

# Historia de CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.

El organismo W3C (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de Mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como "CSS nivel 2". La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2 que aún se está elaborando (la última actualización es del 8 de septiembre de 2009). Al mismo tiempo, la siguiente recomendación de CSS, conocida como "CSS nivel 3", continúa en desarrollo desde 1998 y hasta el momento sólo se han publicado borradores.

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. El mismo año que se publicó CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000.

# Estructura

- Sintaxis
- Comentarios

## Sintaxis

La sintaxis de CSS, es bastante sencilla. Tiene tres elementos importantes. Selectores, propiedades y valores.

Cada selector, “selecciona” cual es el elemento/etiqueta del HTML al que se le va a estar aplicando el diseño. La propiedad es el aspecto de ese elector, como por ejemplo, el color, el color de fondo, los márgenes, tamaños de tipografía, etc.

Los valores, son esos que aplicamos a cada una de estas propiedades.

Un archivo CSS, está compuesto por muchos selectores y cada selector puede tener más de una propiedad.

Ejemplo:



El modo de escribir CSS es el siguiente:

Primero seleccionamos la etiqueta a la cual le queremos aplicar el diseño, por ejemplo p, después abrimos una llave y debajo escribimos la propiedad que le queremos aplicar, en nuestro caso sería `font-size`, justo después de la propiedad, escribimos : (dos puntos) y el valor, `14px`. Al terminar de escribir el valor, se agrega un ; (punto y coma) y finalmente se cierra con la llave.

A medida que empezás a agregar más y más pares propiedad-valor para cada selector CSS es importante que te acuerdes de poner punto y coma (;) al final de cada línea.

El punto y coma le indica a las CSS que es el fin de un par propiedad-valor y que es el momento de continuar con el siguiente.

## Comentarios

Al igual que vimos en HTML, CSS también permite incluir comentarios entre sus reglas y estilos. Los navegadores

ignoran por completo cualquier comentario de los archivos CSS, por lo que es común utilizarlos para estructurar de forma clara nuestros archivos.

El comienzo de un comentario se indica mediante los caracteres `/*` y el final del comentario se indica mediante `*/`, tal y como se muestra en el siguiente ejemplo:

```
/* Este es un comentario en CSS */
```

Es importante ver, que la sintaxis de los comentarios en CSS, es muy diferente a los del HTML.

# Selectores básicos

## Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p { ... }
```

Se pueden aplicar los mismos estilos a dos etiquetas diferentes, encadenando los selectores. En el siguiente ejemplo, los títulos de sección `h1` y `h2` comparten los mismos estilos:

CSS

Result

CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Se pueden incluir todos los selectores separados por una coma (,) y el resultado de la siguiente regla CSS es equivalente a las dos reglas anteriores:

CSS

Result

## Selector descendiente

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

El selector del siguiente ejemplo selecciona todos los elementos de la página que se encuentren dentro de un elemento `<p>`:

```
p span { color: red; }
```

Si el código HTML de la página es el siguiente:

```
<p>
...
<span>texto1</span>
...
<a href="">...<span>texto2</span></a>
...
</p>
```

Este estilo, solo se le va a aplicar a todos los elementos `<span>` que estén dentro de un elemento `<p>`, pero si en otra parte del sitio tenemos un elemento que no esté dentro de un elemento `<p>`, entonces este estilo no se le aplica.

## Selector de ID

A veces, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo `id`. Este tipo de selectores sólo seleccionan un elemento de la página ya que el valor del atributo `id` no se puede repetir en dos elementos diferentes dentro una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo del numeral (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#destacado { color: red; }
```

```
<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

En el ejemplo anterior, el selector `#destacado` solamente selecciona el segundo párrafo (cuyo atributo `id` es igual a `destacado`).

## Herencia

Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

```
<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
  body { color: blue; }
</style>

</head>
```

El selector `body` solamente establece el color de la letra para el elemento `<body>`. No obstante, la propiedad `color` es una de las que se heredan de forma automática, por lo que todos los elementos descendientes de `<body>` muestran ese mismo color de letra. Entonces, establecer el color de la letra en el elemento `<body>` de la página implica cambiar el color de letra de todos los elementos de la página.

# Valores y unidades

- Unidades de medida
- Unidades absolutas
- Unidades relativas
- Colores

## Unidades de medida

Las medidas en CSS se emplean, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto.

Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida.

CSS divide las unidades de medida en dos grupos: absolutas y relativas. Las medidas relativas definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación con el valor indicado. Las unidades absolutas establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

## Unidades absolutas

Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia.

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. El motivo de porque las unidades absolutas se usan poco es que no permiten adaptarse al tamaño de la pantalla que utilice cada usuario.

## Unidades relativas

Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor.

Las tres unidades de medida relativas más utilizadas son:

- `em` relativa respecto del tamaño de letra del elemento.
- `ex` relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.
- `px` (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

La unidad `em` hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, `1em` equivale a 12 puntos. El valor de `1ex` se puede aproximar por `0.5em`.

Ejemplo: `p { margin: 1em; }`

La regla CSS anterior indica que los párrafos deben mostrar un margen de anchura igual a `1em`.

El píxel es una unidad de medida un tanto especial. La consideramos una unidad relativa porque no tiene un valor absoluto y único, sino que presenta variaciones según los dispositivos.

Las pantallas suelen tener una resolución expresada en píxeles, por ejemplo 1280 x 800 px, donde el primer valor indica el número de puntos horizontales y el segundo valor el número de puntos verticales en que se divide la pantalla.

Unidad	Símbolo	Ejemplo
Porcentaje	%	#menu1 {width: 50%;}
Relativa al tamaño de letra	em	#menu1 {font-size: 2.65em;}
Relativo a la x minúscula	ex	#menu1 {font-size: 2.65ex;}
Pixel*	px	#menu1 {font-size: 24px;}

## Colores

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, RGB hexadecimal, RGB numérico y RGB porcentual. Aunque el método más habitual es el del RGB hexadecimal.

### Palabras clave

CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

(Falta Imagen)

### RGB decimal

El modelo RGB consiste en definir un color indicando la cantidad de color rojo, verde y azul que se debe mezclar para obtener ese color. La combinación de las letras de estos tres colores en inglés es RGB y así se denomina al sistema de construcción de colores basado en indicar la proporción de cada uno de estos tres colores.

Por lo tanto, en el modelo RGB un color se define indicando sus tres componentes R (rojo), G (verde) y B (azul). Cada una de las componentes puede tomar un valor entre cero y un valor máximo. La cantidad de cada color se expresa entre un valor mínimo que es cero y un valor máximo que es 255.

El color rgb (0, 0, 0) es el negro y el color rgb (255, 255, 255) es el blanco. El color rojo será el rgb (255, 0, 0), el verde rgb (0, 255, 0) y el azul (0, 0, 255).

### RGB porcentual

Las componentes RGB de un color también se pueden indicar mediante un porcentaje. El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal. La única diferencia es que en este caso el valor de las componentes RGB puede tomar valores entre 0% y 100%. Por tanto, para transformar un valor RGB decimal en un valor RGB porcentual, es preciso realizar una regla de tres considerando que 0 es igual a 0% y 255 es igual a 100%.

### RGB hexadecimal

Para entender el modelo RGB hexadecimal, es preciso introducir un concepto matemático llamado sistema numérico hexadecimal. Cuando realizamos operaciones matemáticas, siempre utilizamos 10 símbolos para representar los números (del 0 al 9). Por este motivo, se dice que utilizamos un sistema numérico decimal. Entre los sistemas numéricos alternativos más utilizados se encuentra el sistema hexadecimal, que utiliza 16 símbolos para representar sus números.

Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números. De esta forma, en el sistema hexadecimal, después del 9 no va el 10, sino la A. La letra B

equivale al número 11, la C al 12, la D al 13, la E al 14 y la F al número 15.

Para definir un color en CSS con el método RGB hexadecimal se deben realizar los siguientes pasos: - Determinar las componentes RGB decimales del color original, por ejemplo: R = 47, G = 62, B = B0 - Transformar el valor decimal de cada componente al sistema numérico hexadecimal.

Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0 - Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior es #4762B0 en formato RGB hexadecimal.

Ejemplo: `p { color: #4762B0; }`

Afortunadamente, todos los programas de diseño gráfico convierten de forma automática los valores RGB decimales a sus valores RGB hexadecimales, por lo que no tienes que hacer ninguna operación matemática:

(Falta Imagen)

Herramienta de color de Photoshop para definir los colores según los modelos RGB, CMYK, Lab, HSB y RGB hexadecimal

Se admiten algunas variantes a la hora de definir un color hexadecimal en CSS. Un color se puede designar con solo tres símbolos en lugar de seis. En este caso se considera que es una abreviación de la repetición de esos símbolos, por ejemplo #FFF se considera abreviación de #FFFFFF y #05F se considera abreviación de #0055FF.

También se admite el uso de letras en minúsculas en lugar de letras mayúsculas. De este modo, #ffffff equivale a #FFFFFF ó #0dab4f equivale a #0DAB4F.



# Textos, Fuentes y Listas

## Estilos

### Itálica

La propiedad que nos permite elegir si queremos mostrar nuestro texto en itálica, se llama `font-style`. Funciona de la misma manera que la etiqueta que vimos en HTML.

Se escribe de la siguiente manera:

```
p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; }
```

### Negrita

Si queremos mostrar nuestro texto en negrita, la propiedad que vamos a utilizar, se llama `font-weight`. La palabra `weight` en inglés, significa peso. En este caso, el valor para que nuestro texto se vea en negrita, se llama **Bold** y funciona igual que la etiqueta `<strong>` del HTML.

Al ejemplo anterior, vamos a agregarle un titular `h1` que queremos que siempre se muestre en negrita.

```
h1 { font-weight: bold; }  
p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; }
```

## Tipografías

CSS dispone de una serie de propiedades que permiten controlar la forma en que se muestra el texto.

Entre ellas vamos a ver algunas como el tamaño, el color, subrayado, interlineado, alineación, etc.

La primera que vamos a ver es la familia tipográfica que queremos utilizar para mostrar el texto, esta propiedad se llama `font-family`.

Ejemplo:

```
p { font-family: Arial; }
```

Por el momento, solo vamos a poder elegir algunas tipografías básicas, que son las que el usuario que va a estar navegando mi web, tenga instaladas en su computadora. Por ejemplo, Helvetica, Arial, Times new roman y Verdana. Como no sabemos exactamente cuáles tipografías tiene instaladas el usuario, podemos elegir más de una de la siguiente manera:

Ejemplo:

```
p { font-family: Arial, Helvetica; }
```

De esta manera, el navegador probará en primer lugar con el primer tipo de letra indicado. Si el usuario la tiene instalada, el texto se muestra con ese tipo de letra. Si el usuario no dispone del primer tipo de letra indicado, el navegador irá probando con el resto de tipos de letra hasta que encuentre alguna fuente que esté instalada en el ordenador del usuario.

Para solucionar este problema se utilizan las familias tipográficas genéricas. Cuando la propiedad `font-family` toma un `sans-serif`

valor igual a `font-family: Arial, Helvetica, sans-serif;`, el diseñador no indica al navegador que debe utilizar la fuente Arial, sino que debe utilizar "la fuente que más se parezca a Arial de todas las que tiene instaladas el usuario".

Ejemplo:

```
p { font-family: Arial, Helvetica, sans-serif; }
```

## Tamaño

La propiedad para definir el tamaño se llama `font-size`.

Este tamaño lo podemos definir con las unidades de medida relativas y absolutas que vimos anteriormente.

Continuando con el ejemplo anterior, el modo de configurar el tamaño de nuestro párrafo es el siguiente:

Ejemplo:

```
p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; }
```

De esta manera, le estamos diciendo que nuestro párrafo va a tener tipografía Arial y un tamaño de 14px.

## Color

También podemos aplicarle color a nuestro borde. La propiedad que utilizamos es `border-color`. Funciona exactamente de la misma manera que `border-width`, con la diferencia que para aplicarle un color debemos utilizar el código hexadecimal que vimos anteriormente.

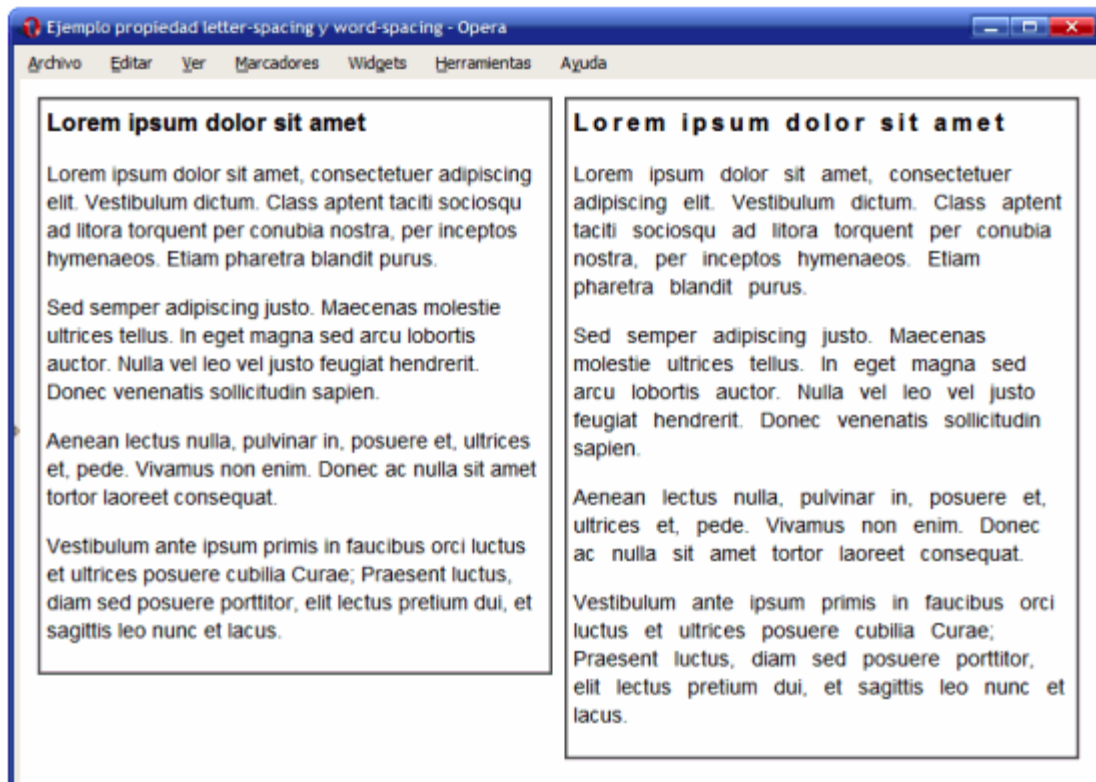
Ejemplo:

```
div { border-top-color: #CC0000; border-right-color: #CCC; border-bottom-color: #00FF00; border-left-color: #CCC; }
```

## Espacio entre letras y palabras

CSS también permite controlar la separación entre las letras que forman las palabras y la separación entre las palabras que forman los textos. La propiedad que controla la separación entre letras se llama `letter-spacing` y la separación entre palabras se controla mediante `word-spacing`.

La siguiente imagen muestra la comparación entre un texto normal y otro con las propiedades `letter-spacing` y `word-spacing` aplicadas:



Esto se elige al igual que con font-size, con una unidad de medida. Como por ejemplo el pixel (px).

Ejemplo:

```
h1 { font-weight: bold; color: #cccccc; }

p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; letter-spacing: 5px; word-spacing: 10px; }
```

## Espacio entre renglones

Siguiente en la misma línea, también podemos modificar el espaciado entre renglón y renglón (interlineado). La propiedad en este caso, se llama `line-height` y quiere decir, la altura de la de línea.

Algo importante a tener en cuenta, es que a la hora de elegir el tamaño de interlineado, tenemos que tener en cuenta el tamaño de tipografía que estamos utilizando, por ejemplo para una tipografía de 14px, un interlineado de 18px es apropiado.

```
h1 { font-weight: bold; color: #cccccc; }

p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; letter-spacing: 5px; word-spacing: 10px; line-height: 18px; }
```

## Alineación del texto

Una de las propiedades más importantes con respecto al texto, es su alineación. Podemos elegir si queremos que el texto se vea alineado a la izquierda, a la derecha, centrado o justificado.

La propiedad que define la alineación del texto se denomina `text-align`.

Los valores que utiliza CSS para definir esto, son los siguientes: la izquierda (`left`), a la derecha (`right`), centrado (`center`) y justificado (`justify`).

La siguiente imagen muestra el efecto de establecer el valor `left`, `right`, `center` y `justify` respectivamente a cada uno de los párrafos de la página.

Ejemplo:

```
h1 { font-weight: bold; color: #cccccc; text-align: center; }  
  
p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; letter-spacing: 5px; word-spacing: 10px; line-height: 18px; text-align: left; }
```

En este caso, decidimos que todos nuestro títulos H1 estén alineados al centro y todos nuestros párrafos hacia la izquierda.

## Decoración y transformación de texto

### Decoración del texto

Esta propiedad, nos permite decorar el texto, la misma se llama `text-decoration`.

Cuando hablamos de decorar el texto, nuestras opciones son: (subrayado, tachado, parpadeante, etc.)

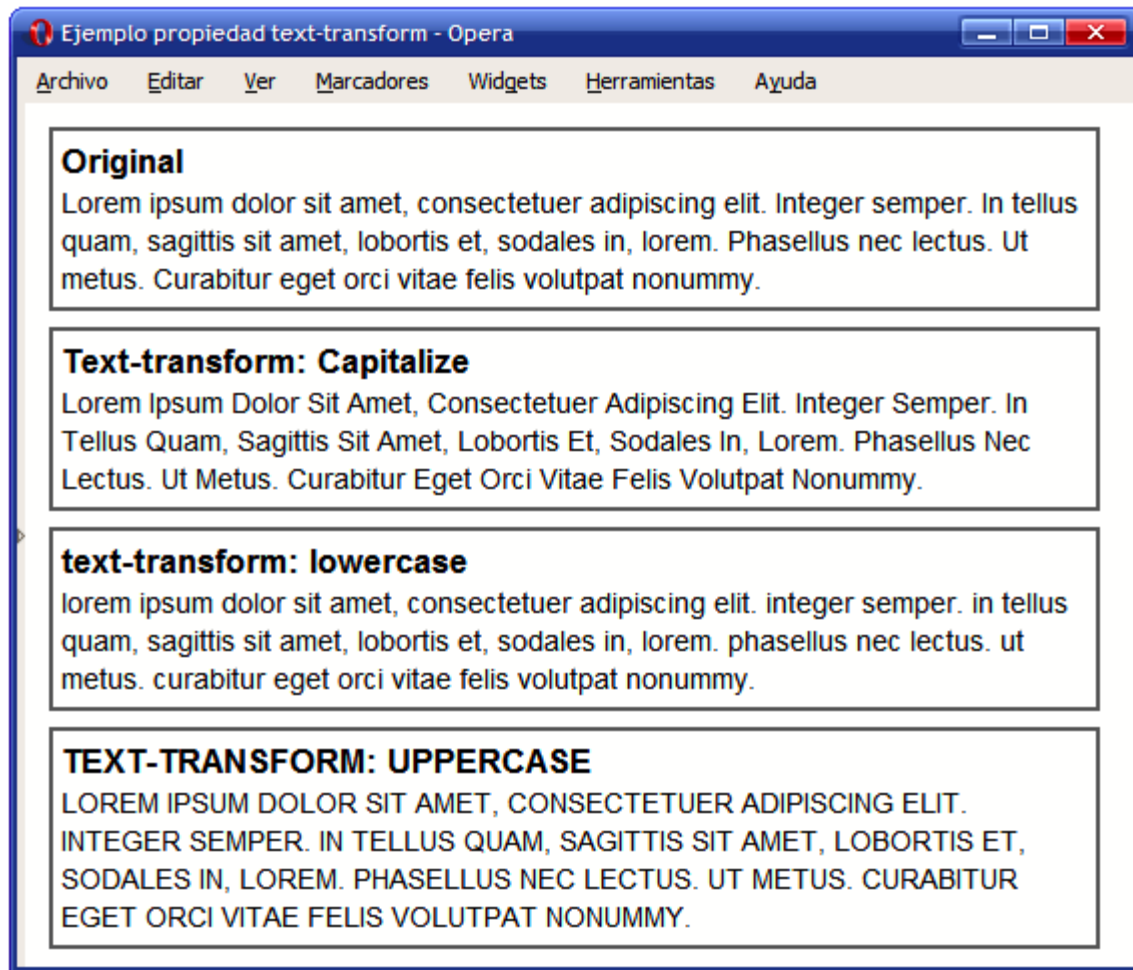
El valor `underline` subraya el texto, hay que tener cuidado donde lo queremos agregar ya que puede confundir a los usuarios haciéndoles creer que se trata de un enlace. El valor `overline` añade una línea en la parte superior del texto, un aspecto que raramente utilizado. El valor `line-through` muestra el texto tachado con una línea continua, por lo que su uso tampoco es muy habitual.

### Transformación del texto

Una de las propiedades de CSS más desconocidas y que puede ser de gran utilidad en algunas circunstancias es la propiedad `text-transform`, que puede variar de forma sustancial el aspecto del texto.

La propiedad `text-transform` permite mostrar el texto original transformado en un texto completamente en mayúsculas (`uppercase`), en minúsculas (`lowercase`) o con la primera letra de cada palabra en mayúscula (`capitalize`).

La siguiente imagen muestra cada uno de los posibles valores.



En nuestro ejemplo, vamos a elegir que todos los títulos sean mostrados en mayúscula.

Ejemplo:

```
h1 { font-weight: bold; color: #cccccc; text-align: center; text-transform: uppercase; }  
p { font-family: Arial, Helvetica, sans-serif; font-size: 14px; font-style: italic; letter-spacing:  
    5px; word-spacing: 10px; line-height: 18px; text-align: left; }
```

## Listas

Por defecto, los navegadores muestran los elementos de las listas no ordenadas con una viñeta formada por un pequeño círculo de color negro.

CSS define varias propiedades para controlar el tipo de viñeta que muestran las listas.

La propiedad básica es la que controla el tipo de viñeta que se muestra y se denomina `list-style-type`.

En primer lugar, el valor `none` permite mostrar una lista en la que sus elementos no contienen viñetas, números o letras. Se trata de un valor muy utilizado, ya que es imprescindible para los menús de navegación creados con listas.

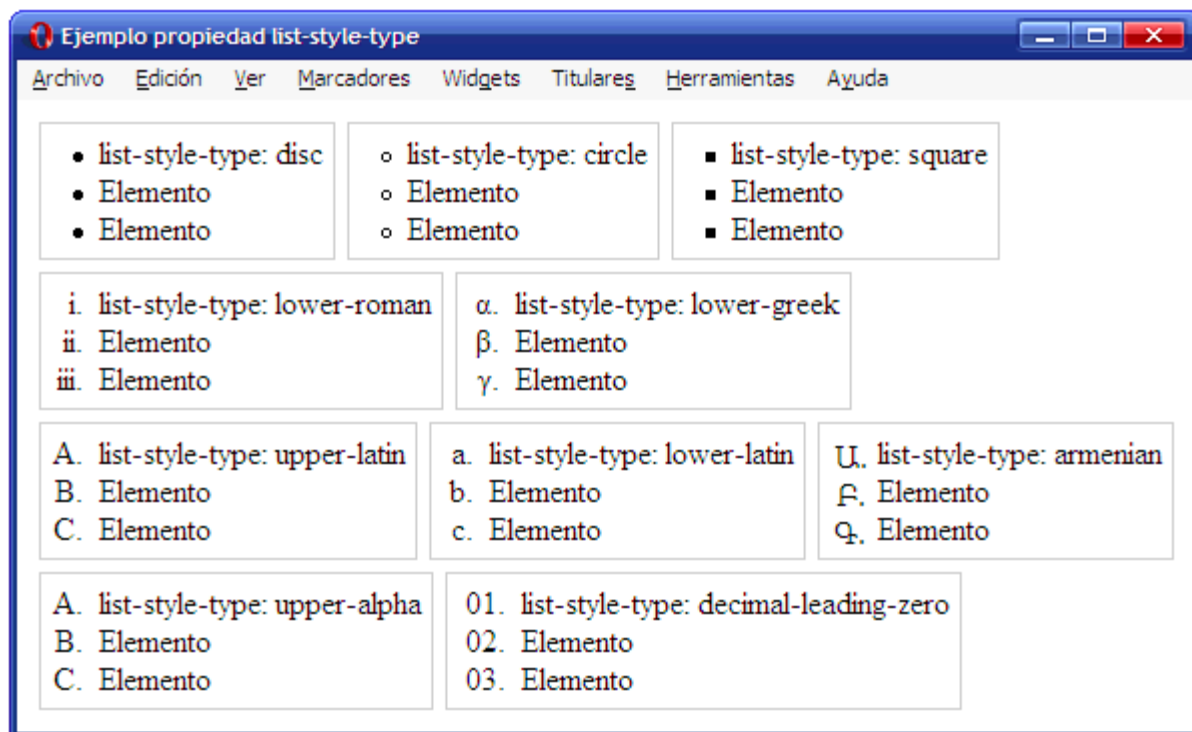
El resto de valores de la propiedad `list-style-type` se dividen en tres tipos: gráficos, numéricos y alfabéticos.

- Los valores gráficos son `disc`, `circle` y `square` y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.
- Los valores numéricos están formados por `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`,

armenian y georgian.

- Por último, los valores alfanuméricos se controlan mediante lower-latin, lower-alpha, upper-latin, upper-alpha y lower-greek.

La siguiente imagen muestra algunos de los valores definidos por la propiedad list-style-type:

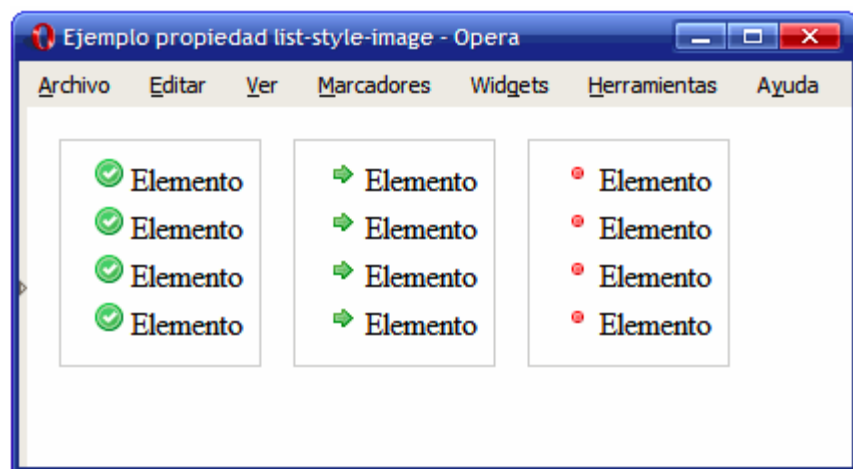


El siguiente ejemplo indica que no se debe mostrar ni viñetas automáticas ni viñetas personalizadas:

```
ul { list-style: none }
```

Cuando queremos personalizar el aspecto de las viñetas, se debe emplear la propiedad list-style-image, esta permite mostrar una imagen propia en vez de una viñeta automática.

Las imágenes personalizadas se indican mediante la URL de la imagen.



Las reglas CSS correspondientes al ejemplo anterior se muestran a continuación:

```
ul.ok { list-style-image: url("imagenes/ok.png"); }
```

```
ul.flecha { list-style-image: url("imagenes/flecha.png"); }  
ul.circulo { list-style-image: url("imagenes/circulo_rojo.png"); }
```

# Tipografía Web

- Tipografía web
- FontSquirrel Kit
- GoogleFonts

## Tipografía Web

Hasta hace muy poco si queríamos que nuestros diseños se vieran más o menos igual para todos los usuarios había que conformarse con las fuentes que ya sabíamos que estaban instaladas en la mayoría de los ordenadores.

CSS creo una regla llamada `@font-face` que enlaza directamente al archivo de una tipografía que esta subida a un servidor. Esta **regla** es interpretada por el navegador que es el encargado de embeber la tipografía.

A continuación verás los servicios mas usados.

## FontSquirrel Kit

Si no queremos usar nuestras propias fuentes FontSquirrel nos ofrece cientos de tipografías libres de derecho que no será necesario convertir, porque se nos permite descargar el Kit o el paquete con todos los archivos y el código necesario.

Link: <http://www.fontsquirrel.com/fonts>

## GoogleFonts

Un servicio de Google para implementar `@font-face`, donde se puede elegir entre 500 tipografías distintas. La gran ventaja, es que te proporcionan un código que debemos incluir en nuestro archivo css y de esta manera no es necesario que la tipografía esté alojada en nuestro propio servidor.

Link: <http://www.google.com/fonts/>

### ¿Como hacer para incorporar una Googlefont a nuestro sitio?

1) Ingresamos a <http://www.google.com/fonts/>

En la columna de la derecha se muestra el número de fuentes disponibles, el buscador interno y diversos filtros. En la zona central de la página vemos el listado de fuentes y una muestra.





Cuando encontremos la tipografía que nos guste, se debe hacer clic en la opción Quick-use que hay en la parte inferior del cuadro. Tras esto, se muestra una página, donde se indican los pasos a seguir para usarla:

1. Elegir el estilo deseado. También se informa del impacto que tiene la fuente seleccionada en la carga de la página.
2. Elegir el juego de caracteres que deseamos.
3. Incluir en nuestro sitio web la fuente.
4. Especificar que queremos aplicar la fuente en los elementos deseados de un sitio web. Hacemos uso de CSS.

Por ejemplo, si decidimos usar la fuente “Euphoria Script”, creada por Sabrina López, incluiremos la siguiente línea de código dentro del elemento `<head>` del documento HTML.

```
<link href='http://fonts.googleapis.com/css?family=Euphoria+Script' rel='stylesheet' type='text/css'>
```

Con esto ya hemos incluido de alguna forma la fuente “Euphoria Script” en nuestro sitio web. Ahora, podremos aplicarla a los elementos que deseemos mediante la propiedad `font-family` de CSS, como se muestra en el siguiente ejemplo, que aplica esta fuente al elemento `<h1>`.

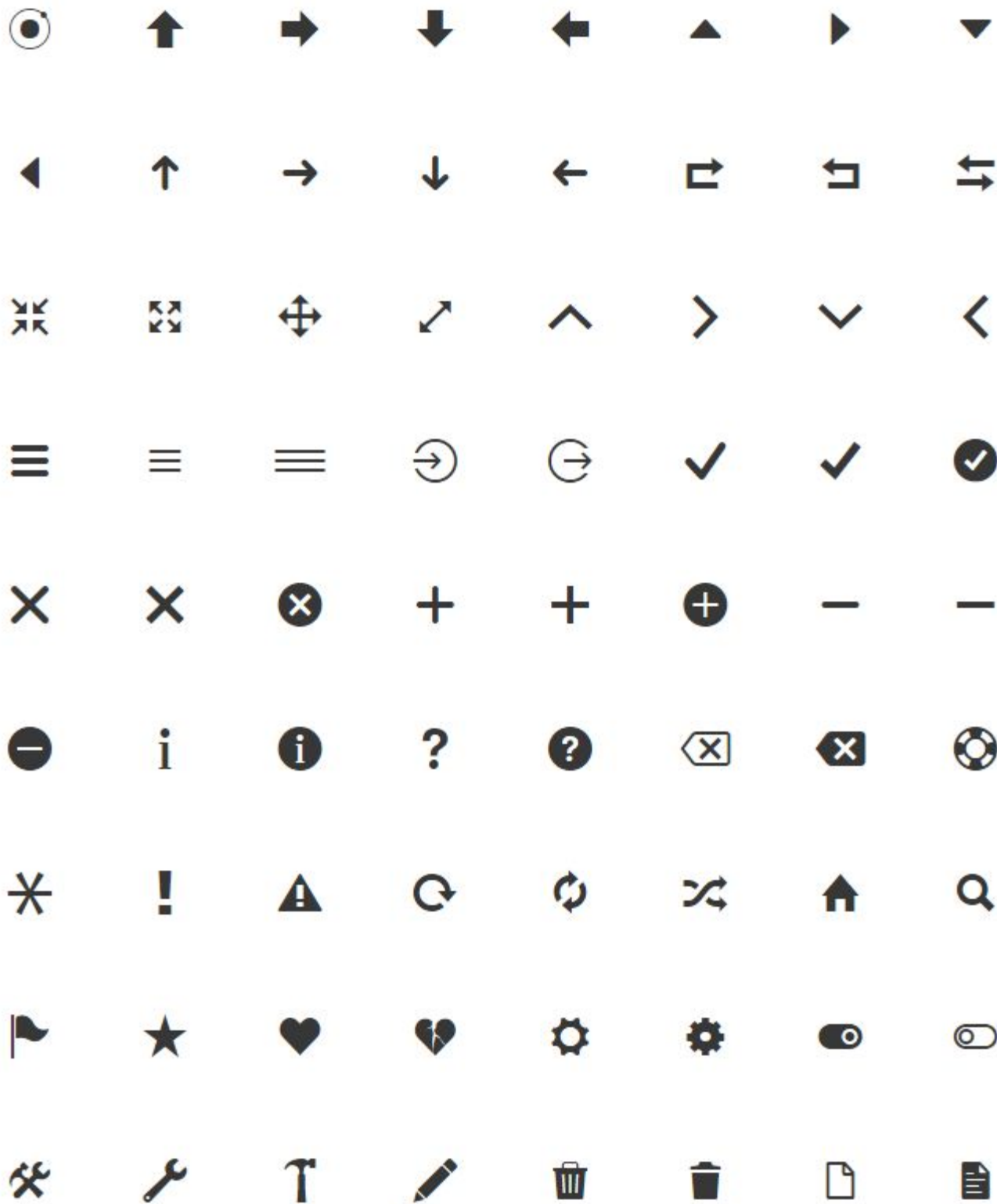
```
h1 { font-family: 'Euphoria Script', cursive; }
```

Esta es la manera más utilizada para agregar fuentes a nuestro sitio web.

## Ionicons

Ionicons es una interesante plataforma que trae más de 400 fuentes de iconos gratuitos de calidad listos para ser empleados en cualquier tipo de proyecto web (comerciales, personales, etc.) y cuya temática va desde iconos sociales, multimedia, comentarios, cargadores, mapas, hasta otros elementos de interfaz de usuario.

```
http://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css
```



Para más ejemplos y obtener mayor información se puede visitar: <http://ionicframework.com/getting-started/>

# Selector de clase

Si queremos aplicar estilos a un elemento en particular de nuestra página, el mejor modo es utilizar el atributo `class` de HTML sobre ese elemento. De esta manera le podemos indicar directamente la regla CSS que le queremos aplicar.

Ejemplo:

```
<body>
  <p>Loem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et est adipiscing accumsan...</p>
  <p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

Supongamos que le queremos aplicar estilos CSS sólo al primer párrafo, ¿Cómo hacemos? El selector de tipo o etiqueta `<p>` no se puede utilizar porque seleccionaría todos los párrafos. El selector descendente (`body p`) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio. Entonces, lo que hacemos, es agregar a la etiqueta `<p>` el atributo llamado `class`:

```
<body>
  <p class="destacado">Loem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et est adipiscing accumsan...</p>
  <p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

En el archivo CSS, se crea una nueva regla llamada `destacado` con todos los estilos que se van a aplicar al elemento.

Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo `class` con un punto (.) tal y como muestra el siguiente ejemplo:

```
.destacado { color: red; }
```

El selector `.destacado` se interpreta como "cualquier elemento de la página cuyo atributo `class` sea igual a `destacado`", por lo que solamente el primer párrafo cumple esa condición.

Este tipo de selectores se llaman selectores de clase. La principal característica de este es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo

Ver ejemplo:

HTML

CSS

Result

# Fondos

Unos de los atributos más utilizados es el fondo de la caja del elemento. El fondo puede ser un color simple o una imagen. Este solamente se visualiza en el área ocupada por el contenido y su relleno.

Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento `<body>`.

CSS define cinco propiedades para establecer el fondo de cada elemento (`background-color`, `background-image`, `background-repeat`, `background-attachment`, `background-position`)

## Color de fondo

La forma de indicar el color de fondo de una etiqueta desde CSS es con la propiedad `background-color`. Como valor, podemos utilizar uno de los colores predeterminados, escribiendo simplemente la palabra, por ejemplo `green` o podemos utilizar el sistema hexadecimal, escribiendo el código de color.

Ejemplo:

```
body {
  background-color: #F5F5F5;
}
```

## Imagen de fondo

No solo podemos aplicar color, sino que además podemos aplicar una imagen como fondo de una etiqueta. La propiedad a utilizar, en este caso, es `background-image`

Las imágenes de fondo se indican a través de su URL, que puede ser absoluta o relativa. Suele ser recomendable crear una carpeta de imágenes que se encuentre en el mismo directorio que los archivos CSS y que almacene todas las imágenes utilizadas en el diseño de las páginas.

CSS nos permite establecer de forma simultánea un color y una imagen de fondo. De hecho, es recomendable aplicar ambas, ya que si por alguna razón, no se muestra la imagen de fondo, vamos a poder ver el color.

Ejemplo:

```
body {
  background-image: url("imagenes/fondo.jpg")
}
```

Algo importante a tener en cuenta, son los formatos de imágenes que podemos utilizar para el fondo. Los mismos son: JPG, PNG o GIF. En todos los casos, tienen que estar en modo RGB.

## Bordes

CSS permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su grosor, color y estilo.

El grosor de los bordes, se indica mediante una unidad de medida, como el pixel. La propiedad que utilizamos para esto, se llama `border-width`.

Como ya vimos antes, podemos utilizar esta propiedad indicando específicamente a que lado le queremos aplicar el estilo, o podemos utilizar la propiedad por sí sola.

Ejemplo:

```
div {
  border-top-width: 10px;
  border-right-width: 5px;
  border-bottom-width: 10px;
  border-left-width: 5px;
}

div {
  border-width: 10px;
}
```

Si indicamos un solo valor, este se aplicará por igual a los cuatro bordes. Pero si se indican los cuatro valores, el orden de aplicación es superior, derecho, inferior e izquierdo.

## Estilo

En este caso, CSS no da varias opciones. El estilo de nuestro borde puede ser punteado, sólido, con rayas, doble, etc.

Para poder seleccionar que estilo queremos, debemos utilizar uno los valores que CSS ya tiene definido.

Los valores son: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset

Los mismos se ven a de la siguiente manera:

Para establecer de forma simultánea los estilos de todos los bordes de una caja, utilizamos la propiedad llamada border-style.

Ejemplo:

```
div {
  border-top-style: dashed;
  border-right-style: double;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

Al igual que sucede con los márgenes y los padding, CSS define una serie de propiedades de tipo "shorthand" que permiten establecer todos los atributos de los bordes de forma simultánea.

Esta propiedad global se llama border.

Ejemplo:

```
div {
  border: 1px solid #cccccc;
}
```

De esta manera le estamos diciendo que nuestra caja va a tener un borde sólido, de 1px de grosor en sus cuatro lados y de color gris.

## Posicionamiento

Además de seleccionar el tipo de repetición de las imágenes de fondo, CSS permite controlar la posición de la imagen dentro del fondo del elemento mediante la propiedad background-position.

En los casos que elegimos que la imagen no se repita, lo que sucede es que esta imagen se alinea con la esquina superior izquierda de la caja. La propiedad background-position nos permite elegir la posición de dicha imagen dentro de la caja.

Esta propiedad puede manejarse con palabras clave, como `top`, `bottom`, `center`, `left` y `right`. Estas también se pueden combinar. Por ejemplo podemos decir que nuestra imagen de fondo esté alineada `top center`.

```
body {  
    background-image: url("imagenes/fondo.jpg") repeat-x;  
    background-position: top center;  
}
```

Lo que nos va a mostrar, es nuestra imagen alineada arriba al centro.

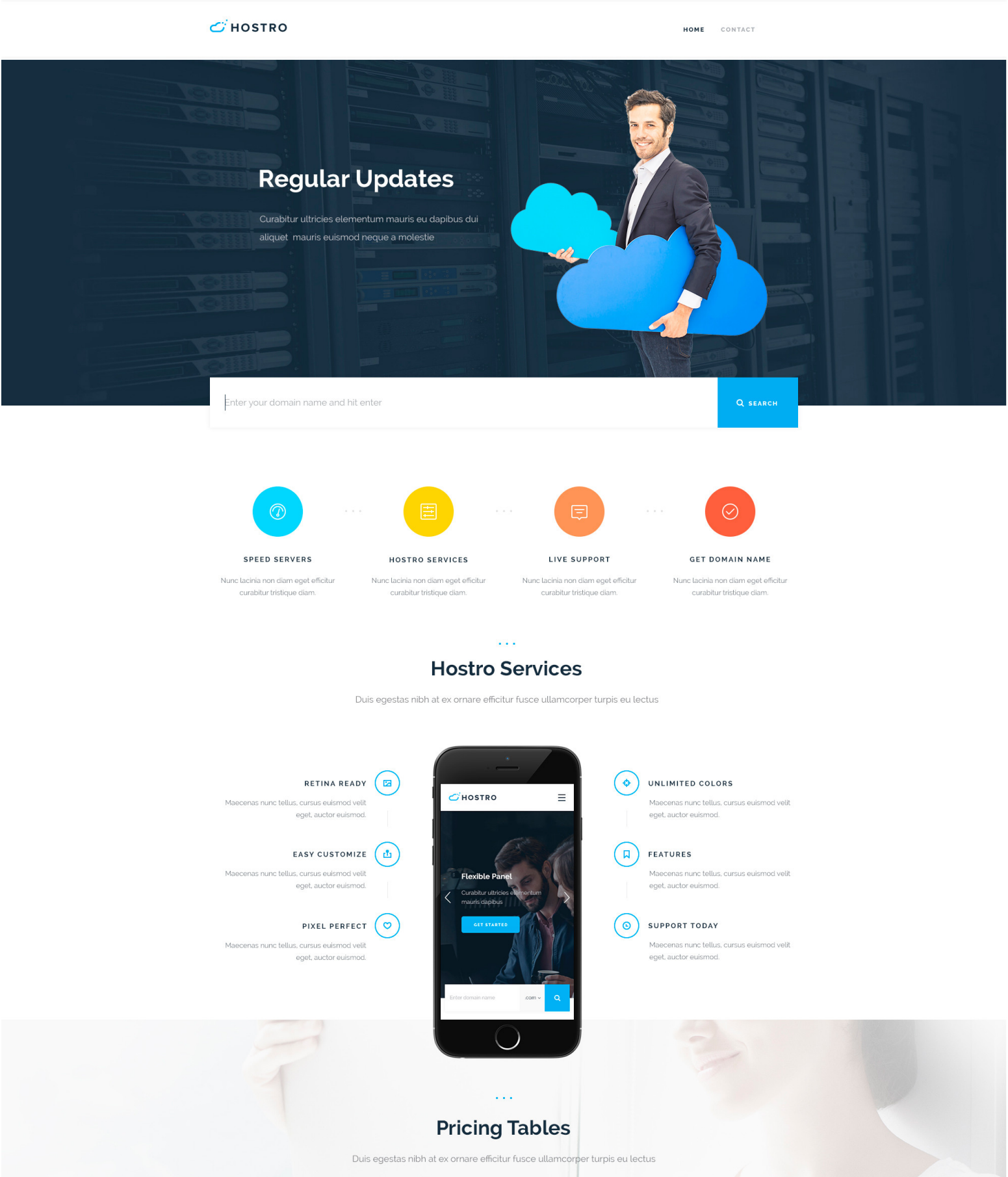
Otra opción es utilizar números, de esta manera vamos a poder indicar la coordenada exacta de la posición de la imagen con respecto a la caja en donde está ubicada como fondo.

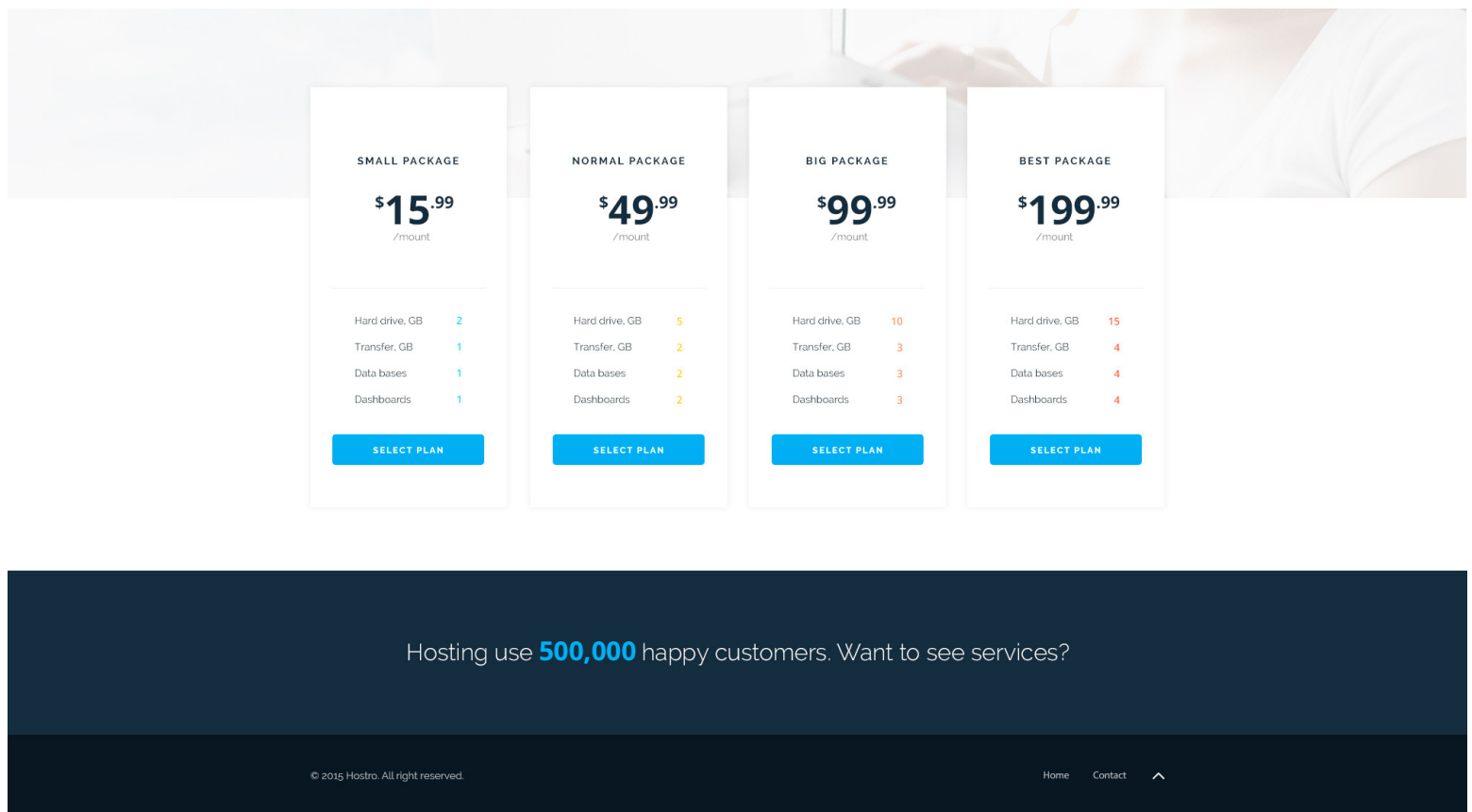
El modo de escribirlo es el siguiente:

```
body {  
    background-image: url("imagenes/fondo.jpg") repeat-x;  
    background-position: 10px 20px;  
}
```

Como pueden ver en el ejemplo, tenemos dos números, el primer número es la coordenada en X (horizontal) y el segundo número es la coordenada en Y (vertical).

# Trabajo Práctico 3: CSS





## Descripción:

1. Definir selectores y clases de cada elemento.
2. Utilizar las fuentes donde sea necesario(Raleway y Open Sans)
3. Cambiar colores y fotos de fondos

## Objetivo:

- Familiarizarse con las propiedades de css

## Requerimientos:

1. Haber realizado los trabajos anteriores.
2. Tener conocimientos css en cambios de colores y fuentes
3. Crear una carpeta para los assets dentro del proyecto
4. Subir la resolución de tu trabajo a tu repositorio de Github en la subcarpeta “CSS TP3”.

## Bonus:

- - Busca sobre las propiedades que permite CSS en la interfaz de usuario

## Recursos:

-



- [Google Fonts](#)
- [Selectores](#)
- [Estilos de texto](#)
- [Propiedades UI \(bonus\)](#)
- [Assets](#)