

Desarrollo Web con HTML

Módulo 2

Propiedades avanzadas CSS

Estilos y vínculos

Por *default*, los vínculos se comportan de cierta manera: tienen un color asignado, que a veces cambia cuando pasamos el mouse por arriba.

Estas características predeterminadas pueden modificarse con las *pseudo clases*. Existen varias opciones:

<code>a:hover</code>	Cuando pasamos el <i>mouse</i> por encima del vínculo.
<code>a:link</code>	Cuando aún no hemos visitado el vínculo.
<code>a:visited</code>	Cuando hemos visitado el vínculo.
<code>a:active</code>	Cuando le hacemos clic al vínculo.

```
<style>
a:hover {color: pink}
</style>
```



Veámoslo con un ejemplo:

```
a { color: white; text-decoration: none; }  
a:hover { text-decoration: underline; }
```

Primera línea:

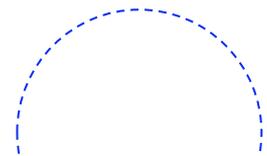
Afecta a los links, en general.

- **color: white** le asigna color blanco.
- **text-decoration: none** elimina el subrayado (que tienen los links por defecto).

Segunda línea:

Modifica la apariencia de los links al pasarle el mouse por arriba.

- **text-decoration: underline** agrega el subrayado.



Alineando contenido

En el código del ejemplo en la derecha trabajamos la alineación de los párrafos mediante la propiedad **text-align**.

En el primer caso, los elementos están orientados hacia la izquierda, en el segundo hacia el centro.

Si quiero orientar mi texto hacia la derecha utilizaré el tercer valor y para justificarlo el último.

Actualmente podemos ver cómo también figuran los valores **end** y **start**, estos valores también son válidos y responden a **left** y **right** respectivamente. Es decir **start/left** de izquierda a derecha y **right/end** de derecha a izquierda.

```
<style>
p { text-align: left; }
p { text-align: center; }
p { text-align: right; }
p { text-align: justify; }

</style>
```

Valores nuevos CSS: experimental

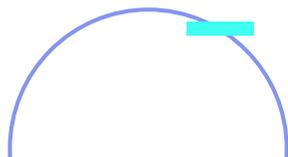
Justify

Con respecto al valor **justify**, justificar un texto es obligar a que todas las líneas de un texto tengan el **mismo ancho**.

Sin embargo, más allá de la sugerencia de que no siempre este valor queda bien muchas veces lo tomamos y el resultado es que todo se obliga a generar el mismo ancho menos la última línea.

Entonces, actualmente se encuentra en fase experimental el valor **justify-all** de modo tal que a futuro esa última línea también pueda ser forzada. Esto quiere decir que este valor aún no es tomado por la mayoría de los navegadores.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita illum, asperiores eveniet vel repellendus eligendi magnam neque iste iure earum ab perferendis sed aut dolorum molestias quae accusantium in dignissimos! Lorem ipsum dolor sit amet consectetur adipisicing elit. Atque doloribus facilis excepturi libero tempore! Reprehenderit iure dignissimos ipsa, fugiat id vitae accusamus magnam dolor, vel possimus commodi sequi suscipit corrupti?



Text-align-last

Para resolver el punto anterior, siendo que aún ese valor (`justify-all`) no ha sido aprobado por todos los navegadores, existe la propiedad **`text-align-last`**. Esta propiedad también relativamente nueva, pero funciona en la mayoría de los navegadores actuales.

La idea es forzar a que suceda algo con **la última línea del texto**, ya sea que se quiera justificar al mismo ancho que el resto, o que se quiera un comportamiento diferente de ésta.

El resultado de aplicar **`text-align-last`** será:

```
<> index.html # estilos.css X
# estilos.css > p
1
2
3 p{ text-align:justify;
4   text-align-last: justify ;}
5
```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Expedita illum, asperiores eveniet vel repellendus eligendi magnam neque iste iure earum ab perferendis sed aut dolorum molestias quae accusantium in dignissimos! Lorem ipsum dolor sit amet consectetur adipiscing elit. Atque doloribus facilis excepturi libero tempore! Reprehenderit iure dignissimos ipsa, fugiat id vitae accusamus magnam dolor, vel possimus commodi sequi suscipit corrupti?

Por supuesto, también podemos **sólo afectar esa línea** sin indicar nada distinto para el resto del texto:

```
<> index.html × # estilos.css ×
# estilos.css > ...
1
2
3   p{ text-align-last: center ;}
4
5
```

De modo tal que el resultado será que todo el texto se orientará de **izquierda a derecha** (valor por defecto `start` o `left`) y luego la última línea se encontrará **centrada**.

Lorem ipsum dolor sit amet consectetur adipiscing elit. Expedita illum, asperiores eveniet vel repellendus eligendi magnam neque iste iure earum ab perferendis sed aut dolorum molestias quae accusantium in dignissimos! Lorem ipsum dolor sit amet consectetur adipiscing elit. Atque doloribus facilis excepturi libero tempore! Reprehenderit iure dignissimos ipsa, fugiat id vitae accusamus magnam dolor, vel possimus commodi sequi suscipit corrupti?

Transformación de texto

Otra propiedad útil es **text-transform**. La misma nos permite transformar un texto, por ejemplo, en mayúsculas o minúsculas.

Su valores posibles son:

- uppercase.
- lowercase.
- capitalize.
- none.

```
nav a {  
  |   text-transform: capitalize;  
}
```

Decoración de texto

Otra propiedad muy utilizada especialmente en los vínculos es **text-decoration**. Nos **permite subrayar (o no hacerlo)** un elemento.

En el ejemplo estamos utilizando un *selector descendente* y una *pseudo clase*. En este caso afectamos sólo a aquellos vínculos dentro de nav cuándo pasamos el mouse sobre ellos.

Los valores posibles son, entre otros:

- **underline.**
- **none.**
- **overline.**
- **line-through.**

```
nav a:hover {  
  text-decoration: underline;  
}
```

Nuevas posibilidades

Actualmente a la propiedad **text-decoration** podemos agregarle estilo y color imitando la propiedad border, que tiene funciones similares. Siempre indicamos el estilo y luego el color.

Algunos **valores posibles de estilos** son:

- **solid** (sólido).
- **dashed** (una línea al lado de la otra).
- **dotted** (un punto al lado del otro).
- **wavy** (una onda).

```
# estilos.css ×  
# estilos.css > h1  
1 h1 {  
2   text-decoration: dashed green underline;  
3 }
```

Enunciado

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita illum, asperiores eveniet vel repellendus eligendi magnam neque iste iure earum ab perferendis sed aut dolorum molestias quae accusantium in dignissimos! Lorem ipsum dolor sit amet consectetur adipisicing elit. Atque doloribus facilis excepturi libero tempore! Reprehenderit iure dignissimos ipsa, fugiat id vitae accusamus magnam dolor, vel possimus commodi sequi suscipit corrupti?

Si por ejemplo utilizamos el valor **wavy** con otro color y decoración el resultado será:

```
# estilos.css X
# estilos.css > h1
1 h1 {
2   text-decoration: wavy red line-through;
3 }
```



Enunciado



Anchos: propiedad *width*

Esta propiedad indica el **ancho** de los elementos de mi documento.

Es sólo aplicable a **elementos de bloque o línea bloque** (como las imágenes, videos, etc.) Por ejemplo, si trato de implementar esta propiedad en un vínculo, no funcionará.

Los valores posibles son amplios (puedo trabajar con **diferentes medidas de longitud** como vh, vw, %, etc.)

Durante este curso trabajaremos con px y %.

```
footer ,  
main ,  
header ,  
aside {  
    width: 100%;  
}
```

Altos: propiedad *height*

Esta propiedad indica el **alto** de los elementos de mi documento.

Es sólo aplicable a **elementos de bloque o línea bloque** (como las imágenes, videos, etc.). Por ejemplo, si trato de implementar esta propiedad en un vínculo no funcionará.

Los valores posibles son amplios (puedo trabajar con diferentes medidas de longitud, como vh, vw, %, etc.) **Durante este curso trabajaremos con px y %.**

Si no indico un valor, este será *auto*. Lo que significa que fluye con el contenido del elemento que tenga aplicada esta propiedad.

```
footer {  
  height: 100px;  
  padding: 10px;  
  background-color:  #FCEE21;  
}
```

Márgenes en el documento

La **propiedad margin** fija un espacio entre elementos contiguos. Su valores pueden ser medidas de longitud o bien **un porcentaje en base al ancho o alto** (dependiendo el margen) del elemento contenedor.

También puede trabajarse con **valores negativos**, aunque no es lo más común, en todo caso, la lógica sería la siguiente:

Si genero un margen de **izquierda negativo** más que alejar al elemento situado hacia ese lado, lo iré acercando, incluso esto puede generar una superposición de elementos.

A continuación, vamos a entender mejor de qué se trata esta propiedad y qué alternativas tenemos.

La propiedad de margen se puede trabajar de forma simple en nuestro documento.

Veamos el ejemplo a la derecha. En este caso, se ha indicado los valores para los márgenes de un vínculo, pero lo interesante es que esto **se puede simplificar** como en el ejemplo debajo:

```
margin: valor1;

margin-top: valor1;
margin-left: valor1;
margin-right: valor1;
margin-bottom: valor1;
```

```
<!DOCTYPE html>
<html>
<head>
  <title> CSS</title>
  <style>

  a {
margin-top: 12px;
margin-left: 10%;
margin-right: -12px;
margin-bottom: 10px;
  }

</style>
</head>
<body>
```

Padding en mi documento

Esta propiedad indica el **espacio entre los bordes de un elemento y su contenido**. A diferencia del margen, pocos elementos tienen **padding**. Por ejemplo, las listas tienen **padding**. Si queremos quitar el **padding**, lo haremos como en la imagen de la derecha. Si buscamos asignar diferentes valores lo haremos como se muestra debajo:

```
div { padding-top: 12px;  
padding-left: 10%;  
padding-right: -12px;  
padding-bottom: 10px;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
  <title> CSS</title>  
  <style>  
  
<style>  
  
ul, ol { margin: 0; padding: 0;}  
  
</style>  
  
</style>  
</head>
```

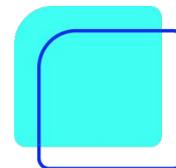
Box-sizing

Esta propiedad **permite o no que el padding se sume al ancho o alto total de un elemento.**

Por ejemplo, si a un elemento le agregamos:
`width 800px + padding 10px` = el ancho total quedará en `820px`.

Es decir, `10px` de padding de cada lado, en total `20px`, se sumarán a los `800px` de width y el elemento será más grande que su width inicial.

Si bien podemos trabajar con esta característica, es ideal modificarla a través de la propiedad **box-sizing** y específicamente con su valor **border-box**, para así hacer que ese padding se tome del width inicial y, finalmente: `width 800px + padding 10px` = quedará en `800px`.



Generalmente, como deseamos aplicar esta propiedad a todos los elementos del documento, la implementamos a través del **selector *** también llamado ***Universal***.

Este selector es el menos específico de los selectores y **afecta a todos los elementos**. Siempre ubicamos este selector **al comienzo de nuestra hoja de estilos**.

```
* {  
    box-sizing: border-box;  
}
```



Propiedad *font-weight*

Si queremos poner en **negrita** un contenido lo podemos hacer de la siguiente manera, a través de la propiedad **font-weight**.

Si bien lo mencionamos al ver *Google Fonts*, es importante recordar que **si deseamos trabajar con varios valores del eje weight debemos importar todos aquellos a utilizar,**

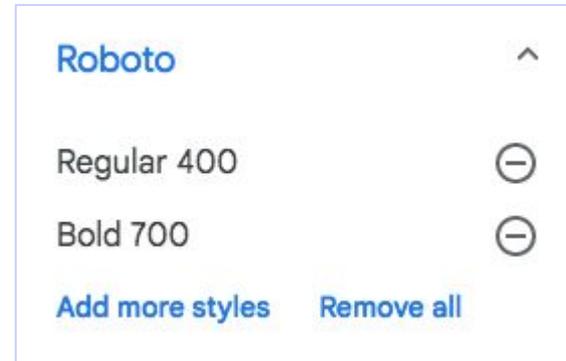
si no el navegador no generará una versión real de la tipografía en ese eje.

Esta propiedad de CSS tiene la posibilidad de trabajar con valores como **bold** donde el contenido se pone en negrita o **normal** para que éste deje de estarlo.

```
<p style="width:200px; height: 30px;background-color: red; color: white; text-align: center; font-weight: bold;" > Mi párrafo </p>
```

Esta propiedad tiene **valores numéricos**. Por ejemplo, si importamos *Roboto* en su valor 400 y 700 haremos referencia a esa variable mencionando el número y no diciendo `bold` o `normal`.

```
body {  
  color: #333;  
  font-weight: 400;  
  font-size: 24px;  
  font-family: 'Roboto', sans-serif;  
  margin: 0;  
}
```



Aplicando la propiedad *font-style*

Si queremos poner en **itálica** un contenido lo podemos hacer de la siguiente manera, a través de la propiedad **font-style**.

Esta propiedad de CSS tiene la posibilidad de trabajar con valores como **italic** donde el contenido se pone en itálica o **normal** para que éste deje de estarlo:

```
<p style="width:200px; height: 30px;background-color: red; color: white; text-align: center; font-style: italic;" > Mi párrafo </p>
```

Tipografía: alertas

Como se mencionó en el módulo anterior, es importante prestar atención a la tipografía y sus rangos (*weight*, *style*, etc.) para trabajarlas de manera correcta con las propiedades antes vistas.

Por ejemplo

No se puede trabajar en Roboto con un peso 200 porque no existe. Si se escribiera en la hoja de estilo como vemos en la imagen superior derecha, el navegador buscará el peso más cercano dentro de los importados (se debe importar para usarlo).

Nota: sugerimos revisar y realizar todas las prácticas del módulo anterior.

```
# estilos.css X
# estilos.css > p
1 p {
2   font-weight: 200;
3 }
```

Thin 100

Whereas recognition of the inherent dignity

Thin 100 Italic

Whereas recognition of the inherent dignity

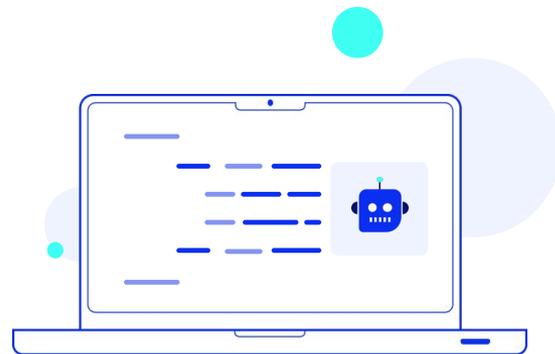
Propiedades avanzadas CSS: integración con IA

Trabajo con IA

Si bien hemos ya trabajado con el Chat GPT, nos parece interesante sugerir la siguiente herramienta [ZZZ Code AI](#).

La misma permite sugerir un **código específico y nos ayuda a generarlo**. Por ejemplo en este caso vamos a pedir ayuda para generar un css adecuado a nuestro desarrollo donde un header tendrá ancho, padding y margin.

Recordemos que **las herramientas de IA son sólo un complemento y debemos corregirlas**, encauzarlas y guiarlas en todo el proceso.



El **prompt** a escribir lucirá como el siguiente:

Debemos prestar especial atención en la opción > **generar un código con explicación** para obtener mayores detalles.

Contributing to this website **take 10 seconds**. Help us this week with: Step 1: Click on [EF Core 8 - BulkInsertOptimized](#) Step 2: And share the page everywhere!'." data-bbox="308 183 934 863"/>

AI Code Generator

Language / Software

html

Description of the code to generate

Quiero un css que ponga un ancho de 200px a un header, con un padding de 10px y un margen de 20px

I want

Generate code with explanation

Tone

Professional

Output in

English

Do you know this website cost us over 150\$ per day to stay free?
[Contributing](#) to this website **take 10 seconds**. Help us this week with:
Step 1: Click on [EF Core 8 - BulkInsertOptimized](#)
Step 2: And share the page everywhere!

El **resultado** será similar al siguiente:



```
language-html
```

```
Copy code
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Styled Header</title>
  <style>
    header {
      width: 200px;
      padding: 10px;
      margin: 20px;
      background-color: lightblue;
      color: white;
      text-align: center;
    }
  </style>
</head>
<body>
  <header>
    This is a styled header element.
  </header>
</body>
</html>
```

Revisión

- Repasar todas las **propiedades de CSS** vistas hasta el momento.
- Cambiar **tamaños y estilos** para ganar más capacidad de trabajar con ambos lenguajes.
- Incluir en el proyecto guardado las **propiedades y valores** vistos.
- Realizar las preguntas necesarias al/la docente antes de continuar.



**¡Sigamos
trabajando!**