párrafos y las cabeceras y los párrafos. Con frases en negrita y cursiva subrayadas y tachadas y con texto preformateado y sangría.