



## 2º de Bachillerato

# Tecnologías de la Información y Comunicación

### Contenidos

### Diseño web funcional.Prevencción y buenas prácticas: Hojas de estilo "CSS"

```
h1 { color: white;
background: orange;
border: 1px solid bla
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```

CSS

Imagen en Wikipedia de [www](#) bajo  
licencia [Creative Common](#)

En un principio diseñar una página web consistía en unir todo tipo de etiquetas, mezclando el diseño de estructura con el formato, hoy en día la separación de la estructura de un documento (HTML) de su diseño o presentación (CSS) ofrece múltiples ventajas.

En el trimestre anterior estudiamos las etiquetas fundamentales de HTML, incluso estamos familiarizados con algunas etiquetas del nuevo estándar, el HTML5. ¿Sabemos ya diseñar un sitio web? Sólo parcialmente. En la actualidad ya no basta con manejar HTML para realizar estas tareas; necesitamos conocer también el estándar que lo complementa, CSS, acrónimo de hojas de estilo en cascada.

¿Qué ventajas nos ofrece las CSS? Su utilización es muy práctica ya que un solo archivo CSS puede modificar el aspecto de varios documentos HTML, de modo que, en caso de realizar cambios en el diseño general, solo sería necesario modificar dicho archivo y no cada uno de los documentos HTML.

¿Qué aprenderemos en este tema? Crearemos nuestras propias CSS comprobando que efecto tiene tras vincularlas con páginas HTML. En el siguiente ejemplo observaremos cómo cambia una misma página HTML tras asociarla a diferentes CSS que definen cada una un estilo distinto. Para ver los efectos hace clic en las distintas hojas de estilo del menú, [demoCss](#).

### Para saber más

#### HTML5 & CSS3

El uso de hojas de estilo es obligatorio en las páginas web creadas con *HTML5*, este estándar define la estructura de la página y el CSS el estilo y el diseño.

El navegador, a su vez, tomará cada una de las partes y las mezclará, dando el resultado deseado: una página con sentido semántico gracias a HTML y con una apariencia estética apropiada mediante el estilo CSS.

Las hojas de estilo han ido evolucionando desde la primera versión del estándar en el 2001. En la actualidad la versión 3 de este conjunto de normas está aún en desarrollo, pero la mayoría de los navegadores son capaces de ejecutar todas las propiedades principales de CSS3.



# 1. Aplicando estilos



## ¿Qué son las hoja de estilo y por qué las usamos?

Una hoja de estilo en cascada, del inglés **Cascading Style Sheets "CSS"**, es un estándar que especifica un conjunto de reglas en las que definimos el formato que va a tomar un determinado elemento de una página web. Es decir, describe como los elementos HTML serán mostrados en la pantalla.

Las hojas de estilo se utilizan para definir el estilo de una página web, incluyendo el diseño, la disposición y las variaciones en la imagen para los diferentes dispositivos y tamaños de pantalla. Con una CSS, podemos cambiar fácilmente la apariencia de un sitio web sin afectar al código HTML.



Imagen en pixabay de  
Valco bajo licencia Creative  
Common CCO

## Funcionamiento

Hasta el momento sabemos que una regla CSS se puede aplicar mediante el uso del parámetro **style**. Por ejemplo, la línea mostrará el título de cabecera de color azul.

```
<h1 style="{color:blue}">Bienvenidos </h1>
```

Ese método es válido, pero es difícil de mantener. Si el documento tiene con 100 títulos tendríamos que aplicar el formato a cada uno de ellos. ¿Y si después nos cansamos y los queremos poner todos rojos? Tendríamos que ir cambiándolos todos uno a uno. Tedioso ¿no?

Las hojas de estilo son la solución, recopilan todas las reglas en un mismo espacio haciendo que sea más sencillo su mantenimiento y modificación. En nuestro ejemplo, solo deberíamos modificar una línea no 100. De este modo, una regla que hiciese exactamente lo mismo quedaría así:

```
h1 { color:blue; }
```

La línea anterior, situada en la cabecera (entre **<head>** y **</head>**) de la página, haría que se cambiasen de color todos los **<h1>** de nuestra página. Como se puede observar es más rápido y más sencillo. Incluso el mantenimiento de la página se vuelve trivial, ya que el cambio de color se reduciría a poner *green* (verde) donde ahora pone *blue* (azul).



Imagen en pixabay de  
clker bajo licencia Creative  
Common

### ¿Cómo se vinculan las CSS y el código HTML?

Cuando un navegador encuentra una hoja de estilos o CSS, formatea el documento HTML a partir de lo establecido en dicha CSS. Existen tres formas de insertar los estilos, el método más recomendable y habitual es el primero, mediante archivos externos.

#### Hoja de estilos externa

Esta es una de las opciones más comunes, ya que posteriormente es posible utilizar el mismo fichero CSS para aplicar el mismo estilo a otras páginas html. Con una hoja de estilo externa, podemos cambiar la apariencia completa de un sitio web modificando solo un fichero.

Cada página html incluye una referencia al fichero CSS mediante la etiqueta **<link>** que siempre se sitúa dentro de la etiqueta **<head>**.

```
<head>
<link rel="stylesheet" type="text/css" href="css/estilos.css">
</head>
```

La etiqueta **<link>** consta de los siguientes atributos :

- **rel**: indica la relación entre el documento y el archivo que se carga; en este caso nuestra hoja de estilos.
- **type**: indica el tipo de información. Tanto este valor como el de **rel**, son valores fijos (debes mantenerlos).
- **href**: (es el valor que debes modificar) como en los enlaces normales, indica la URL del archivo. Al encontrarse en la misma carpeta, sólo indicaremos el nombre; pero si estuviese en otro lugar o incluso en otro servidor, crearíamos la ruta siguiendo las normas de los enlaces. En este caso hemos llamado a la CSS "estilos.css" y además la hemos ubicado dentro de una carpeta css.
- **media** (opcional): especifica a qué medio se aplicará la hoja de estilo. En este caso se aplica a todos los medios, pero podríamos tener hojas de estilo diferentes para impresión (print), móviles (handheld), televisiones (tv), etc.

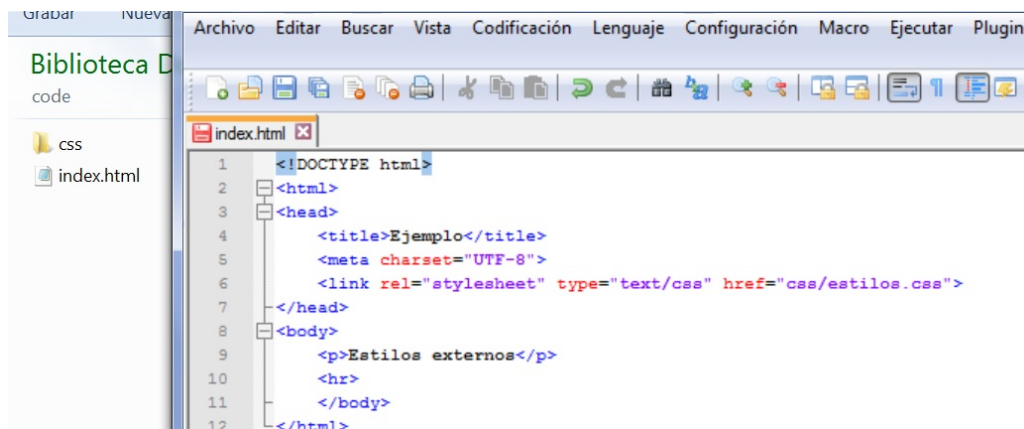


Imagen ejemplo página html vinculada con css externa de creación propia.

Una hoja de estilos externa se codifica con editor de texto como puede ser notepad++. El fichero no contiene etiquetas html y debe ser guardado con la extensión **".css"**. Ya veremos en el siguiente punto cómo se crean estos ficheros, un ejemplo de la CSS estilos.css podría ser:

```
body {
background-color:lightblue;
}

h1 {
color: navy;
margin-left: 20px;
}
```

¿Sabrías decir qué efectos aplica esta hoja de estilos? El color de fondo será azul claro y todos los textos de cabecera h1 tendrán un margen izquierdo de 20 px y el texto de color azul oscuro.

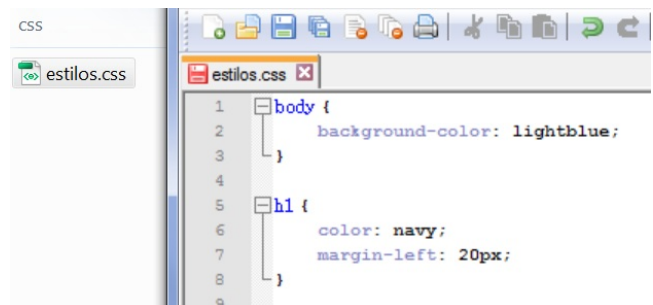


Imagen ejemplo archivo css de creación propia.

#### Hoja de estilos interna

Una hoja de estilos interna solo es utilizada para esa página html concreta aplicando un único estilo, se define dentro de la sección **<head>** de la página HTML a través del atributo **<style>**. Siguiendo el ejemplo anterior, el código completo quedaría de la siguiente manera:

```
<head>
<style>
body {
background-color: lightblue;
}
```

```
h1 {  
  color: navy;  
  margin-left: 20px;  
}  
  
</style>  
  
</head>
```

### **Estilos en línea**

Son utilizados para aplicar un estilo concreto a un determinado elemento. Se debe añadir el atributo `style`, que incluye las propiedades CSS. Es la forma que hemos visto en los ejemplos del tema HTML. Para modificar las propiedades de `<h1>` según el ejemplo:

```
<h1 style="color:navy;margin-left:20px;">This is a heading.</h1>
```

## *Para saber más*

### **Orden en cascada**

¿Qué estilo se utilizará cuando se especifica más de un estilo de un elemento HTML ?

En términos generales se puede decir que todos los estilos siguen un orden en cascada establecido por las reglas siguientes, en las que el número uno tiene la más alta prioridad :

1. Estilo en línea ( dentro de un elemento HTML ).
2. Hojas de estilo externas e internas (en la sección de cabecera ).
3. Establecido por defecto por el propio navegador.

Por lo tanto, un estilo en línea ( dentro de un elemento HTML específica ) tiene la prioridad más alta, lo que significa que va a anular un estilo definido dentro de la etiqueta `< head>`, o en una hoja de estilos externa, o bien un valor por defecto del navegador.



Imagen en pixabay de  
[unsplash](#) bajo licencia  
[Creative Common](#)

## 2. Sintaxis y selectores



### ¿Cómo se define una regla CSS?

Consta de varias partes: un selector y un bloque de declaraciones.

- El selector hace referencia al elemento html al que queremos aplicar el estilo.
- El bloque de declaración contiene una o más declaraciones separadas por punto y coma (;).
- Cada declaración incluye el nombre de la propiedad CSS y un valor separados por dos puntos (:).
- Una declaración CSS siempre termina en punto y coma, y los bloques de declaración están contenidos entre llaves.



¡Veamos un ejemplo!

Creemos una regla css que permita mostrar el texto rojo de todos los elementos párrafo <p> y además que queden alineados a la derecha. De esta forma todos los elementos <p> que aparezcan en la página html se verán afectados.

```
p {  
    color: red;  
    text-align: right;  
}
```

### Selectores

Los selectores CSS son usados para "encontrar" o "seleccionar" elementos basados en elemento nombre, id, class, attribute y otros que estudiaremos a continuación.

#### Selector elemento

Este selector se basa en el nombre y es el ejemplo que hemos visto anteriormente. Nosotros podemos seleccionar todos los <h2> de una página para que el texto que contenga sea azul y esté centrado. Deberíamos incluir la siguiente regla css:

```
h2 {  
    color: blue;  
    text-align: center;  
}
```

#### Selector id

Podemos diferenciar un único elemento entre todos los demás de una página mediante el atributo **id** (abreviatura de "identificador") de un elemento HTML. Con **id** asignaremos un nombre único que podremos posteriormente emplear en nuestra sección de estilos, precediéndolo en este caso del signo #.

Por ejemplo, la siguiente regla solo se aplicará al elemento en concreto cuyo id sea "para1".

```
#para1 {  
    color: red;  
    text-align: center;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Ejemplo</title>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" type="text/css" href="css/estilos.css">  
  </head>  
  <body>  
    <p>Estilos externos</p>  
    <p id="para1"> Estilo texto centrado y rojo</p>  
    <hr>  
  </body>  
</html>
```

Imagen de código html de creación propia.



Imagen resultado en navegador de creación propia.

## Selector class

Pensemos ahora en una página en la que queremos cambiar la apariencia de algunos <h2>, pero no de todos. ¿Qué opciones tenemos? Para ello tenemos un parámetro general de HTML denominado class ("clase" en español, en el sentido de "tipo"). Este parámetro nos sirve para crear un conjunto de etiquetas que forman parte del mismo grupo. El objetivo es que luego podremos referirnos a esa clase de forma conjunta.

Para seleccionar elementos con una clase específica, se precede de un punto (.) seguido del nombre de la clase. En el siguiente ejemplo de hoja de estilos, todos los elementos HTML con class = "centro" será de color rojo y alineado al centro:

```
.centro {  
    color: red;  
    text-align: center;  
}
```

Observa en el código html, ahora todos los elementos con atributo class="centro" tendrán este estilo, la diferencia del id con respecto a class es que este es único, solo puede ser asignado a un elemento en cada página html.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Ejemplo</title>  
    <meta charset="UTF-8">  
    <link rel="stylesheet" type="text/css" href="css/estilos.css">  
  </head>  
  <body>  
    <h1 class="centro"> Cabecera de estilos</h1>  
    <p class="centro">Estilos externos</p>  
    <p class="centro"> Estilo texto centrado y rojo</p>  
    <p> Sin estilos </p>  
    <hr>  
  </body>  
</html>
```

Imagen de código html de creación propia.



Imagen resultado en navegador de creación propia.

Se puede también especificar que una clase afecte a un elemento HTML en concreto. En el siguiente ejemplo, solamente <p> elementos con class = "centro" estarán alineados - centro: ¿cómo afectaría este cambio en la hoja de estilos sobre el html anterior? Mira la imagen para saber si estabas en lo cierto.

```
p.centro {  
    text-align: center;  
    color: red;  
}
```

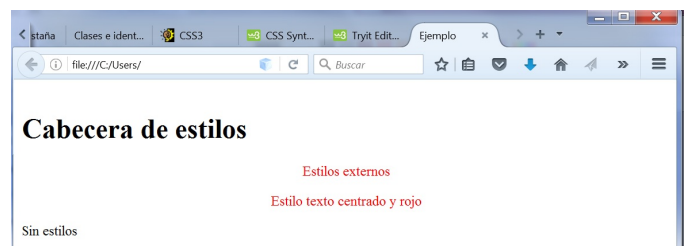


Imagen resultado en navegador de creación propia bajo.

## Para saber más

Existe además una serie de reglas e indicaciones para trabajar con los selectores, los principales son:

- El nombre de **id** y de **class** no puede empezar con un número
- Se pueden combinar los parámetros **class** e **id** en un mismo elemento, con lo que conseguiremos que se le apliquen tanto las reglas definidas para la **clase** como para el **identificador**. En el siguiente ejemplo da como resultado una mezcla de todas las reglas que hemos aplicado hasta ahora, tanto las del elemento <h1>, como las de la **clase** y las del **identificador**.

```
<h1 class="centrado" id="anexo">Bienvenidos</h1>
```

- El signo \* se emplea para definir a todo el conjunto de etiquetas de una página web. Nos sirve para indicar "esto se aplica a cualquier etiqueta". En el siguiente ejemplo hace que cualquier etiqueta de la página tome ese tipo de letra.

```
* {font-family:san-serif; }
```

- También podemos emplear la coma para agrupar varias etiquetas, clases e identificadores. Por ejemplo:

```
p, h1, h2 {font-family: times, word-spacing: 2px;}
```

### 3. Diseño web con estilos

En este tema nos sumergiremos definitivamente en el conocimiento de las diferentes propiedades de las hojas de estilo que nos permiten controlar la apariencia de una página web. Por ejemplo, vamos a tratar las propiedades, valores y selectores CSS más habituales y utilizados para la maquetación de texto, disposición de <div> y personalización de enlaces.

Existen propiedades CSS que afectan al texto, otras a las cajas (<div>) que contienen elementos, así como otras que afectan a los enlaces entre otros. Es de obligado cumplimiento tener bien claro el funcionamiento de los **selectores CSS** y **la sintaxis** estudiada en el punto anterior.

Como punto de partida necesitaremos un documento base con abundante texto y diferentes etiquetas, donde iremos realizando las modificaciones. Probablemente nos resulte cómodo emplear un editor web, como el que hemos venido empleando hasta ahora.



Imagen en Pixabay de [geralt](#) bajo licencia [Creative Common CCO](#)

No obstante todas las reglas que vamos a analizar pueden ser aplicadas directamente mediante una hoja de estilos externa, nuestro método favorito de trabajo con estilos.

#### *Para saber más*

##### **Navegadores: css soportados**



Imagen en Pixabay de [isomar](#) bajo licencia [Creative Common CCO](#)

En el diseño web es fundamental tener en cuenta que nuestra página será visualizada a través de distintos navegadores existentes en el mercado. Es imprescindible comprobar que los elementos CSS que utilizamos están soportados por los navegadores.

Para averiguar qué versión de los principales navegadores web dan soporte a un elemento CSS concreto, es posible buscar el texto CSS3 browser support en Google. Otra opción es acudir a una [web específica](#) de consulta que informa del soporte de distintas propiedades CSS en diferentes navegadores.

Una práctica extendida es el uso de prefijos de navegador o prefijos comerciales (*vendor prefixes*) a un prefijo que se antepone a una regla CSS destinado a que dicha regla sea leída y aplicada exclusivamente por un navegador concreto (por ejemplo Chrome) pero no por el resto de navegadores.

El uso de prefijos suele aplicarse a propiedades que se encuentran en fase experimental o que aún no se han convertido en un estándar.



Los colores en CSS se pueden especificar de la siguiente forma.

#### ● Nombre de colores

Un nombre predefinido ( en inglés) como "red" o "blue".Actualmente todos los navegadores modernos soportan 140 nombres de colores. El siguiente enlace nos muestra la [gama cromática de colores](#) disponibles. El nombre no es sensible a las mayúsculas por lo tanto podrías escribir "red" o "RED".

#### ● RGB(Red, Green, Blue)

Los colores de código RGB usan la fórmula que consiste en dar valor de 0 a 255 a los colores base rojo, verde y negro, para así formar el resto. Por ejemplo, rgb(255,0,0) correspondería al color rojo.

#### ● Hexadecimal

Los valores de RGB pueden también especificarse mediante el sistema hexadecimal de tal forma: #RRGGBB, donde RR (rojo), GG (verde) and BB (azul) son valores hexadecimales entre el 00 y el FF (equivale al decimal 0-255). Por ejemplo, #FF0000 sería rojo porque el valor FF es el más alto y el resto de colores están a 00.

#### ¿Cómo aplicamos color ?

##### Texto

En CSS existe la propiedad color: ; para indicar el color de un texto. Por defecto, el texto tendrá el color especificado en el selector body. El siguiente ejemplo, define el color de texto azul por defecto salvo para el texto de la cabecera <h2> que sería rojo.

```
body{
    color:blue;
}
h2{
    color:red;
}
```

##### Fondo

Si queremos especificar el color de fondo de un elemento debemos utilizar la propiedad background-color: (valor); Por ejemplo, para especificar un color azul oscuro (#00008B) al fondo de la página, lo añadimos como propiedad del elemento body.

```
body{
    background-color:#00008B;
}
```









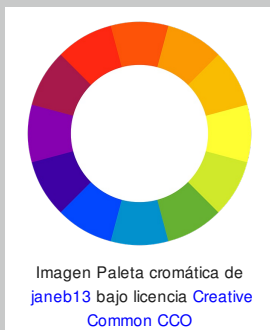
| Color Name | HEX     | Color   |
|------------|---------|---|
| Black      | #000000 |  |
| Navy       | #000080 |  |
| DarkBlue   | #00008B |  |
| MediumBlue | #0000CD |  |
| Blue       | #0000FF |  |
| DarkGreen  | #006400 |  |
| Green      | #008000 |  |
| Teal       | #008080 |  |

Imagen Colores CSS de [W3C](#) bajo licencia [Creative Common CCO](#)

## Para saber más



En el diseño web es fundamental utilizar una adecuada paleta cromática. El color es un factor influyente a la hora de transmitir el mensaje que deseas en tu sitio web. El color representa un estímulo visual potente siendo significativo el efecto que puede generar en el usuario. El uso del color se extiende mucho más allá de destacar simplemente ciertas zonas o ciertos fragmentos de texto. Por ejemplo, para una página de salud los colores utilizados serán básicamente colores de la misma gama cromática: azul, verde, gris y blanco, junto con el negro para textos.

Ser capaz de seleccionar los colores correctos es clave para un diseño eficaz e impactante. En este caso se escoge el azul al considerarse institucional además provoca en las personas que se sientan relajadas y dispuestas de aceptar nuevas situaciones e influencias. Por otro lado el color verde, tiene una influencia sobre las personas es de tranquilidad.

Hoy en día, existen múltiples herramientas que nos ayudan a la hora de seleccionar los colores. Algunas de las mejores son [paleton](#) o [circulo cromático](#).

A continuación estudiaremos las propiedades del texto a definir en un página web. En el punto anterior aprendimos como cambiar el color de un texto.

```
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}

div {
  text-align: justify;
}
```

Imagen alineación de texto creación propia

### Alineación

La propiedad **text-align** se utiliza para establecer la alineación horizontal de un texto , este puede ser alineado a izquierda o derecha , centrado o justificado . El siguiente ejemplo muestra las distintas opciones, la izquierda es la alineación por defecto. Cuando la propiedad text-align es " justify" , cada línea se estira de manera que cada línea tiene la misma anchura , y los márgenes izquierdo y derecho son rectos (como en revistas y periódicos ) .

### Transformación

La propiedad **text-transform** es utilizada para especificar las letras mayúsculas (**uppercase**) o minúsculas (**lowercase**) del todo el texto o de la primera letra de una palabra en mayúsculas (**capitalize**). Por ejemplo, la siguiente regla cambiaría el texto del párrafo a mayúsculas.

```
p.uppercase {
  text-transform: uppercase;
}
```

### Sangría

La sangría o indexación de la primera línea de texto se especifica con la propiedad **text-indent**.

```
p {
```

```
  text-indent: 50px;}
```

### Fuentes

En CSS, hay dos tipos de familias que definen una tipografía:

- **generic family** : un grupo de fuentes similar a "Serif" o "Monospace".
- **font family** : especifica una fuente de familia tal y como "Times New Roman" or "Arial".

En las pantallas de ordenador , las fuentes Sans serif que se consideran más fácil de leer que las fuentes serif .

### Tipo de letra

La propiedad **font-family**, seguida de una tipografía concreta o de una serie de tipos de letra separadas por comas, define la apariencia del elemento HTML determinado.

La práctica nos irá marcando la pauta de qué tipo será el más adecuado en cada caso. Normalmente siempre indicaremos un grupo de fuentes, ya que, si el navegador no tiene instalada la primera, recurrirá a las alternativas, hasta llegar a la última que hemos indicado, que define una fuente genérica.

Veamos un ejemplo, en el que definimos un formato concreto para toda la página y otro para los títulos.

```
body {
  font-family: Helvetica, Arial, Sans-serif;
}

h1 {
  font-family: Georgia, "Times New Roman", Times, Serif;
}
```

### Tamaño de letra

Para el tamaño de letra se emplea la propiedad **font-size**, en el valor, además de un número, indicaremos la medida que empleamos

```
body {
  font-size: 40px;
}
```

Hay diferentes opciones para especificar las medidas, los más importantes son:

- **px, cm, pt o in**: para indicar medidas en **píxeles**, centímetros, puntos o pulgadas. Tenemos el control total sobre el texto pudiendo utilizar la herramienta "zoom". Tiene como inconveniente que cada navegador puede mostrar diferencias en este sentido, aunque es adecuado para estilos de impresión.
- **em**: es el valor recomendado por el W3C, si queremos emplear valores fijos. 1em es su valor base y equivale a 16 píxeles. 2em sería el doble de tamaño .5em la mitad (es lo mismo que 0.5em), etc.

Existe otra opción , que consiste en combinar porcentajes y em. Muestra el mismo tamaño del texto en todos los navegadores , y permite que dichos navegadores puedan ampliar o cambiar el tamaño del texto.

### Texto en negrita

La propiedad **font-weight** especifica si el texto irá en negrita o no, con diferentes grados:

- **bold**: negrita.
- **bolder y lighter**: valor relativo que aplica un efecto más pronunciado en el primer caso o menos en el segundo.
- **100..900**: secuencia entre 100 y 900 el grado de "negrita".
- **normal**: evita que se muestre en negrita.

| Generic family | Font family                   |
|----------------|-------------------------------|
| Serif          | Times New Roman<br>Georgia    |
| Sans-serif     | Arial<br>Verdana              |
| Monospace      | Courier New<br>Lucida Console |

Imagen tipo de fuentes creación propia

En el siguiente ejemplos se definen reglas CSS para párrafos "normal" y párrafos cuyo textos se verán en negrita simple.

```
p.normal {  
    font-weight: normal;  
}  
p.negrita{  
    font-weight:bold;  
}
```

En el código html haríamos referencia de esta forma:

```
<p class="normal">Párrafo normal</p>  
<p class="light">Párrafo en negrita</p>
```

Párrafo normal  
Párrafo en negrita

### Estilo de fuente

La propiedad **font-style** permite aplicar cursiva al texto. El valor **italic** transforma el texto en cursiva , el valor **oblique** gira ligeramente el texto y el valor **normal** retira cualquier propiedad cursiva.

En el siguiente ejemplo creamos tres reglas CSS para definir los tipo de párrafo según su estilo de fuente:

```
p.normal {  
    font-style: normal;  
}  
p.italica {  
    font-style: italic;  
}  
p.oblicua {  
    font-style: oblique;  
}
```

## Para saber más

### Tipografías

HTML5 permite hacer uso de fuentes (tipografías) poco comunes sin que el usuario las tenga que tener instaladas en su disco duro y sin tener que convertirlas en imagen.

En este caso basta con incluir el fichero de la fuente a utilizar correspondiente en una carpeta y crear la referencia a dicha fuente desde el código CSS3. Se deben seguir los siguientes pasos:

1. Decidimos el formato de fuente a utilizar (existen varios formatos de fuentes):

- TrueType (**ttf**): formato más estándar y aceptado por todos los navegadores, excepto Internet Explorer.
- Embedded OpenType (**eot**): formato de fuente que sólo utilizar Internet Explorer.
- WebOpen Font (**woff**): formato de fuente oficial de la W3C, no soportado por navegadores antiguos.

2. Copiamos las fuentes que vayamos a utilizar (ficheros ttf/eot): las pegamos en la misma carpeta donde tenemos el fichero .CSS.

3. Referenciamos las fuentes con CSS3: Dentro del código CSS añadimos la referencia utilizando el selector '@font-face'. El nombre que indicamos con el atributo 'font-family' es el nombre que podemos asignarle según nuestras preferencias.

4 Aplicamos la tipografía: Si queremos que un elemento utilice una de las fuentes referenciadas con @font-face utilizamos la propiedad **font-family** para indicar qué tipografía de las referenciadas en el punto anterior queremos utilizar.



Imagen en Pixabay de Fill bajo licencia [Creative Common CCO](#)

```
#apartado_2{  
    font-family:fuentes_ieda1;  
    font-size:38px;  
    color:white;  
}
```

Imagen de aplicación de tipografía de creación propia bajo licencia [Creative Common CCO](#)

```
@font-face{  
    font-family: fuentes_flor;  
    src: url('fleur.eot');  
    src: url('fleur.eot?#iefix') format('embedded-opentype'), url('fleur.ttf')  
    format('truetype');  
}  
  
@font-face{  
    font-family: fuentes_pigeon;  
    src: url('pigeon.eot');  
    src: url('pigeon.eot?#iefix') format('embedded-opentype'), url('pigeon.otf')  
    format('opentype');  
}
```

Imagen de referenciación de fuentes de creación propia bajo licencia [Creative Common CCO](#)



### 3.3 Imagen de fondo



En HTML existen algunas propiedades CSS que nos permiten configurar el fondo de página, ya sea de un simple color (opaco o con transparencia) como estudiamos anteriormente, una imagen o múltiples imágenes una encima de la otra. Para definir una imagen de fondo en un elemento, se utilizan las siguientes propiedades:

- **background-image:** su valor será la ruta a la imagen desde el directorio donde se encuentra el archivo css. Esta ruta debe incluirse entre comillas simples y paréntesis, precedida de la palabra url.

```
background-image:url('../imagenes/imagen.png');
```

- **background-repeat:** Si no añadimos ningún otro parámetro la imagen por defecto se irá repitiendo hasta ocupar toda la ventana html. Si por el contrario únicamente queremos que se repita una sola vez (horizontal o verticalmente), tenemos que utilizar la propiedad **background-repeat**, los valores son **repeat-x (horizontalmente)**, **repeat-y (verticalmente)** y **no-repeat**. Por ejemplo, si solo queremos que se repita horizontalmente deberíamos crear una regla como la siguiente:

```
body {  
    background-image: url('../imagenes/imagen.png');  
    background-repeat: repeat-x;  
}
```

- **background-position:** Esta propiedad establece la alineación vertical y horizontal. Se establecen los valores **left** (izquierda), **center**(centro) y **right**(derecha) para la alineación horizontal; en la alineación vertical las opciones son **top** (arriba) y **bottom** (abajo). Un ejemplo de alineación horizontal derecha y vertical arriba sería:

```
body {  
    background-position: right top;  
}
```

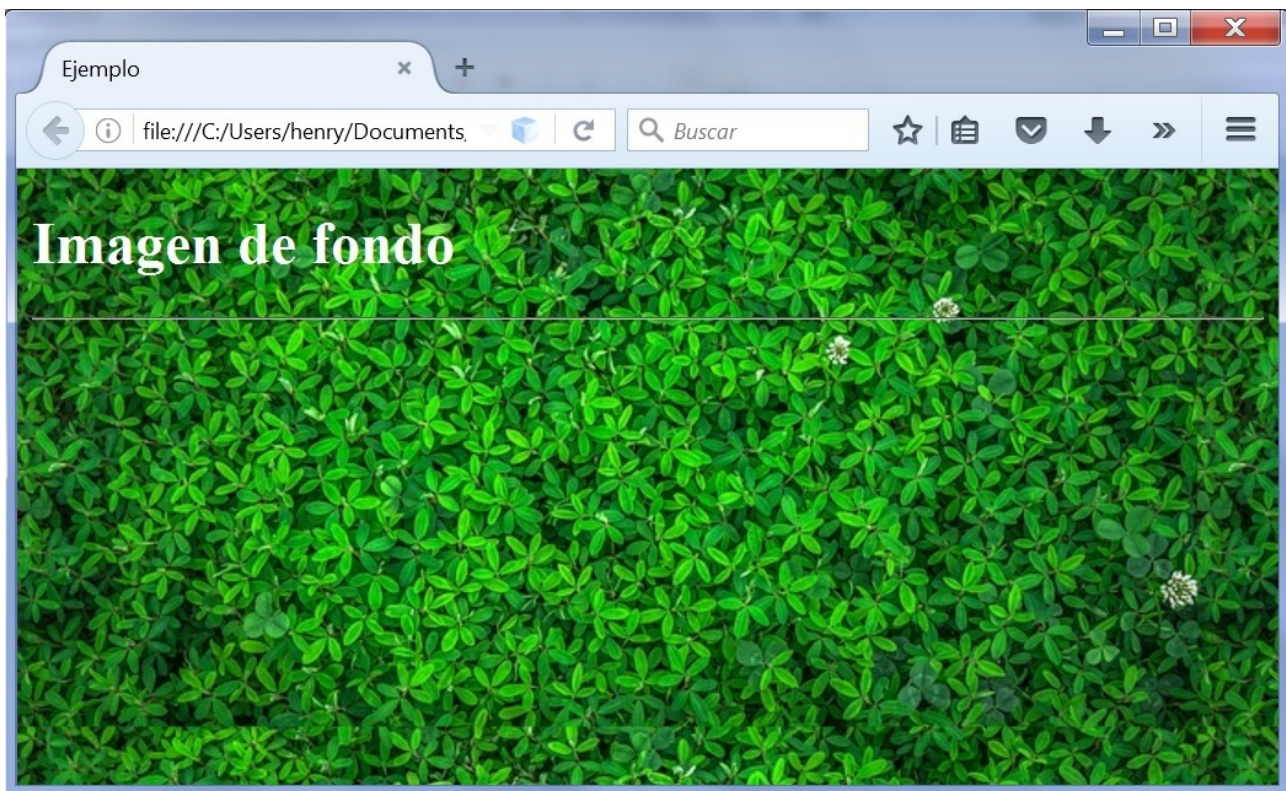


Imagen ejemplo imagen de fondo de creación propia



## 3.4 Enlace y lista

Los elementos CSS son extensos y variados, en este tema el objetivo es que conozcas los principales. En este apartado estudiaremos los enlaces y las listas.

### Enlaces

Existen reglas CSS aplicadas a los enlaces (*link*) para otorgarle un estilo e identificar el estado del enlace, por ejemplo, si ha sido visitado o no. Los posibles estados de un enlace son:

- **a:link** : normal, es decir no visitado.
- **a:visited** : enlace visitado.
- **a:hover** : enlace cuando el cursor se sitúa sobre él.
- **a:active** : enlace en el momento de ser pulsado.

El estilo a aplicar en cada uno de los estados se define a través de la propiedad **color** (asigna un color al texto), **text-decoration** (añade una línea debajo del texto) y **background-color** (especifica el color de fondo del link).

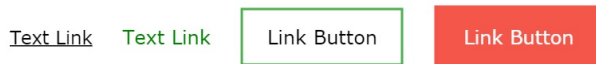


Imagen de enlaces con estilos en el navegador de creación propia bajo licencia Creative Common CCO

```
a:link {
  text-decoration: underline; /* Con línea*/
  color:#008CBA;
}

a:visited {
  text-decoration: none; /* Sin línea*/
  color: green;
}

a:hover {
  text-decoration: none;
  background-color: white;
  color: black;
  border: 2px solid #4CAF50;
  text-align: center;
}

a:active {
  text-decoration: none;
  background-color: #f44336;
  color: #FFFFFF;
  text-align: center;
}
```

Imagen de enlaces con estilos código fuente de creación propia.

### Listas

Recuerda que en HTML, hay dos tipos principales de listas: Listas no ordenadas (< ul >) y listas ordenadas (< ol >).

Las propiedades de la lista de CSS le permiten:

- Establecer diferentes marcadores de elementos de lista para las listas (propiedad **list-style-type** y **list-style-position**).
- Configurar una imagen como el marcador de la lista de elementos (**list-style-image**).
- Añadir colores de fondo para las listas y los elementos de la lista (**background**).

Los estilos se pueden combinar y especificarlos en una sola línea.

¡Veamos algunos ejemplos!

La siguiente regla define el estilo de una lista no ordenada, posición sangría y que en caso de que exista mostrará la imagen definida por la url, si hubiese algún problema sería un cuadrado.

```
ul {
  list-style: square inside url("sqpurple.gif");
}
```

### Para saber más

Puedes consultar todas las opciones en el estilo del línea en esta [web W3School](http://www.w3schools.com/css/css_list.asp).

#### Asignaturas en IEDA:

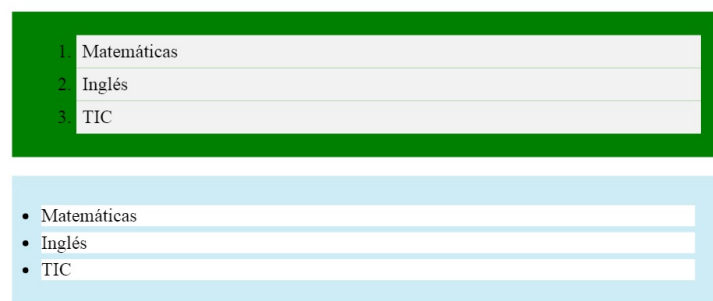


Imagen ejemplo de listas de creación propia bajo licencia Creative Common CCO

## 3.5 Tabla



Las tablas son un elemento muy utilizado en HTML y definir reglas CSS que mejoran su apariencia. ¿Recuerdas que elementos forman parte de el elemento tabla?

**<table>**: elemento superior que engloba al resto y permite definir una tabla. La tabla se compone de:

- **<tr>**: se emplea para definir cada fila de las tablas de datos. Habrá tantos **<tr>** como filas sean necesarias.
- **<td>**: se emplea para definir cada una de las celdas que forman las filas de una tabla, siempre dentro del elemento **<tr>**.
- **<th>**: se emplea para definir las celdas que son cabecera de una fila o de una columna de la tabla.

```
<table>
<tr>
  <th>Nombre</th>
  <th>Apellido</th>
</tr>
<tr>
  <td>Ana</td>
  <td>López</td>
</tr>
</table>
```

Imagen de tabla básica de creación propia

Veamos un ejemplo de la misma tabla con y sin estilos aplicados.

| Nombre  | Apellido | Nota |
|---------|----------|------|
| Antonio | García   | 4.5  |
| Marta   | Pozo     | 7.5  |
| Juan    | León     | 3.0  |
| Carla   | Blanco   | 8.5  |

Siendo la misma tabla, su apariencia mejora con respecto a la tabla sin estilo. A continuación estudiaremos que reglas CSS permiten dar estilo a este elemento HTML.

*¿Qué elementos de la tabla tendrán aplicados un estilo?*

| Nombre  | Apellido | Nota |
|---------|----------|------|
| Antonio | García   | 4.5  |
| Marta   | Pozo     | 7.5  |
| Juan    | León     | 3.0  |
| Carla   | Blanco   | 8.5  |

Imagen de una tabla sin estilo de creación propia

Imagen de tabla con estilo de creación propia

### Borde

Para especificar el borde de una tabla en CSS se utiliza la propiedad **border** con tres valores con un orden establecido, el grosor en píxeles (px), tipo de línea (por ejemplo discontinua) y el color. Por ejemplo si queremos asignar un borde de grosor 1px, sólido y color negro a una tabla tanto en el exterior como en su cuadrícula sería así:

```
table, th, td {
  border: 1px solid black;
}
```

| Nombre | Apellido |
|--------|----------|
| Ana    | López    |
| Juan   | Peña     |

Imagen tabla con borde de creación propia

La propiedad **border-collapse** establece si los bordes de la tabla deben ser colapsados en un solo borde.

### Alineación

La propiedad **text-align** establece la alineación horizontal (como izquierda "left", derecha "right" o centro "center") parte del contenido de **<th>** o **<td>**. La propiedad **vertical-align** establece la alineación vertical (como superior "top", inferior "bottom" o central "center") del contenido en **<th>** o **<td>**.

Por ejemplo, alineación horizontal de la cabecera a la izquierda y vertical superior.

```
th {
  text-align: left;
  vertical-align: top;
}
```

### Altura y anchura

La altura y la anchura de una tabla se definen por la propiedad **width** y **height** respectivamente. Por ejemplo, una tabla con anchura 100% y 30px para la cabecera se define con la siguiente regla CSS:

```
table {
  width: 100%;
}

th {
  height: 30px;
}
```

### Relleno

Para controlar el espacio entre el borde y el contenido de una tabla, se utiliza la propiedad **padding** (ejemplo *padding: 15px;*) en los elementos **<td>** y **<th>**.

### Color

Se aplica color a los elementos de la tabla HTML (**<th>**, **<td>**, **<tr>**) mediante las propiedades ya estudiadas, color al texto "**color**" y color de fondo "**background-color**". Por ejemplo, la cabecera de fondo rojo y letra blanca de una tabla tendría la siguiente regla CSS:

```
th {
  background-color: red;
  color: white;
}
```

Un estilo muy frecuente que ayuda a la lectura de las tablas es alternar las filas con un color de fondo (par e impar) . Se utiliza el selector **nth-child(even)** para filas pares y **nth-child(odd)** para filas impares. Veamos aplicada esta propiedad en el ejemplo final.

Analicemos el código fuente que permite dar la apariencia a nuestra tabla inicial con estilos, después de haber estudiado una a una la posibles propiedades a aplicar.

```
table {  
    border-collapse: collapse;  
    width: 100%;  
}  
  
th, td {  
    text-align: left;  
    padding: 8px;  
}  
  
tr:nth-child(even){background-color: #f2f2f2}  
/* Si no se especifica odd, por defecto es blanco*/  
  
th {  
    background-color: #4CAF50;  
    color: white;  
}
```

Imagen de código fuente de tabla con borde de creación propia



## 4. Maquetación con estilos

La separación entre el contenido de una página web y la disposición de cada pequeña parte de ese contenido no sería posible sin la intervención de las hojas de estilo en cascada y sus múltiples propiedades.

Las hojas de estilo ofrecen todas las posibilidades necesarias para definir una maquetación correcta que realce nuestros contenidos y nos proporcione un control absoluto del aspecto que toma cada parte, que mejora notablemente en CSS3.

En este cuarto punto estudiaremos como cada elemento HTML que podemos encontrar en una página web, se encuentra rodeado de una caja con varias propiedades que pueden ser modificadas. Además del contenido, cada elemento puede tener un borde (**border**) o no y, además, podemos modificar la distancia entre ese borde y el comienzo del contenido (**padding**) y la distancia entre el borde y el límite de la caja (**margin**).

En definitiva, disponemos de propiedades que nos permitirán posicionar los elementos (**position**) según la estructura elegida que a su vez está determinada por una altura y anchura ( **height y width**).

Caja de un elemento HTML

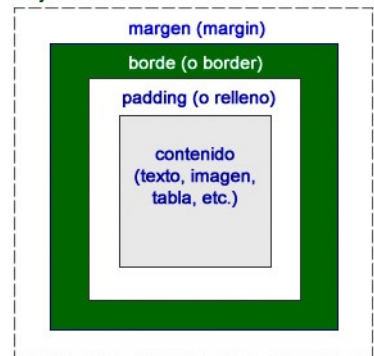


Imagen de tabla con estilo de creación propia

Antes de analizar cómo definir el tamaño con CSS debes recordar que hay dos tipos de elemento HTML según como se despliegan o visualizan:

- Elementos bloque: ocupan todo el ancho disponible.
- Elementos de línea: en función del espacio que ocupe el contenido.

La propiedad **display** es la más importante para controlar la plantilla, toma cuatro valores que permiten al navegador mostrar un elemento:

- **display:none**. Se utiliza comúnmente con JavaScript para ocultar y mostrar elementos.
- **display:hidden**. Oculta un elemento de cara al usuario y no al programador.
- **display:inline**. El elemento se comporta como elemento línea, si se incluyen <div> seguidos, estos se situarán uno junto a otro.
- **display:block**. El elemento se considera elemento bloque (<div> o <p>).

El tamaño de un elemento viene determinado por dos propiedades calculadas por el navegador el **width** (ancho) y **height** (alto), cuyos valores se especifican en píxeles (absoluto) o en porcentaje (relativo al tamaño que lo contiene).

La propiedad **max-width** se utiliza para establecer la anchura máxima de un elemento. El ancho máximo se puede especificar en los valores de longitud, como píxeles, cm, etc., o en porcentaje ( % ) del bloque de contención, o *default* ( esto es por defecto ). Significa que no hay ancho máximo ). De igual forma, existen los valores **max-height**, **min-width** y **min-height**.

El siguiente ejemplo muestra un elemento <div> con un altura de 100 píxeles y un ancho máximo de 500 píxeles.

```
div {  
  max-width: 500px;  
  height: 100px;  
  background-color: blue;  
}
```



Imagen ejemplo de tamaño de creación propia

Las propiedades del borde CSS (***border***) permiten especificar el estilo , el ancho y el color del borde de un elemento. A su vez, existen diferentes propiedades para especificar el borde a un elemento, las más importantes se describen a continuación.

- ***border-style***: esta propiedad especifica el tipo de borde que se visualiza , siendo las opciones más usuales **solid** (línea continua), **double** (línea continua doble), **dashed** (línea discontinua) y **dotted** (línea con puntitos).
- ***border-width***: permite añadir el grosor a cada uno de los lados del borde. Se puede establecer de forma independiente (**top border**, **right border**, **bottom border**, y **left border**).
- ***border-color***: esta propiedad asigna un color al borde, de los colores que hemos estudiado hasta ahora. También permite especificar el color individual de un lado.
- ***border***: es una propiedad genérica que define las tres propiedades anteriores a la vez siendo necesario especificar en este orden anchura (**width**), estilo que es obligatorio (**style**) y color (**color**). Por ejemplo:

```
p {
  border: 5px solid red;
}
```

- ***border-radius***: establece el valor de curvatura de las esquinas, normalmente especificado en **px**.

```
p.normal {
  border: 2px solid green;
}

p.radio1 {
  border: 2px solid green;
  border-radius: 5px;
}

p.radio2 {
  border: 2px solid green;
  border-radius: 12px;
}
```

Imagen ejemplo CSS (border-radius) de creación propia

Normal border

Round border

Rounder border

Imagen ejemplo en navegador (border-radius) de creación propia

## 4.3 Márgenes y relleno



En CSS es necesario establecer un espacio de separación entre un elemento y los elementos que lo rodean, o entre el borde del elemento y su propio contenido. Existen dos propiedades que permiten establecer estas configuraciones.

### **Margen (margin)**

Las propiedades de margen CSS se utilizan para generar el espacio alrededor de los elementos, define el tamaño del espacio en blanco fuera del borde. CSS tiene un control total sobre los márgenes de cada lado de un elemento (**margin-top**, **margin-right**, **margin-bottom** y **margin-left**).

Los valores que puede tomar son los siguientes:

- **auto** : el navegador calcula el margen.
- **length**: especifica un margen en px, pt, cm, etc.
- **%** : establece un margen en % de la anchura del elemento que contiene.
- **inherit** : especifica que el margen debe ser heredado de elemento padre.

Si usamos porcentajes en lugar de una medida exacta, obtendremos un resultado relativo al tamaño de la ventana del navegador. Según la ocasión, nos convendrá emplear un tipo u otro. El siguiente ejemplo especifica todos los márgenes del párrafo.

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

¿Podríamos establecer todos los márgenes de una sola vez? Existe la posibilidad de especificar el margen de todos los lados con una única propiedad (de cuatro a uno valor) que sigue siempre el mismo orden (dirección de las agujas de reloj desde top). Veamos un ejemplo en el que top margin es 25px, right margin es 50px, bottom margin es 75px y left margin es 100px.

```
margin: 25px 50px 75px 100px;
```

### **Relleno (padding)**

La propiedad se utiliza para generar el espacio alrededor del contenido. padding borra un área alrededor del contenido (en el interior del borde) de un elemento. Acepta los mismos valores especificados para el margen.

```
div {  
  border: 1px solid black;  
  background-color: lightgreen;  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```

Imagen ejemplo CSS (border-radius) de creación propia

## Usando propiedades individuales

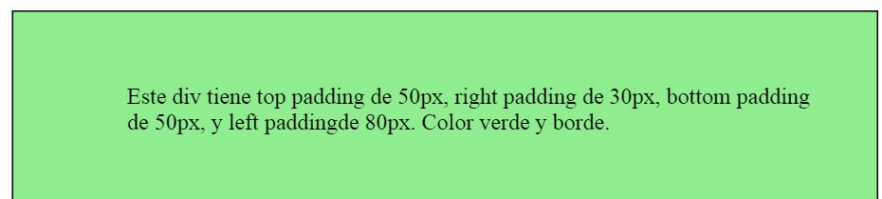


Imagen ejemplo en navegador (border-radius) de creación propia

## 4.4 Posicionamiento



El posicionamiento de los elementos en una página web se establecen a partir de la estructura HTML elegida, que el navegador interpreta organizando los elementos unos respecto a otros (dentro, adyacentes, debajo etc).

La propiedad **position** se emplea para definir el tipo de posicionamiento de un objeto en la pantalla según diferentes valores. Además se acompaña de las propiedades **left** (izquierda) y **top** (arriba) para especificar la distancia que tendrá el objeto respecto al borde izquierdo y al superior. Existen cuatro valores posibles:

- **position: static.** Es el valor por defecto no está colocado de manera especial , siempre se coloca de acuerdo con el flujo normal de la página. Desaparece en CSS3.
- **position: relative.** Es el tipo de posicionamiento por defecto( sin no existe static como ocurre en CSS3, en CSS será static). Es la posición normal. La configuración de las propiedades " top, right, bottom, y left " de un elemento posicionado relativamente hará que se puede ajustar fuera de su posición normal.
- **position: fixed.** Se posiciona en relación con el área de visualización, lo que significa que siempre permanece en el mismo lugar, incluso si la página se desplaza. El elemento siempre será visibles incluso haciendo scroll.
- **position: absolute.** Se posiciona con respecto al antecesor posicionado más cercano ( en lugar de posicionado con respecto a la ventana gráfica , como fixed) . Por defecto, se posiciona en la esquina izquierda superior.

A continuación tienes un ejemplo de posicionamiento con <div>, pulsa el el [siguiente enlace](#) para que puedas probarlo de forma sencilla por ti mismo:

```
div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}  
  
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

Imagen ejemplo CSS (position) de creación propia

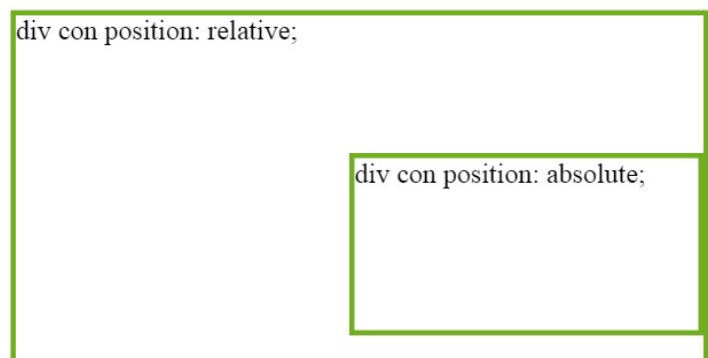


Imagen ejemplo en navegador (position) de creación propia

## 4.5 Formulario



El formulario es un elemento que aparece frecuentemente en cualquier página web. En el tema pasado estudiamos los campos HTML que podía formar parte de un formulario básico. En este punto aprenderemos a añadir un estilo profesional al formulario. Encontrarás más ejemplos en la web [w3school](http://w3school.com).

¡Veamos un ejemplo! En la izquierda aparece el código HTML de un formulario con tres campos de entrada y un botón de envío. Este formulario no tiene estilos asociados y si lo ejecutas en un navegador obtendrás el resultado que se aprecia en la imagen de la derecha. *¿Cómo podemos mejorar su apariencia?*

```
<!DOCTYPE html>
<html>
<body>
  <h3>Usado CSS para dar estilo al formulario HTML</h3>
  <div>
    <form action="pageaction.php">
      <label for="fname">Nombre</label>
      <input type="text" id="fname" name="firstname">

      <label for="lname">Apellidos</label>
      <input type="text" id="lname" name="lastname">

      <label for="country">País</label>
      <select id="country" name="country">
        <option value="australia">España</option>
        <option value="canada">Francia</option>
        <option value="usa">Portugal</option>
      </select>
      <input type="submit" value="Enviar">
    </form>
  </div>
</body>
</html>
```

Imagen código HTML formulario sin estilo de creación propia

### Usado CSS para dar estilo al formulario HTML

Nombre

Apellidos

País España ▼

Imagen de ejecución en le navegador del código HTML sin estilo de creación propia

En primer lugar debemos asociar el código HTML a una hoja de estilos (CSS) en ella definiremos las propiedades que ves en la imagen de la izquierda. Todas las propiedades las hemos estudiado en este tema, la principal novedad es como se indica en CSS a qué tipo de campo (input) nos estamos refiriendo.

```

input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

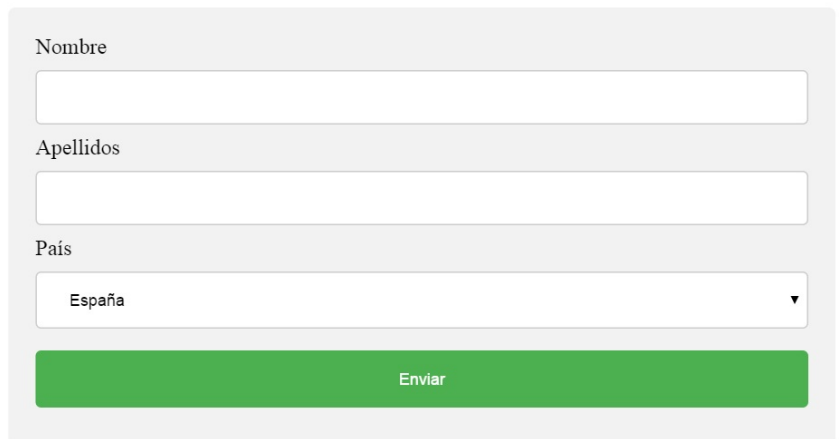
input[type=submit]:hover {
  background-color: #45a049;
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}

```

Imagen código CSS (formulario) de creación propia

## Usado CSS para dar estilo al formulario HTML



Nombre

Apellidos

País

España ▼

Enviar

Imagen de ejecución en el navegador del código HTML vinculado a la hoja de estilo de creación propia

### ¿Qué es el diseño adaptativo?

El diseño adaptable , **Responsive Web Design** en inglés, hace que una página web se visualice bien en todos los dispositivos sólo utiliza HTML y CSS ,no es un programa o un JavaScript .

Las páginas web son accesibles desde diferentes dispositivos : ordenadores de sobremesa , tabletas y móviles. Una página web debe tener una correcta apariencia , y ser fácil de usar, independientemente del dispositivo . Las páginas web no deben omitir información para adaptarse a dispositivos más pequeños , sino más bien adaptar su contenido a cada dispositivo según su resolución.

Hoy en día, existen distintos frameworks (marcos de referencia) que definen plantillas que establecen un diseño que se adapta a los dispositivos a través de reglas CSS concretas. En otras palabras, nos ayudan con el diseño de la página web.

Un ejemplo sería el tratamiento de imágenes según el tamaño de la pantalla donde se visualizará. Imagina que incluyes una imagen en tu página *¿cómo logramos que se vea correctamente en una pantalla de ordenador de gran dimensión y a la vez en un móvil de resolución reducida?*

Un solución es disponer de dos fotografías iguales pero con diferente tamaño y crear una regla CSS para que escoja la imagen correcta según el dispositivo.

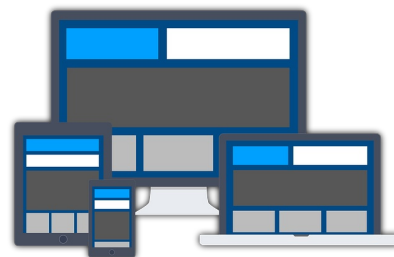


Imagen en Pixabay de Clovit bajo licencia Creative Common CCO

```
/* Para dispositivos más pequeños que 400px: */
body {
    background-repeat: no-repeat;
    background-image: url('img_smallflower.jpg');
}

/* Para dispositivos de 400px en adelante: */
@media only screen and (min-device-width: 400px) {
    body {
        background-image: url('img_flowers.jpg');
    }
}
```

Imagen ejemplo imagen responsive de creación propia

### Para saber más

#### Bootstrap

Bootstrap es un potente framework de CSS que fue creado por Twitter para simplificar el proceso de maquetación web responsive. Sin tener grandes conocimientos en CSS, con esta herramienta podéis comenzar a maquetar un sitio web adaptable a todo tipo de dispositivos.

Se basa en la simplicidad de sus interfaces, lo cual es una tendencia del mercado, en las que tiende a diseño plano, botones grandes, etc para facilitar la **usabilidad** en los dispositivos más pequeño, atendiendo a la **Experiencia del usuario**.

Podemos descargar múltiples plantillas gratuitas de páginas como <https://startbootstrap.com/template-categories/all/> . La siguiente imagen muestra la interfaz gráfica de una página web realizada a partir de una **plantilla bootstrap original**.





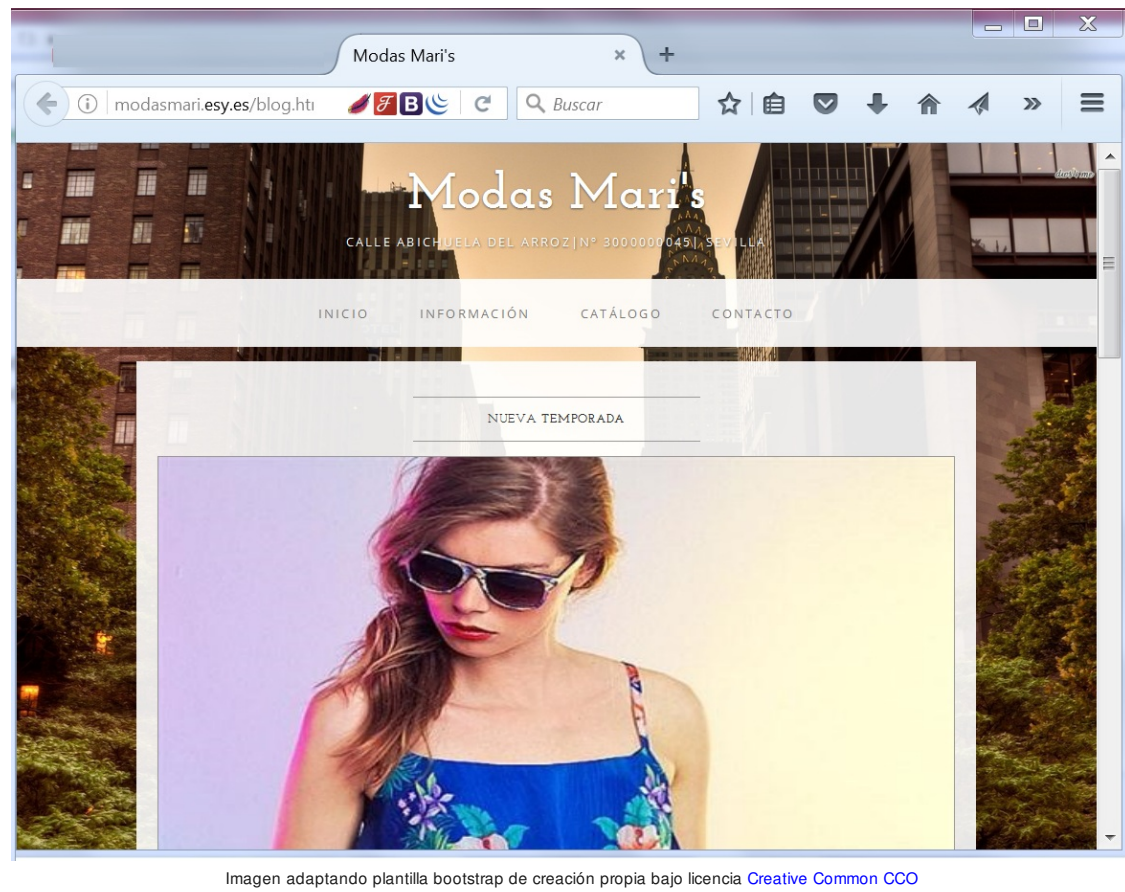


Imagen adaptando plantilla bootstrap de creación propia bajo licencia [Creative Common CCO](#)

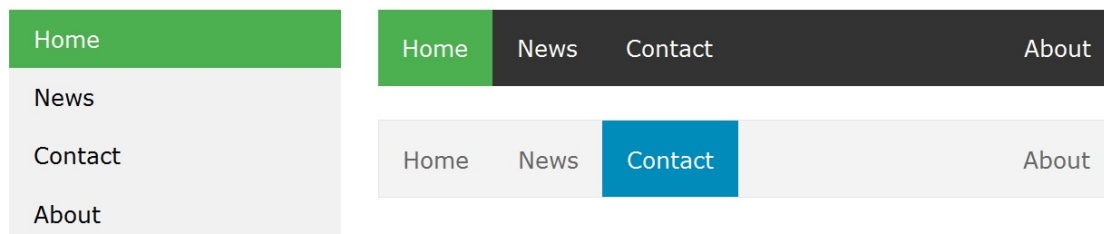
## 6. Para ampliar



Tal y como vimos en el tema de HTML, existen múltiples páginas y plataformas que nos ayudan con el aprendizaje, en este caso de las hojas de estilo (CSS). Puedes consultar más información al respecto en [css w3schools](#). Entre las propiedades de las hojas de estilos destacan los siguientes elementos:

### **Navigation bars**

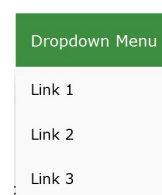
La barra de navegación es una lista de enlaces. Actualmente, tener una navegación fácil de usar es importante para cualquier sitio web . Con CSS se puede transformar menús HTML básicos en barras de buen aspecto de navegación.



Imágenes de ejemplo de navigation bars de creación propia

### **Dropdowns**

Esta propiedad permite crear un cuadro desplegable que aparece cuando el usuario mueve el puntero del ratón sobre un elemento.



Imágenes de ejemplo dropdowns de creación propia

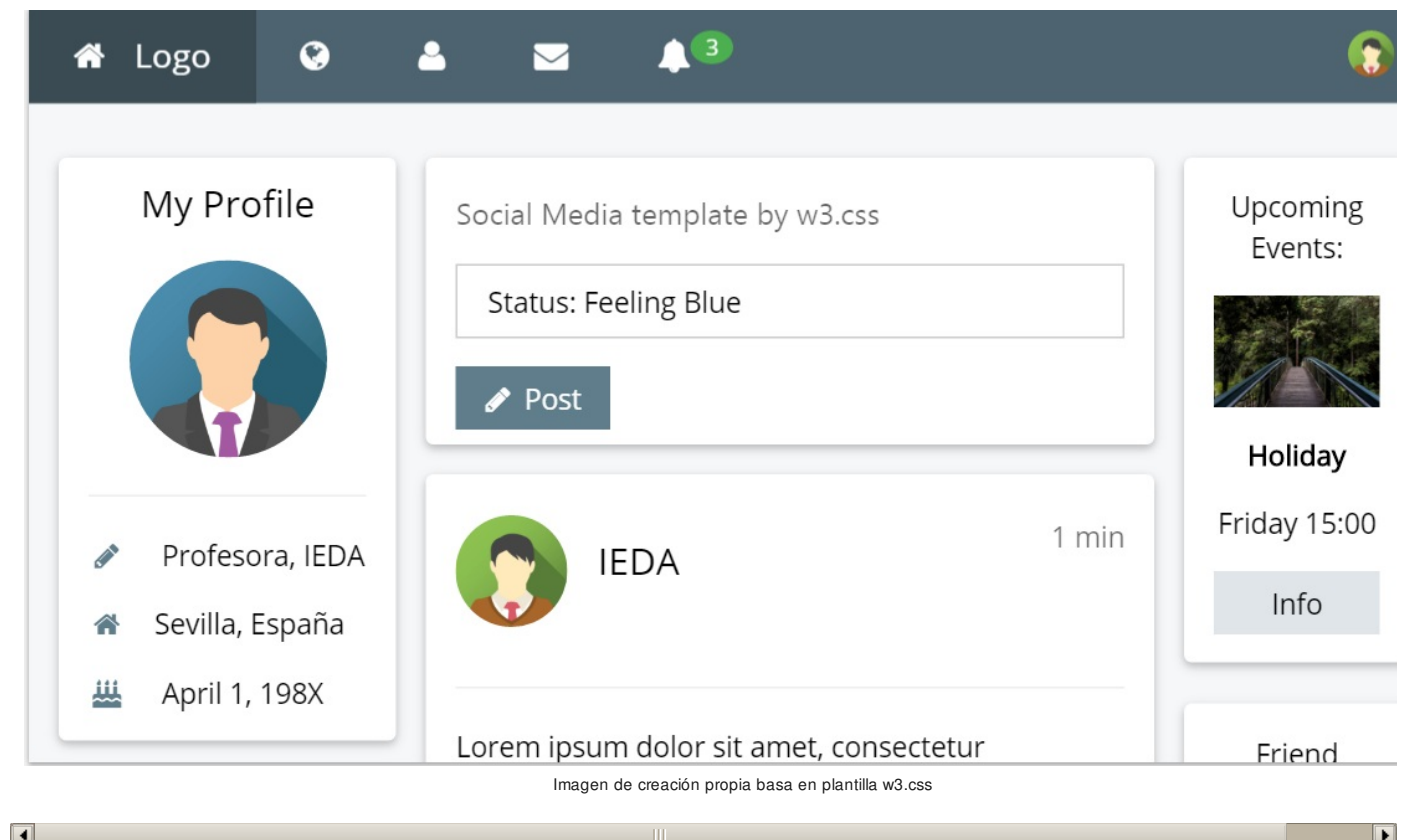
Al igual que ocurría con HTML, existen múltiples y variadas páginas así como tutoriales o vídeo tutoriales para aprender hojas de estilo. En concreto para CSS3 existe dos páginas que será de gran ayuda [w3schools](#) y [html5&css3](#).

#### ¿Pero qué es CSS3?

CSS3 es el último estándar de CSS, es compatible con versiones completamente con versiones anteriores de CSS. La mayor parte de las nuevas propiedades de CSS3 se implementan en los navegadores modernos. Algunos de los módulos CSS3 más importantes son:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

Existen múltiples plantillas responsive libres de modificar, guardar, compartir y utilizar. Un ejemplo de plantilla la puedes encontrar en este [enlace](#).



## Comprueba lo aprendido

¿Qué entiendes por CSS?

☐ Computer Style Sheets

☐ Colorful Style Sheets

☐ Cascading Style Sheets

### Solution

1. [Incorrecto](#)
2. [Incorrecto](#)
3. [Correcto](#)

¿Cuál es el código HTML correcto para referirse a una hoja de estilos externa ?

☐ <style src="mystyle.css">

☐ <stylesheet>mystyle.css</stylesheet>

☐ </stylesheet>mystyle.css</stylesheet>

☐ <link rel="stylesheet" type="text/css" href="mystyle.css">

### Solution

1. [Incorrecto](#)
2. [Incorrecto](#)
3. [Incorrecto](#)
4. [Correcto](#)

¿Cuál es la sintaxis correcta?

☐ {body:color=black;}

☐ body {color: black;}

☐ {body:color=black;}

### Solution

1. [Incorrecto](#)
2. [Correcto](#)
3. [Incorrecto](#)

¿Qué propiedad CSS se utiliza para cambiar el color del texto de un elemento?

☐ text-color

☐ color

☐ fgcolor

### Solution

1. [Incorrecto](#)
2. [Correcto](#)

3. [Incorrecto](#)

¿Cuál es el valor predeterminado de la propiedad de posición ?

☐ absolute

☐ relative

☐ fixed

**Solution**

1. [Correcto](#)
2. [Incorrecto](#)
3. [Correcto](#)



## Aviso legal

El presente texto (en adelante, el "**Aviso Legal**") regula el acceso y el uso de los contenidos desde los que se enlaza. La utilización de estos contenidos atribuye la condición de usuario del mismo (en adelante, el "**Usuario**") e implica la aceptación plena y sin reservas de todas y cada una de las disposiciones incluidas en este Aviso Legal publicado en el momento de acceso al sitio web. Tal y como se explica más adelante, la autoría de estos materiales corresponde a un trabajo de la **Comunidad Autónoma Andaluza, Consejería de Educación, Cultura y Deporte (en adelante Consejería de Educación, Cultura y Deporte Andaluza )**.

Con el fin de mejorar las prestaciones de los contenidos ofrecidos, la Consejería de Educación, Cultura y Deporte Andaluza se reservan el derecho, en cualquier momento, de forma unilateral y sin previa notificación al usuario, a modificar, ampliar o suspender temporalmente la presentación, configuración, especificaciones técnicas y servicios del sitio web que da soporte a los contenidos educativos objeto del presente Aviso Legal. En consecuencia, se recomienda al Usuario que lea atentamente el presente Aviso Legal en el momento que acceda al referido sitio web, ya que dicho Aviso puede ser modificado en cualquier momento, de conformidad con lo expuesto anteriormente.

### **1. Régimen de Propiedad Intelectual e Industrial sobre los contenidos del sitio web**

#### **1.1. Imagen corporativa**

Todas las marcas, logotipos o signos distintivos de cualquier clase, relacionados con la imagen corporativa de la Consejería de Educación, Cultura y Deporte Andaluza que ofrece el contenido, son propiedad de la misma y se distribuyen de forma particular según las especificaciones propias establecidas por la normativa existente al efecto.

#### **1.2. Contenidos de producción propia**





