

**HTML**

**CSS**



## 1. Objetivos

En este primer tema se pretenden conseguir los siguientes objetivos:

- ✓ Conocer qué es y para qué sirve HTML5 y CSS3.
- ✓ Ver las diferencias entre HTML5 y sus antecesores HTML4 y XHTML.
- ✓ Repasar las características de HTML4 que son similares o iguales a la nueva versión HTML5.
- ✓ Repasar las características de CSS2.1 que son similares o iguales a la nueva versión CSS3.
- ✓ Aplicar estilos CSS a un documento HTML para lograr el diseño deseado.

## 2. Introducción

Este tema ofrece una visión general de cómo hemos llegado a donde estamos hoy en día, por qué HTML5 y CSS3 son tan importantes para las aplicaciones web modernas y cómo el uso de estas tecnologías nos va a facilitar ampliamente nuestra labor como desarrolladores web.

### ¿Qué es HTML5?

Lo que hoy conocemos por HTML5 ha tenido una historia relativamente turbulenta. Como ya sabemos, HTML es el lenguaje de marcado predominante para describir el contenido o los datos en la World Wide Web. HTML5 es la última versión de este lenguaje, e incluye nuevas características, mejoras en las características existentes, y scripting basado en APIs.

En primer lugar, HTML5 incluye nuevos elementos de marcado cuya semántica está asociada con el significado de los contenidos que vamos a introducir en ellos.

El término "HTML5", además, ha sido utilizado para referirse a una serie de nuevas tecnologías y APIs, como son: el dibujo con el elemento `<canvas>`, almacenamiento fuera de línea, el nuevo `<video>` y `<audio>`, funcionalidades de arrastrar y soltar, microdatos, etc.

Por otra parte, la introducción de audio y video basado en HTML5, significa que habrá una menor dependencia de software de terceros y plugins al publicar contenido multimedia en la Web. En general, hay buenas razones para empezar a utilizar las nuevas características de HTML5 y las APIs asociadas

### ¿Qué es CSS3?

Otra parte separada (pero no menos importante) de la creación de páginas web, son las hojas de estilo en cascada. CSS es un lenguaje de estilo que describe cómo se realiza la presentación de los contenidos HTML, y CSS3 es la última versión de este lenguaje.

CSS3 contiene casi todo lo que se incluye en CSS 2.1 (la versión anterior de la especificación) y, además, agrega nuevas características para ayudar a los desarrolladores a resolver una serie de problemas sin necesidad de marcado no semántico, complejos scripts o imágenes adicionales.

Las nuevas características de CSS3 incluyen soporte para selectores adicionales, sombras, esquinas redondeadas, múltiples fondos, animaciones, transparencias y mucho más.

### 3. Una Plantilla HTML5 básica

Veamos, a continuación, el esqueleto de un documento HTML5:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>HTML</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="estilo.css">
</head>

<body>
  <p>Esta página web es una página HTML válida.</p>
</body>
</html>
```

En este código se pueden reconocer algunos bloques anidados en el que cada bloque corresponde a una etiqueta.



## El doctype

En primer lugar, tenemos el tipo de declaración de documento o doctype. Esto es, simplemente, una manera de decirle al navegador, o cualquier otro software que analice nuestro código, de qué tipo de documento se trata. En el caso de los archivos HTML, indica la versión específica.

El tipo de documento debe ser siempre el primer punto en la parte superior de todos los archivos HTML. Anteriormente, la declaración de tipo de documento era farragosa y difícil de recordar.

Para XHTML 1.0 Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Y para HTML4 Transitional:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5 ha acabado con esa monstruosidad indescifrable. Ahora todo lo que indicamos es lo siguiente:

```
<!DOCTYPE HTML>
```

Lo primero que podemos observar es que el 5 está ausente de la declaración. Aunque la iteración actual del lenguaje de marcado se conoce como "HTML5", en realidad es sólo una evolución de los estándares anteriores, y las futuras especificaciones serán, simplemente, un desarrollo de lo que tenemos hoy. Dado que los navegadores tienen que soportar todo el contenido existente en la web, no se confía en el tipo de documento para indicar qué características deben soportarse en un documento dado.

### **P2.1. Prueba de HTML**

Crea un documento de texto plano y copia y pega este texto. Guarda el documento como "prueba.html". Abre ahora el documento con un navegador.

```
<!DOCTYPE html>
<head>
  <title>Mi primera página HTML</title>
</head>

<body>
  <p>Esta página web es una página HTML válida.</p>
</body>
</html>
```

La validación es el proceso que asegura que un documento escrito en un determinado lenguaje, por ejemplo HTML5, cumple con las normas y restricciones de ese lenguaje. El proceso de validación consiste en probar página a página si el código HTML5 pasa la prueba de validación. Los validadores son las herramientas que se utilizan para validar cada página. El organismo W3C ha creado una herramienta que se puede utilizar gratuitamente a través de Internet (<http://validator.w3.org/>).

## **P2.2. Validación web**

Entra en la web de validator de w3 y valida el código fuente de las siguientes páginas web:

- [www.google.es](http://www.google.es)
- [www.amazon.es](http://www.amazon.es)
- [www.netflix.com](http://www.netflix.com)

¿Aparecen errores? ¿Por qué crees que pasa esto?

## **El elemento html**

El siguiente paso en cualquier documento HTML es el elemento `<html>`, que no ha cambiado significativamente con HTML5. En nuestro ejemplo, hemos incluido el atributo `lang` con un valor de `es`, que especifica que el documento está en español. En la sintaxis basada en XHTML, estaríamos obligados a incluir un atributo `xmlns`. En HTML5, esto ya no es necesario, e incluso podríamos prescindir del atributo `lang` y el documento validaría y funcionaría correctamente.

## El elemento head

### CABECERA DEL DOCUMENTO

```
<head>
  <title></title>
</head>
```

La directiva **<head></head>** delimita la cabecera del documento. Dentro de la cabecera es importante definir el título de la página por medio de la directiva **<title></title>**. Este título será el que aparezca en la barra de nuestro visor de páginas Web.

```
<title>Una página Web de Prueba</title>
```

Dentro de la cabecera de nuestro documento podemos incluir otras directivas adicionales. La directiva **<meta>** indica al visor de Internet las palabras clave y contenido de nuestra página Web. Muchos de los buscadores de páginas Web de Internet utilizan el contenido de esta directiva para incluir la página en sus bases de datos. La directiva **<meta>** lleva generalmente dos parámetros, **name** y **content**.

#### Ejemplo:

```
<meta name = "Página de Prueba" content = " Fútbol y Baloncesto">
```

Indica al visor el nombre de la página y sus contenidos principales.

#### Ejemplo:

```
<meta name = "keywords" content = "Fútbol Baloncesto España">
```

Indica al visor las palabras clave para los buscadores de Internet.

Otro uso de la directiva **<META>** es la de indicar documentos con "refresco automático". Si se indica una **URL** se sustituirá el documento por el indicado una vez transcurridos el número de segundos especificados. Si no se incluye ninguna **URL** se volverá a cargar en el visor el documento en uso transcurridos los segundos indicados.

Esto es útil para páginas que cambian de contenido con mucha frecuencia o para redireccionar a la persona que visita nuestra página Web a una nueva dirección donde se encuentra una versión actualizada de nuestra página Web.

Ejemplo:

```
<meta http-equiv="refresh" content="15;url=http://www.microsoft.com">
```

Transcurridos 15 segundos se accederá a la página Web de Microsoft. La directiva **<base>** indica la localización de los ficheros, gráficos, sonidos, etc... a los que se hace referencia en nuestra página Web. Si no se incluye esta directiva el visor entiende que dichos elementos se encuentran en el mismo lugar donde se encuentra nuestra página Web.

Ejemplo:

```
<base href ="http://www.imagenes.com/celebrities/">
```

En HTML5, podemos usar minúsculas, mayúsculas, o mayúsculas y minúsculas en los nombres o atributos de etiqueta, así como introducir valores de atributos sin comillas (siempre y cuando estos valores no contengan espacios u otros caracteres reservados), y todo será correcto para el validador.

En cualquier caso, es **muy recomendable** seguir unas directrices a la hora de codificar nuestras páginas, por ejemplo, sería una buena práctica mantener las restricciones en cuanto a mayúsculas y minúsculas y/o seguir poniendo los valores de los atributos entre comillas.

### **P2.3. Probando el navegador**

Crea un nuevo documento de texto plano llamado "prueba.html". En la página tiene que aparecer el título en la barra del navegador con tu nombre y en la página web se mostrará el texto "Bienvenid@ a Lenguaje de Marcas".

## **Nombres de documentos html**

Utiliza nombres en tus archivos que tengan algunas normas básicas para ahorrarte disgustos y líos. El consejo es que no utilices acentos ni espacios ni otros caracteres raros. También te ayudará escribir siempre las letras en minúsculas. Esto no quiere decir que debes hacer nombres de archivos cortos, es mejor hacerlos descriptivos para que te aclaren lo que hay dentro. Algún carácter como el guión "-" o el guión bajo "\_" te puede ayudar a separar las palabras. Por ejemplo quienes\_somos.html



## Cuerpo del documento <body>

La **directiva <body></body>** indica el inicio y final de nuestra página Web. Será entre el inicio y el final de esta directiva donde pongamos los contenidos de nuestra página, textos, gráficos, enlaces, etc.... Esta directiva tiene una serie de parámetros opcionales que nos permiten indicar la "apariencia" global del documento:

**background**="nombre de fichero grafico"

Indica el nombre de un fichero gráfico que servirá como "fondo" de nuestra página. Si la imagen no rellena todo el fondo del documento, esta será reproducida tantas veces como sea necesario (en modo mosaico).

**bgcolor** ="código de color"

Indica un color para el fondo de nuestro documento. Se ignora si se ha usado el parámetro **background**.

**text** ="código de color"

Indica un color para el texto que incluyamos en nuestro documento. Por defecto es negro.

**link** ="código de color"

Indica el color de los textos que dan acceso a un Hiperenlace. Por defecto es azul.

**vlink** ="código de color"

Indica el color de los textos que dan acceso a un Hiperenlace que ya hemos visitado con nuestro visor. Por defecto es púrpura.

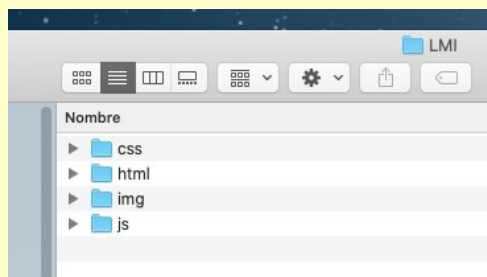
El **código de color** es un número compuesto por tres pares de cifras hexadecimales que indican la proporción de los colores "primarios", **rojo**, **verde** y **azul**. El código de color se antecede del símbolo #.

Estos atributos opcionales de <body> están **obsoletos** desde la versión 4.01 y no son soportados en HTML5, dado que la idea es separar el contenido de la forma en que se presenta utilizando CSS.

Antes de hablar de los nuevos elementos que incorpora HTML5, haremos un repaso de los principales elementos básicos ya existentes en HTML4 y XHTML, y que no varían en HTML5.

### **P2.4. Estructura de carpetas**

Antes de hacer ejercicios más complejos es importante crear siempre una estructura de carpetas. En esta ocasión vamos a crearla de esta forma:









Crea en tu ordenador una carpeta con tu nombre y dentro de ésta la carpeta LMI\_tunombre con la estructura indicada. Así cada documento irá en la carpeta que corresponda y tendremos organizadas nuestras prácticas de lenguaje de marcas.

## **Colores:**

Cada color es definido por un número hexadecimal que está compuesto a su vez de tres partes. Cada una de estas partes representa la contribución del rojo, verde y azul al color en cuestión. RGB Por otra parte, es posible definir de una manera inmediata algunos de los colores más frecuentemente usados para los que se ha creado un nombre más nemotécnico:

#### **Nombres de colores y valores #RGB**

	Black = "#000000" (Negro)		Green = "#008000" (Verde)
	Silver = "#C0C0C0" (Plateado)		Lime = "#00FF00" (Verde lima)
	Gray = "#808080" (Gris)		Olive = "#808000" (Verde oliva)
	White = "#FFFFFF" (Blanco)		Yellow = "#FFFF00" (Amarillo)
	Maroon = "#800000" (Marrón)		Navy = "#000080" (Azul marino)
	Red = "#FF0000" (Rojo)		Blue = "#0000FF" (Azul)
	Purple = "#800080" (Púrpura)		Teal = "#008080" (Azul verdoso)
	Fuchsia = "#FF00FF" (Fucsia)		Aqua = "#00FFFF" (Celeste)

Los colores en HTML se representan mediante un número hexadecimal. Un número hexadecimal se diferencia de un número decimal en que no sólo puede tomar valores del 0 al 9, sino que puede tomar hasta dieciséis valores distintos, que van del 0 al 9, y de la A a la F. Cada color estará representado por un grupo de seis dígitos en hexadecimal, precedidos por una almohadilla, como por ejemplo #FFFFFF.

## 4. Etiquetas básicas

### Cabeceras

Las cabeceras en HTML5 están definidas con los elementos `<h1>` hasta `<h6>`.

```
<h1>Cabecera de nivel 1</h1>
<h2>Cabecera de nivel 2</h2>
<h3>Cabecera de nivel 3</h3>
```

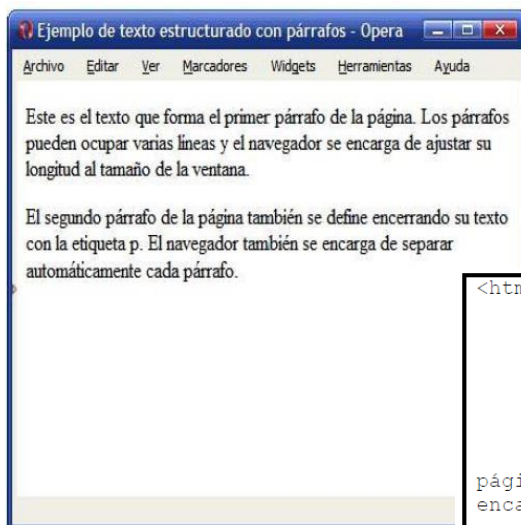
Las cabeceras van a definir el esquema del documento, por lo tanto, debemos utilizarlas de forma coherente. No tiene sentido tener un `<h2>` si no tenemos antes un `<h1>`, por ejemplo.

### Párrafos

Los párrafos de texto los introduciremos mediante la etiqueta `<p>`.

```
<p>Esto es un párrafo</p>
<p>Esto es otro párrafo</p>
```

No debemos utilizar los párrafos para introducir saltos de línea, para eso ya disponemos del elemento `<br>`.



```
<html>

  <head>
    <title>Ejemplo de texto estructurado con párrafos</title>
  </head>

  <body>
    <p>Este es el texto que forma el primer párrafo de la
    página. Los párrafos pueden ocupar varias líneas y el navegador se
    encargade ajustar su longitud al tamaño de la ventana.
    </p>
    <p>El segundo párrafo de la página también se define
    encerrando su texto con la etiqueta p. El navegador también se encarga
    de separar automáticamente cada párrafo.
    </p>
  </body>

</html>
```

Los párrafos delimitados por etiquetas `<p>` pueden ser fácilmente justificados a la izquierda, centro o derecha especificando dicha justificación en el interior de la etiqueta por medio de un atributo `align`.

Así, si deseásemos introducir un **texto alineado a la izquierda** escribiríamos:

```
<p align="left">Texto alineado a la izquierda</p>
```

El resultado sería:

Texto alineado a la izquierda

Para una **justificación al centro**:

```
<p align="center">Texto alineado al centro</p>
```

que daría:

Texto alineado al centro

Para **justificar a la derecha**:

```
<p align="right">Texto alineado a la derecha</p>
```

cuyo efecto sería:

Texto alineado a la derecha

Una forma de simplificar nuestro código y de evitar introducir continuamente el atributo `align` sobre cada una de nuestras etiquetas es utilizando la etiqueta `<div>`.

Esta etiqueta por sí sola no sirve para nada. Tiene que estar acompañada del atributo `align` y lo que nos permite es alinear cualquier elemento (párrafo o imagen) de la manera que nosotros deseemos.

Así, el código:

```
<p align="left"> Parrafo1</p>
```

```
<p align="left"> Parrafo2</p>
```

```
<p align="left"> Parrafo3</p>
```

es equivalente a :

```
<div align="left">
<p>Parrafo1</p>
<p>Parrafo2</p>
<p>Parrafo3</p>
</div>
```

## Espaciados y Saltos de línea

Con la etiqueta `<p>` se introduce un salto y deja una línea en blanco antes de continuar con el resto del documento como hemos visto en el punto anterior. Podemos también usar la etiqueta `<br>`, de la cual **no** existe su cierre correspondiente (`</br>`), para realizar un simple retorno de carro con lo que no dejamos una línea en blanco sino que solo cambiamos de línea.

Puedes comprobar que cambiar de línea en nuestro documento HTML sin introducir alguna de estas u otras etiquetas **no** implica en absoluto un cambio de línea en la página visualizada. En realidad el navegador introducirá el texto y no cambiara de línea a no ser que esta llegue a su fin o bien lo especifiquemos con la etiqueta correspondiente.

### P2.5. Atom



Entra en el programa Atom (si no está instalado en tu pc instálalo)<sup>1</sup>

Abre tu carpeta de LMI\_tunombre creada en el ejercicio anterior como nuevo proyecto. Crea un nuevo documento dentro de la carpeta HTML que contenga el siguiente código (que ya hemos visto antes que es el código básico o plantilla inicial de cualquier doc html):

```
1  <!doctype html>
2  <html>
3  <head>
4  <title> Probando atom</title>
5  <meta autor="maribel">
6  </head>
7  <body>
8
9
10
11 </body>
12 </html>
13
```

- Cambia el título del navegador para que ponga "Probando atom".
- Cambia el nombre del autor y pon el tuyo.
- Agrega dos cabeceras (una con tu nombre y otra con tus apellidos).
- Tres párrafos (uno alineado a la izquierda, otro al centro y otro a la derecha).
- Guarda el documento como P2.5.
- Observa que en el navegador se muestra todo lo especificado.
- Sube el archivo a la plataforma.

1 Instalación de ATOM en Ubuntu:

<https://ubunlog.com/editor-atom-instalacion-ubuntu/#Opcion 1 Instalar desde el repositorio>

## Texto preformateado

<PRE> HTML elimina cualquier espacio en blanco adicional que se inserte pero nos encontramos con una excepción a esta regla cuando utilizamos las tags <pre>.....</pre>. Sin embargo esta tag convertirá el texto afectado a fuente monoespaciada (posiblemente Courier).

```

1 <html xmlns="http://www.w3.org/1999/html" lang="es" xml:lang="es">
2
3
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 <title>Espacios en blanco arbitrarios</title>
7 </head>
8
9 <body>
10
11 <h2>Características de los planetas</h2>
12
13 <pre>
14
15 Nombre      Diametro relativo  Período orbital  Número de lunas
16 -----
17 Mercurio    0,382                0,24 años       0
18 Venus      0,949                0,62 años       0
19 Tierra      1                    1 año           1
20 Marte       0,532                1,88 años       2
21 Júpiter     11,209               11,86 años      49
22 Saturno     9,449                29,46 años      52
23 Urano       4,007                84,01 años      27
24 Neptuno     3,883                164,80 años     13
25
26 </pre>
27
28 </body>
29
30 </html>
31
32

```

### Características de los planetas

Nombre	Diametro relativo	Período orbital	Número de lunas
Mercurio	0,382	0,24 años	0
Venus	0,949	0,62 años	0
Tierra	1	1 año	1
Marte	0,532	1,88 años	2
Júpiter	11,209	11,86 años	49
Saturno	9,449	29,46 años	52
Urano	4,007	84,01 años	27
Neptuno	3,883	164,80 años	13

#### **P2.6. Texto preformateado**

Crea un nuevo documento llamado P2.6 insertando el código de la etiqueta pre del ejemplo anterior de las características de los planetas (puedes duplicar el documento P2.5 y modificarlo para ir más rápido). Abre tu documento con el navegador y comprueba si se muestra tal y como lo has escrito.



## Contenedores o Layers (Capas)

**Etiqueta <div>** Si tenemos un texto largo y queremos que todos los párrafos estén justificados de alguna manera utilizaremos la etiqueta <div>. Todo lo que esté incluido dentro será alineado tal y como hayamos especificado en el atributo **align**:

```
<!doctype html>
<html>
<head>
<title> Probando atom</title>
<meta autor="maribel">
</head>
<body>

<p> Para simplificar el código a la hora de justificar utilizamos la etiqueta <div>.</p>
<div align="right">
  <p>centrado a la derecha </p>
  <p> insisto en que está centrado a la derecha </p>
</div>

<div align="center">
  <p>todo lo incluido dentro estará centrado</p>
  <p> insisto en que está centrado</p>
</div>

<div align="justify">
  <p> Con el valor "justify" hacemos un justificado a ambos márgenes.</p>
</div>

</body>
</html>
```

### **P2.7. Div**

Copia el código anterior en un nuevo documento creado en atom que se tiene que llamar P2.7 (puedes duplicar P2.6 y modificarlo). Observa qué se muestra en el navegador.  
¡¡MUY IMPORTANTE!! Puede parecer una tontería pero es importante tabular para poder leer el código a simple vista.

## Etiquetas para formatear textos

HTML utiliza una gran variedad de etiquetas para formatear textos, algunas de las más importantes son las siguientes:

→ **Espacios en blanco:**

Sustituir cada nuevo espacio en blanco por el texto `&nbsp;`;

[illegible]

### P2.8. Espacios en blanco

Copia el código del ejemplo anterior en un nuevo documento de atom para comprobar cómo poner espacios en blanco en HTML.

→ Etiqueta `<span>`:

- Es un contenedor en línea.
- Sirve para aplicar estilo al texto o agrupar elementos en línea
- Crea una caja.
- Es habitual usarla junto con los **atributos id y class** para modificar posteriormente su aspecto con **CSS**
- En HTML5 pasa a ser obsoleta puesto que se recomienda crear los estilos independientes de la página HTML.

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo etiqueta span - aprenderaprogramar.com</title>
  </head>
  <body>
    <div style="text-align: center; font-size: 60px; border: 1px solid red;">
      Cabecera
      <span style="color: green;">de</span> la <span style="color: orange;">página</span>.
    </div>
    <div style="border: 1px solid green;">
      Cuerpo de la página.
    </div>
    <div style="border: 1px solid blue;">
      Pie de la página
    </div>
  </body>
</html>
```

### **P2.9. Span**

Copia el código del ejemplo anterior en un nuevo documento de atom. ¿Cómo se muestra la página en el navegador?

#### **➔ Blockquote**

Crea citas en bloque, marca a otros autores o documentos:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Mi primera página web</title>
  </head>
  <body>
    <p>iHola a todos! bienvenidos a mi primer sitio web.</p>

    <blockquote>
      <p>
        Ser el hombre más rico del cementerio no es lo que más
        me importa... Ir a la cama a la<br > noche diciendo
        "hemos hecho algo maravilloso" es lo que realmente me
        preocupa.
      </p>
    </blockquote>

    <p>En un lugar de la Mancha, de cuyo nombre no quiero
    acordarme</p>
  </body>
</html>
```

### **P2.10. Blockquote**

Copia el código del ejemplo anterior en un nuevo documento de atom. ¿Cómo se muestra la cita de blockquote?

#### **→ Negrita**

Podemos escribir texto en negrita incluyéndolo dentro de las etiquetas `<b>` y `</b>` (bold). Esta misma tarea es desempeñada por `<strong>` y `</strong>` siendo ambas equivalentes. Escribiendo un código de este tipo:

```
<b>Texto en negrita</b>
```

Obtenemos este resultado:

#### **Texto en negrita**

La diferencia entre `<b>` y `<strong>` se podrá entender ahora. Mientras que `<b>` significa simplemente negrita y todos los navegadores la interpretarán como negrita, `<strong>` es una etiqueta que significa que se tiene que resaltar fuertemente el texto y cada navegador es el responsable de resaltarlo como desee. En la práctica `<strong>` coloca el texto en negrilla, pero podría ser que un navegador decidiese resaltar colocando negrilla, subrayado y color rojo en el texto.

#### **→ Itálica**

También en este caso existen dos posibilidades, una corta: `<i>` e `</i>` (italic) y otra un poco más larga: `<em>` y `</em>`. En este manual, y en la mayoría de las páginas que veréis por ahí, os encontraréis con la primera forma sin duda más sencilla a escribir y a acordarse.

He aquí un ejemplo de texto en itálica:

```
<i>Texto en itálica</i>
```

Que da el siguiente efecto:

*Texto en itálica*

### → Subrayado

El HTML nos propone también para el subrayado el par de etiquetas: `<u>` y `</u>` (underlined). Sin embargo, el uso de subrayados ha de ser aplicado con mucha precaución dado que los enlaces hipertexto van, a no ser que se indique lo contrario, subrayados con lo que podemos confundir al lector y apartarlo del verdadero interés de nuestro texto.

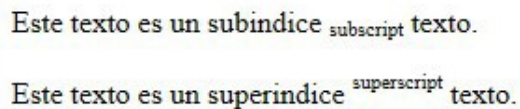
### → Subíndices y supraíndices

Este tipo de formato resulta de extremada utilidad para textos científicos. Las etiquetas empleadas son:

```
<!DOCTYPE html>
<html>
<body>

    <p>Este texto es un subíndice <sub>subscript</sub> texto.</p>
    <p>Este texto es un supraíndice <sup>superscript </sup>
texto.</p>

</body>
</html>
```



Este texto es un subíndice subscript texto.

Este texto es un superíndice superscript texto.

### → Anidar etiquetas

Todas estas etiquetas (y las que veremos a continuación) se pueden anidar. Así, podemos sin ningún problema crear texto en negrita e itálica embebiendo una etiqueta dentro de la otra:

```
<b>Esto solo está en negrita <i> y esto en negrita e itálica</i></b>
```

Esto nos daría:

**Esto solo está en negrita y esto en negrita e *itálica***

Consejo: Cuando anides etiquetas HTML hazlo correctamente. Si abres etiquetas dentro de otra más principal, antes de cerrar la etiqueta principal cierras las etiquetas que hayas abierto dentro de ella. Debemos evitar códigos como el siguiente:

```
<b>Esto está en negrita e <i>itálica</b></i>
```

En favor de códigos con etiquetas correctamente anidadas:

```
<b>Esto está en negrita e <i>itálica</i></b>
```

Esto es muy aconsejable, aunque los navegadores entiendan bien las etiquetas mal anidadas, por dos razones:

1. Sistemas como XML no son tan permisivos con estos errores y puede que en el futuro nuestras páginas no funcionen correctamente.
2. A los navegadores les cuesta mucho tiempo de procesamiento resolver este tipo de errores, incluso más que construir la propia página y debemos evitarles que sufran por una mala codificación.

```
<etiqueta1>
  <etiqueta2>
    <etiqueta3>
      . . .
    </etiqueta3>
  </etiqueta2>
</etiqueta1>
```

## P2.11. Probando etiquetas

Intenta reproducir el siguiente fragmento de una página web utilizando las etiquetas que correspondan. Fíjate que se han utilizado encabezados de distintos tamaños, negrita, cursiva y que está todo formateado con párrafos.

### Biografía de Tim Berners Lee

**Berners-Lee** trabajó en el CERN desde junio hasta diciembre de 1980. Durante ese tiempo, propuso un proyecto basado en el *hipertexto* para facilitar la forma de compartir y la puesta al día de la información entre investigadores. En este periodo también construyó un programa llamado *ENQUIRE* que no llegó a ver la luz.

Después de dejar el CERN, en 1980, se fue a trabajar a la empresa de *John Poole Image Computer Systems Ltd.*, pero regresó al CERN otra vez en 1984.

#### El nacimiento de la web

En 1989, el CERN era el nodo de Internet más grande de Europa y Berners-Lee vio la oportunidad de unir Internet y el hipertexto (**HTTP** y **HTML**), de lo que surgiría la **World Wide Web**.

Desarrolló su primera propuesta de la Web el 12 de marzo de 1989, pero no tuvo mucho eco, por lo que en 1990 y con la ayuda de *Robert Cailliau*, hicieron una revisión que fue aceptada por su gerente, *Mike Sendall*.

Usó ideas similares a las que había usado en el sistema *Enquire*, para crear la **World Wide Web**, para esto diseñó y construyó el primer navegador (llamado WorldWideWeb y desarrollado con NEXTSTEP) y el primer servidor Web al que llamó httpd (HyperText Transfer Protocol daemon).

#### La democratización de la web

El **primer servidor Web** se encontraba en el CERN y fue puesto en línea el 6 de agosto de 1991. Esto proporcionó una explicación sobre lo que era el World Wide Web, cómo uno podría tener un navegador y cómo establecer un servidor Web. Este fue también el primer directorio Web del mundo, ya que Berners-Lee mantuvo una lista de otros sitios Web aparte del suyo.

Debido a que tanto el **software del servidor como del cliente fue liberado de forma gratuita** desde el CERN, el corazón de Internet Europeo en esa época, su difusión fue muy rápida. El número de servidores Web pasó de veintiséis en 1992 a doscientos en octubre de 1995 lo que refleja cual fue la velocidad de la difusión de internet.

#### El nacimiento del W3C

En 1994 entró en el *Laboratorio de Ciencias de la Computación e Inteligencia Artificial* del *Massachusetts Institute of Technology*. Se trasladó a EE. UU. y puso en marcha el **W3C**, que dirige actualmente.

El **W3C** es un organismo internacional de estandarización de tecnologías Web dirigido conjuntamente por el *Instituto Tecnológico de Massachusetts*, el *ERCIM francés* y la *Universidad de Keiō* en Japón. Este organismo decidió que todos sus estándares fuesen libres, es decir, que los pudiese utilizar todo el mundo libremente sin coste alguno, lo que sin lugar a dudas fue una de las grandes razones para que la Web haya llegado a tener la importancia que tiene hoy en día.

En su libro *Tejiendo la red*, publicado en 1999, Berners-Lee explica por qué **la tecnología web es libre y gratis. Se considera al mismo tiempo el inventor y el protector de la web.**

## ➔ Color, tamaño y tipo de letra

Esto se hace a partir de la etiqueta `<font>` y su cierre correspondiente. Dentro de esta etiqueta deberemos especificar los atributos correspondientes a cada uno de estos parámetros que deseamos definir. A continuación os comentamos los atributos principales de esta etiqueta:

### Atributo face

Define el tipo de letra. Hay que tener cuidado con este atributo ya que cada usuario, dependiendo de la plataforma que utilice, puede no disponer de los mismos tipos de letra que nosotros con lo que, si nosotros elegimos un tipo del que no dispone, el navegador se verá forzado a mostrar el texto con la fuente que utiliza por defecto (suele ser Times New Roman). Para evitar esto, dentro del atributo suelen seleccionarse varios tipos de letra separados por comas. En este caso el navegador comprobará que dispone del primer tipo

enumerado y si no es así, pasará al segundo y así sucesivamente hasta encontrar un tipo que posea o bien acabar la lista y poner la fuente por defecto.

Veamos un ejemplo:

```
<font face="Comic Sans MS,arial,verdana">Este texto tiene otra tipografía</font>
```

Que se visualizaría así en una página web (en comic sans):

Este texto tiene otra tipografía

### Atributo size

Define el tamaño de la letra. Este tamaño puede ser absoluto o relativo. Si hablamos en términos absolutos, existen 7 niveles de tamaño distintos numerados de 1 a 7 por orden creciente.

Elegiremos por tanto un valor size="1" para la letra más pequeña o size="7" para la más grande.

```
<font size=4>Este texto es más grande</font>
```

Podemos modificar el tamaño de nuestra letra con respecto al del texto mostrado anteriormente definiendo el número de niveles que queremos subir o bajar en esta escala de tamaños por medio de un signo + o -. De este modo, si definimos nuestro atributo como size="+1" lo que queremos decir es que aumentamos de un nivel el tamaño de la letra. Si estábamos escribiendo previamente en 3, pasaremos automáticamente a 4.

Los tamaños reales que veremos en pantalla dependerán de la definición y del tamaño de fuente elegido por el usuario en el navegador. Este tamaño de fuente puede ser definido en el navegador yendo al menú superior, Ver/Tamaño de la fuente. Esta flexibilidad puede en más de una ocasión resultarnos embarazosa ya que en muchos casos desearemos que el tamaño del texto permanezca constante para que éste quepa en un determinado espacio.

### **P2.12. Simular noticia**

Busca en un periódico digital una noticia de hoy. Crea una nueva página en atom donde vas a incluir la noticia que acabas de leer con las siguientes características:

- Titular con encabezado de nivel 1, en cursiva y de color rojo.
- Titular con encabezado de nivel 2 y de color verde oscuro.
- El color de fondo de la página amarillo.
- Texto de la página con la fuente "verdana" y si no lo encuentra fuente Arial.
- Utiliza negrita y cursiva cuando corresponda en la noticia.
- Utiliza párrafos para que quede mejor formateada la noticia.
- Alinea el texto centrado o justificado según corresponda.

Abre en tu navegador la página que acabas de crear para ver el resultado de tu noticia.



## Enlaces

Para crear enlaces, utilizaremos la etiqueta `<a>` con su atributo `href` para indicar la url de destino del enlace.

```
<a href="url">Texto del enlace</a>
```

### **P2.13. Enlace externo**

Crea en atom un nuevo documento con la estructura de siempre y que contenga lo siguiente:

- Titular con encabezado de nivel 1, en negrita y verde donde ponga "Enlace a página externa".
- El color de fondo de la página azul claro.
- Un enlace a la página de google con el texto "Ir a google".

Abre en tu navegador la página que acabas de crear para ver el resultado.

Si queremos crear un enlace que se dirija a un sitio determinado del documento actual (ANCLA), haremos lo siguiente:

- 1 Colocaremos un ancla en el lugar al que nos queremos dirigir:

```
<a id="contacto">Contacto</a>
```

- 2 En el enlace pondremos como `href` el símbolo `#` seguido del `id` que le hemos puesto al ancla:

```
<a href="#contacto">Ir a contacto</a>
```

### **P2.14. Ancla**

Crea en atom un nuevo documento con la estructura de siempre y que contenga lo siguiente:

Escribe cinco nombres de personas, cada uno se tiene que mostrar en una línea diferente. Pon 10 líneas entre cada uno de los nombres.

Crea un ancla a tu nombre.

Al principio del documento crea un ancla a tu nombre en el documento.

## Imágenes

Para vincular imágenes a nuestro documento, utilizaremos la etiqueta `<img>` con su atributo `src` para indicar el origen de la imagen. `<img>` es un elemento vacío, por lo tanto, no es necesario que introduzcamos la etiqueta de cierre. Debemos de poner el atributo `alt` a todas las imágenes para indicar un texto alternativo (por si no puede cargarse la imagen).

```

```

### **P2.15. Imágenes**

Crea un nuevo documento en atom que muestre lo siguiente:

- Un encabezado grande con este texto y distintos colores.
- Una imagen de Bart Simpson que te guste
- El texto de qué pasada de página alineado a la derecha de tu web.

Me encanta LENGUAJE DE MARCAS



Qué pasada de páginas estoy creando

### **P2.16. Enlaces con imágenes**

Duplica la práctica anterior y haz que la imagen de Bart Simpson sea un enlace a la página web de wikipedia donde habla de los Simpson.

## Tablas

Las tablas se definen con la etiqueta `<table>` y está dividida en filas con la etiqueta `<tr>`. Una fila se divide en celdas de datos con la etiqueta `<td>`, aunque también se puede dividir en celdas de cabecera con la etiqueta `<th>`. Los elementos `<td>` son los contenedores de datos en la tabla y pueden contener todo tipo de elementos HTML, como texto, imágenes, listas, otras tablas, etc.

Para crear una tabla como la siguiente:

Firstname	Lastname	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Introduciríamos el siguiente código html:

```
<table border="1">
  <tr><th>Firstname</th><th>Lastname</th><th>Points</th></tr>
  <tr><td>Jill</td><td>Smith</td><td>50</td></tr>
  <tr><td>Eve</td><td>Jackson</td><td>94</td></tr>
  <tr><td>John</td><td>Doe</td><td>80</td></tr>
  <tr><td>Adam</td><td>Johnson</td><td>67</td></tr>
</table>
```

Evidentemente, con esto sólo conseguimos la estructura de la tabla, si queremos que nuestra tabla tenga el mismo aspecto que aparece en la imagen, tenemos que aplicar los estilos CSS correspondientes (empezaremos a verlos en el siguiente tema).

### **P2.17. Interpretar Tabla básica**

Crea una página html donde se muestre la tabla anterior. Incluye además los colores de fondo y las alineaciones del texto que correspondan en cada celda.

### **P2.18. Crear Tabla básica**

Crea en una página web la siguiente tabla básica con los colores y alineados de texto correspondientes

DEPARTAMENTO	RESPONSABLE	MIEMBROS	NÓMINA
COMPRAS	JUAN MARÍN	8	1.500 €
VENTAS	MARÍA PEREZ	10	1.650 €
PUBLICIDAD	OSCAR RUÍZ	7	1.750 €
CALIDAD	CAROLINA TINAO	6	1.900 €
TECNOLOGÍAS	PAULA TERROBA	7	2.350 €

### **Atributos colspan y rowspan**

Los atributos colspan y rowspan permiten unir una celda con las celdas contiguas, tanto horizontal como verticalmente. El valor de colspan indica la cantidad de celdas unidas en horizontal y el valor de rowspan indica la cantidad de celdas unidas en vertical.

#### **Ejemplo 1:**

```
<table>
  <caption>Esto es una tabla</caption>
  <tr>
    <td colspan="2">celdas a1 y b1 unidas</td>
  </tr>
  <tr>
    <td>celda a2</td>
    <td>celda b2</td>
  </tr>
</table>
```

Esto es una tabla	
celdas a1 y b1 unidas	
celda a2	celda b2

### Ejemplo 2:

```
<table>
  <caption>Esto es una tabla</caption>
  <tbody>
    <tr>
      <td rowspan="2">celdas a1 y a2 unidas</td>
      <td>celda b1</td>
    </tr>
    <tr>
      <td>celda b2</td>
    </tr>
  </tbody>
</table>
```

Esto es una tabla

celdas a1 y a2 unidas	celda b1
	celda b2

### P2.19. Rowspan y Colspan

Crea una página html donde se muestre la dos tablas de los ejemplos anteriores y la siguiente tabla:

Esto es una tabla

celdas a1, a2, b1 y b2 unidas		celda c1
		celda c2
celda a3	celda b3	celda c3

## P2.20. Más ejercicios de Rowspan y Colspan

Crea una página html donde se muestren la siguiente tablas. Utiliza los atributos rowspan y colspan en las celdas que sean necesarios. Aplica los alineados y colores que se indican.

	1ª Eval		2ª Eval		3ª Eval	
	Aprobados	Suspendidos	Aprobados	Suspendidos	Aprobados	Suspendidos
Informática	20	2	19	3	22	0
Matemáticas	10	12	15	7	12	10

Horas	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-9:00	Matemáticas	Lengua	Música		Matemáticas
9:00-10:00		Inglés	Inglés	E.Física	
10:00-10:30	Recreo				
10:30-11:30	E.Física	Valenciano	Lengua	Informática	Inglés
12:30-13:30	Música	E.Física		Matemáticas	Informática
13:30-14:30		Informática			

### P2.21. Repasando Rowspan y Colspan

Crea una página html donde se muestren la siguiente tablas. Utiliza los atributos rowspan y colspan en las celdas que sean necesarios.

Tabla 1

Lorem		
ipsum	dolor	sit

Tabla 2

amet	consetetur		
sadispicing	elit,	sed	diam
	volupta.	At	vero

Tabla 3

eos	et	accusam
		et
	justo	duo
		dolores

Tabla 4

et	ea	
	rebum.	Stet
	clita	kasd

Tabla 5

gubergren	no
	sea
sanctus	takimata

Tabla 6

Duis	autem		
	eum	iriure	dolor
		in	
		hendrerit	
	in		vel

## Listas

En HTML disponemos de dos tipos de listas: listas desordenadas, cuyos elementos no están numerados, y listas ordenadas, cuyos elementos están numerados de alguna forma, ya sea con números, letras, números romanos, etc.

### Listas desordenadas

Una lista desordenada comienza con la etiqueta `<ul>`, y cada elemento de la lista comienza con la etiqueta `<li>`.

Los elementos de la lista están marcados con viñetas, normalmente, pequeños círculos negros.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

El código anterior se vería así:

- Coffee
- Milk

### Listas ordenadas

La única diferencia entre una lista desordenada y una ordenada es que la ordenada comienza con la etiqueta `<ol>`, en lugar de la etiqueta `<ul>`.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Se vería así:

1. Coffee
2. Milk



### Listas de definición

Se utilizan para insertar definiciones. Se utilizan la etiquetas <dl> y <dt>:

- las etiquetas <dl> ...</dl> delimitan la lista completa
- las etiquetas <dt> ...</dt> delimitan los términos
- las etiquetas <dd> ...</dd> delimitan las definiciones

### **P2.22. Listas ordenadas y desordenadas y de definición**

Crea en página HTML donde se muestre:

- Una lista desordenada donde se muestren los días de la semana.
- Una lista ordenada donde se muestren los meses del año
- Una lista de definición con la definición de HTML.

### **P2.23. Listas anidadas**

Crea en página HTML donde se muestren las siguientes listas anidadas:

- Ítem 1
  - 1. Ítem 1 de sublista
    - Ítem 1 de sub-sublista
    - Ítem 2 de sub-sublista
  - 2. Ítem 2 de sublista
    - 1. Ítem 1 de sub-sublista
    - 2. Ítem 2 de sub-sublista
  - 3. Ítem 3 de sublista
  - 4. Ítem 4 de sublista
- Ítem 2
  - Ítem 1 de sublista
  - Ítem 2 de sublista

#### **Listas anidadas**

- HTML
  - 1. Historia
  - 2. Etiquetas Básicas
  - 3. Ejemplos sencillos
- CSS
  - 1. Conceptos e Importancia
  - 2. Reglas CSS
  - 3. Frameworks(Bootstrap,Fundation)
- JavaScript
- PHP y MySQL
- Servidor WEB

## Formularios

Los formularios HTML se utilizan para pasar datos a un servidor. Un formulario HTML puede contener elementos de entrada como campos de texto, casillas de verificación, radio botones, etc. Para crear un formulario HTML utilizamos la etiqueta `<form>` y dentro del `<form>` introduciremos los campos que nos interesen: `<input>`, `<select>`, etc.

En la siguiente tabla podemos ver los principales campos que podemos introducir en un formulario.

Tag	Description
<code>&lt;form&gt;</code>	Defines an HTML form for user input
<code>&lt;input&gt;</code>	Defines an input control
<code>&lt;textarea&gt;</code>	Defines a multiline input control (text area)
<code>&lt;label&gt;</code>	Defines a label for an <code>&lt;input&gt;</code> element
<code>&lt;fieldset&gt;</code>	Groups related elements in a form
<code>&lt;legend&gt;</code>	Defines a caption for a <code>&lt;fieldset&gt;</code> element
<code>&lt;select&gt;</code>	Defines a drop-down list
<code>&lt;optgroup&gt;</code>	Defines a group of related options in a drop-down list
<code>&lt;option&gt;</code>	Defines an option in a drop-down list

### **P2.24. Introducción a formularios**

Crea en página HTML donde se muestre el siguiente código de formulario:

```
<form action="" method="get">
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
  <p>Año de nacimiento: <input type="number" name="nacido" min="1900"></p>
  <p>Sexo:
    <input type="radio" name="hm" value="h"> Hombre
    <input type="radio" name="hm" value="m"> Mujer
  </p>
  <p>
    <input type="submit" value="Enviar">
    <input type="reset" value="Borrar">
  </p>
</form>
```

Sube una captura de pantalla del resultado del formulario.  
¿Qué ocurre si rellenamos el formulario y le damos a enviar? ¿Por qué ocurre esto?

En el atributo action podemos redirigir los datos de nuestro formulario a una web .php o a por ejemplo a una dirección de correo-e.

Añade el siguiente código a tu atributo action: "mailto: [mcampos3@iesfuentesanluis.org](mailto:mcampos3@iesfuentesanluis.org)" para que me lleguen tus datos del formulario a mi correo.

## Una página web dentro de otra

Para introducir una página web dentro de otra, utilizamos el elemento `<iframe>`: `<iframe src="URL"></iframe>`

La URL apunta a la dirección de la página que queremos mostrar dentro de esta. Podemos indicar el alto y ancho del `<iframe>` mediante los atributos `height` y `width`, normalmente expresados en píxeles, aunque también podemos utilizar porcentajes.

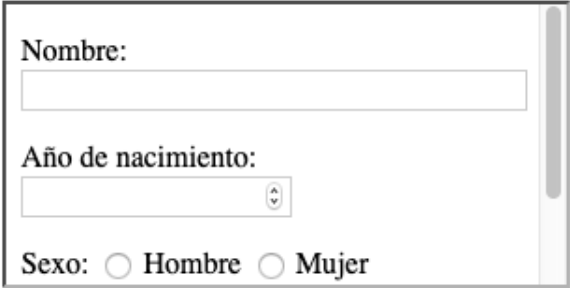
```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

Un `<iframe>` puede ser utilizado como destino de un enlace, para ello, sólo tenemos que indicar el nombre del `<iframe>` en el atributo `target` del enlace.

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>  
<p><a href="http://www.google.es" target="iframe_a">Ir a  
google</a></p>
```

### **P2.25. Una web dentro de otra**

Fijándote en el ejemplo anterior crea una web que muestre lo siguiente:



Nombre:

Año de nacimiento:

Sexo: ☐ Hombre ☐ Mujer

[Ir a intranet](#)

- Una página HTML donde hay un enlace a la intranet del centro
- Arriba aparecerá incrustada tu práctica P2.24 (la del formulario). Cuando pinches sobre Ir a intranet se abrirá la intranet en el iframe que has creado.

## 6. Definiendo la estructura del cuerpo

Organizando el diseño de la página

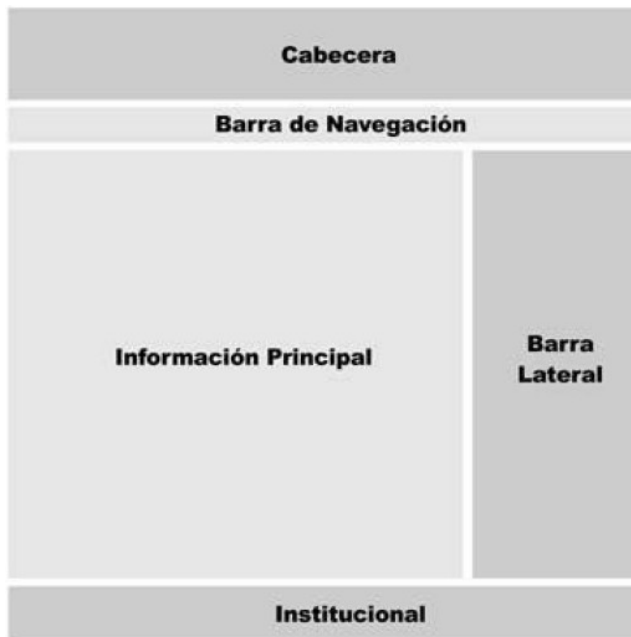


Imagen 1: diseño web clásico

En la imagen anterior podemos ver un diseño común encontrado en la mayoría de los sitios webs. A pesar del hecho de que cada diseñador crea sus propios diseños, en general, podremos identificar las siguientes secciones en cada sitio web estudiado.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro. Esta sección presenta la información principal de la página y la mayoría de las veces está dividida en varias filas y columnas. En el ejemplo de la imagen se utilizaron sólo dos columnas: *Información Principal* y *Barra Lateral*, pero esta sección es extremadamente flexible y cada diseñador la adapta a sus necesidades.

Evidentemente, si mostramos la páginas anteriores en el navegador, el resultado obtenido no será el esperado, ya que, cada elemento aparecerá uno debajo del otro. Para solucionar este problema tenemos que crear nuestra hoja de estilos CSS que le dé a la página el aspecto que nos interese.