# ÍNDICE

[**ÍNDICE 1**](#_qk542gkpvj5p)

[**CÓMO UNIR UN CSS A UN HTML 2**](#_tghadfa43qch)

[**4.3.1. Color y fondo 3**](#_27cg9asxe6ov)

[**4.3.2. Texto 4**](#_yn6lw6uwmnih)

[**4.3.3. Fuente 5**](#_cc9ru9a70r9d)

[**4.3.4. Modelos de caja 6**](#_7fjvaisspr1z)

[**4.3.5. Enlaces 8**](#_d3bymvwn5t6h)

[**4.3.6. Listas 9**](#_bj2ppt7sf8na)

[**4.3.7. Tablas 10**](#_39ofcbfy3swt)

[**4.4. Definición y uso de clases 10**](#_dtul3n7dwghq)

[**4.5. El atributo ID y la etiqueta SPAN 12**](#_jd8tb7s7790d)

[**4.6. El modelo de cajas 12**](#_1zcyzz8nx42q)

[**4.6. Mejoras de CSS3 13**](#_illpxl22d6hu)

[1. Personalización de estilos 14](#_7qnkh9fhogcs)

[1.1. Colocación del código CSS 14](#_8a2joxlt88yo)

[1.2. Selectores 15](#_gtdbl0aw92fj)

[1.3. Colores 17](#_ojnavb2qy4j6)

[1.4. Texto 18](#_gv1qb65kpras)

[1.7. Tamaño 19](#_a1vkn3a6myo3)

[1.8. Bordes 20](#_90sftilwhtbw)

[1.9. Márgenes 21](#_n3hrg03k6rhq)

[1.10. Posicionamiento 23](#_mh6t7hn8uorj)

[1.12. Imagen de fondo 24](#_yhzlp4mpu0ed)

[1.16. FlexBox 25](#_3f6fn4uygq9z)

[**COSITAS (HTML) 26**](#_v13hlu3p53sv)

[**ESTRUCTURA: 26**](#_ynlve5b4un3a)

[**FORMATOS: 27**](#_xe22p93cly5h)

[**FORMULARIOS: 27**](#_h33l09n6s0t1)

[Fieldset 29](#_f71bm5egvcbk)

[Label 30](#_88kobmhy1n8b)

[Input 30](#_dwydn1pogx57)

[Button 32](#_ftb03uxxcrcw)

[Select 33](#_wnge3s7pjdlh)

[**Atributos-Formularios 37**](#_787pjkislkqc)

[Name 37](#_9yha9wyhef6d)

[Value 37](#_himk8gpgtyg1)

[Required 38](#_z3uly7srngaf)

[Disabled 38](#_bvlx7pmozrfv)

[Readonly 38](#_s8j04xnflmwn)

[ID 39](#_tru4hppzhxd4)

[CLASS 39](#_yj3jgsfoyw0c)

[**COMENTARIOS 40**](#_xdjm85u09iiu)

[**LISTAS: 40**](#_fvfs5sg3bdm6)

[**Enlaces y marcadores 41**](#_2rni6o5g6qrt)

[**Imágenes: 41**](#_cumsj1gybeiz)

[**Audio y video: 42**](#_rhm3ct7m63qb)

[**Ejemplos css y html: 42**](#_ua13osxspabe)

[● modificar el color de un enlace y hacer el texto mas grande cuando pasas el ratón por encima: 42](#_l3zybmyuko8)

[● Para cambiar el tipo de letra: 43](#_3jousx6ekoqc)

[● Cambiar tamaño y tipo de letra: 43](#_hriuj6lzxgvp)

[● Botones: 43](#_h86ug0wmnndk)

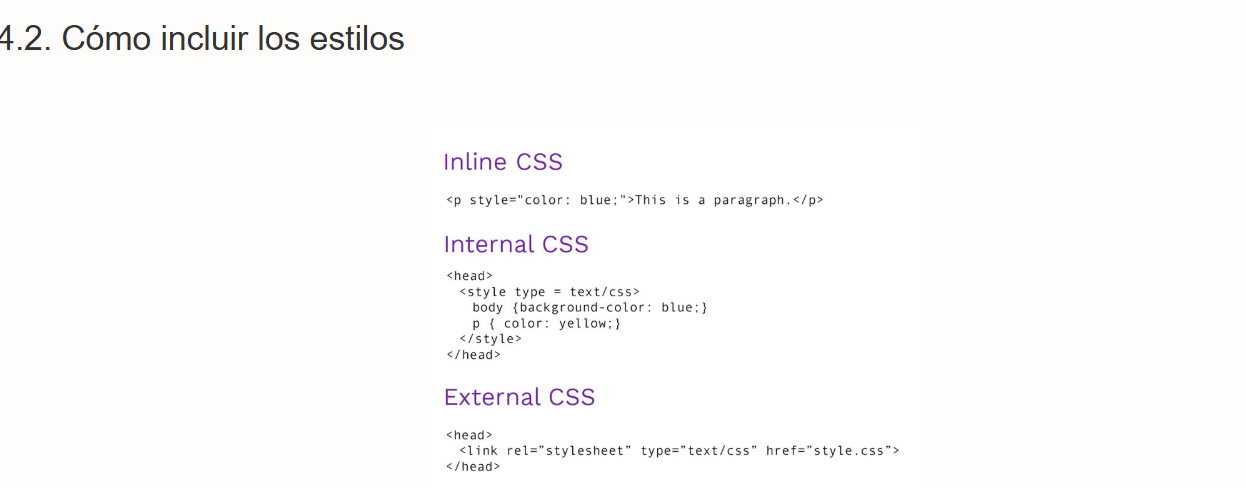
[● pie de página: 43](#_vg4rbw2o6ftw)

[● Secciones 44](#_mo4kz8qu8sdr)

[● Barra de navegación 44](#_5ln5a11t254i)

UA04.1

# CÓMO UNIR UN CSS A UN HTML

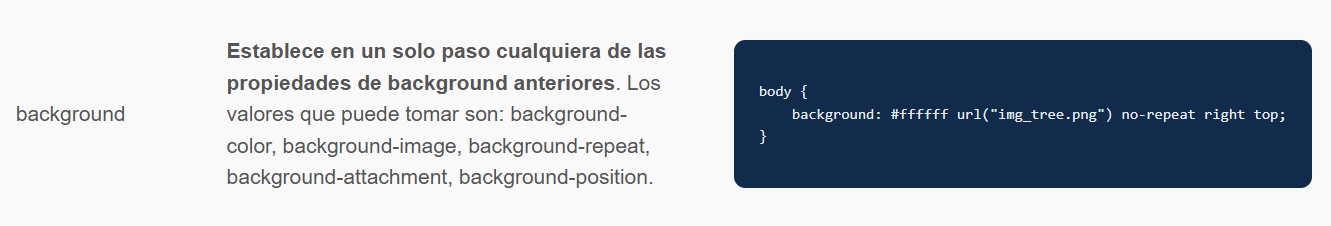


<link rel="stylesheet" type="text/css" href="formatos.css" />

# 4.3.1. Color y fondo



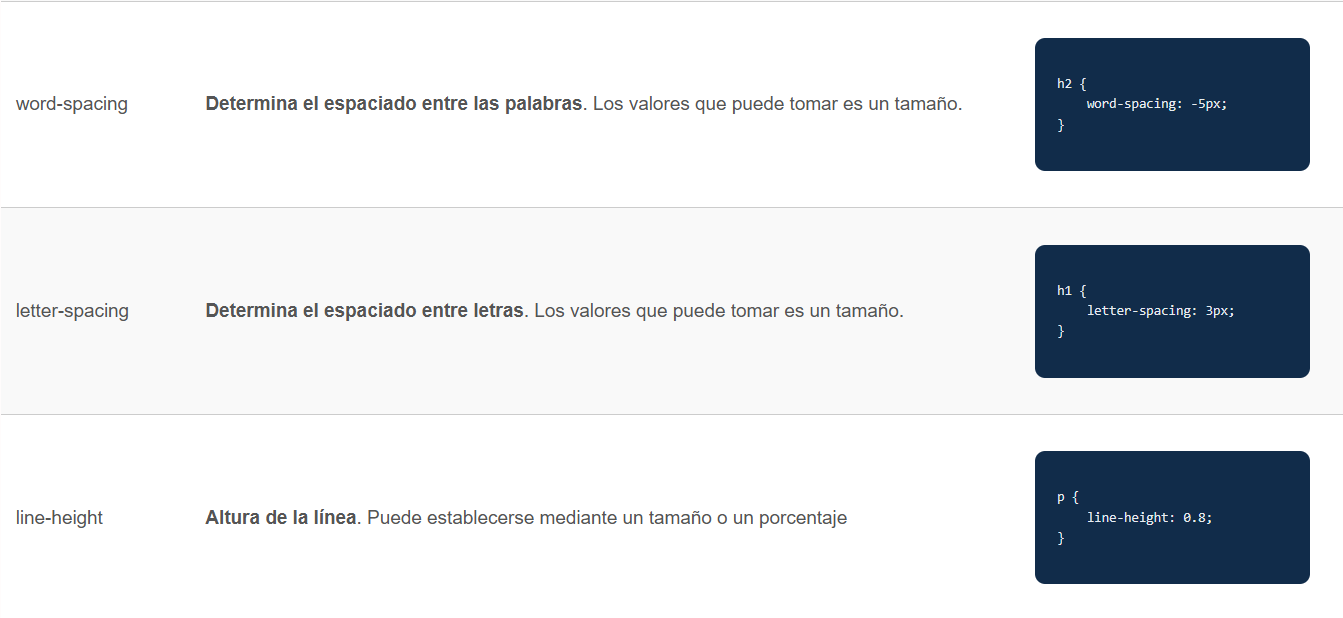




# 

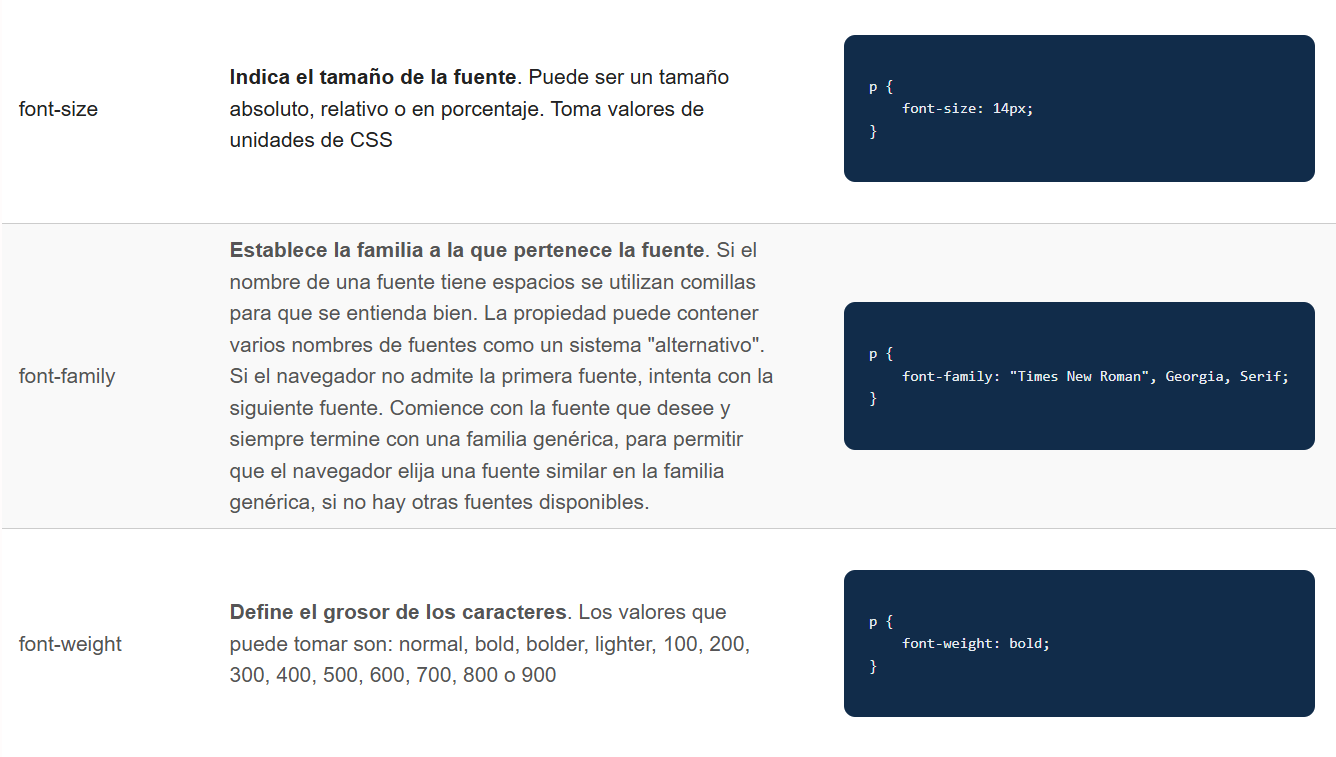
# 4.3.2. Texto

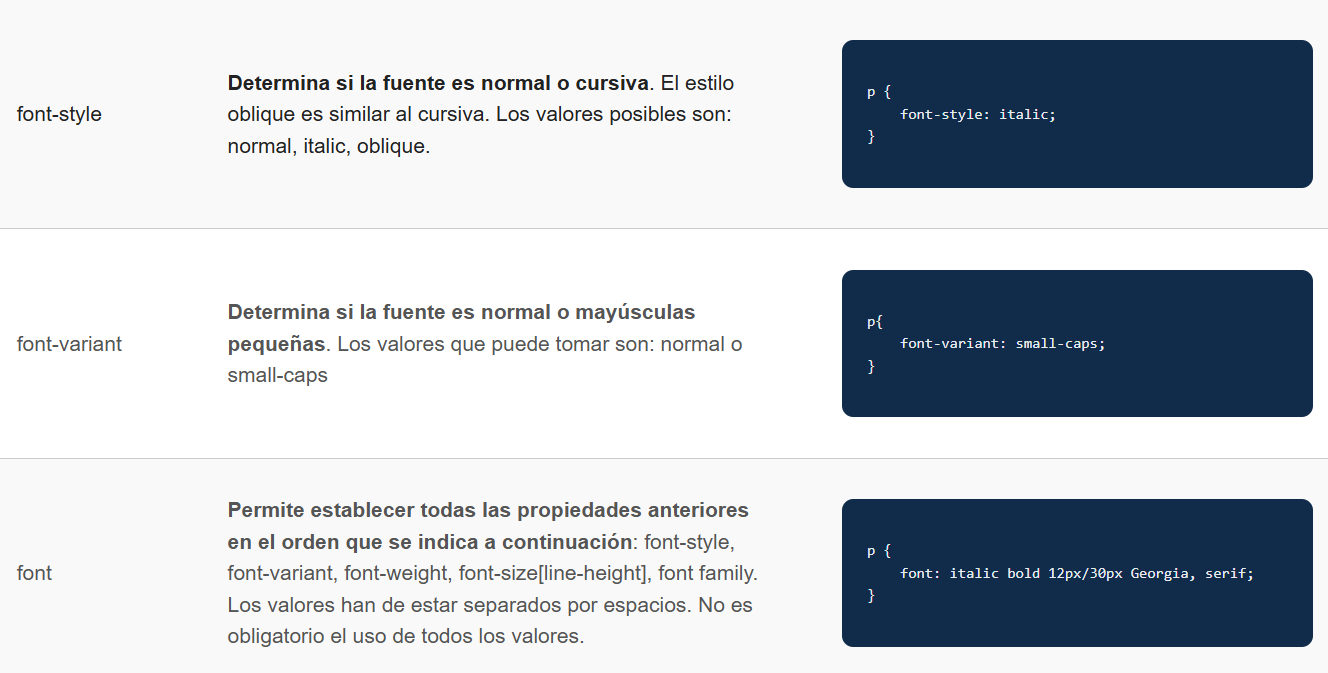




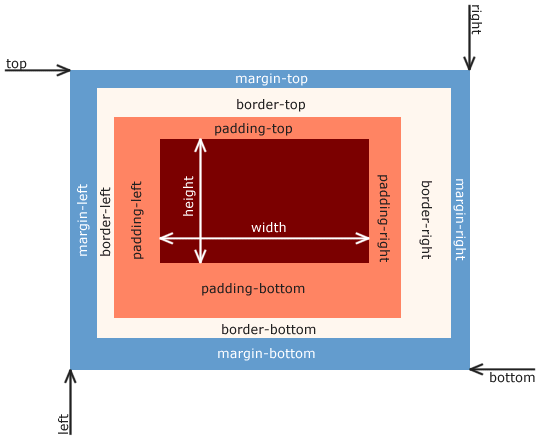
# 

# 4.3.3. Fuente





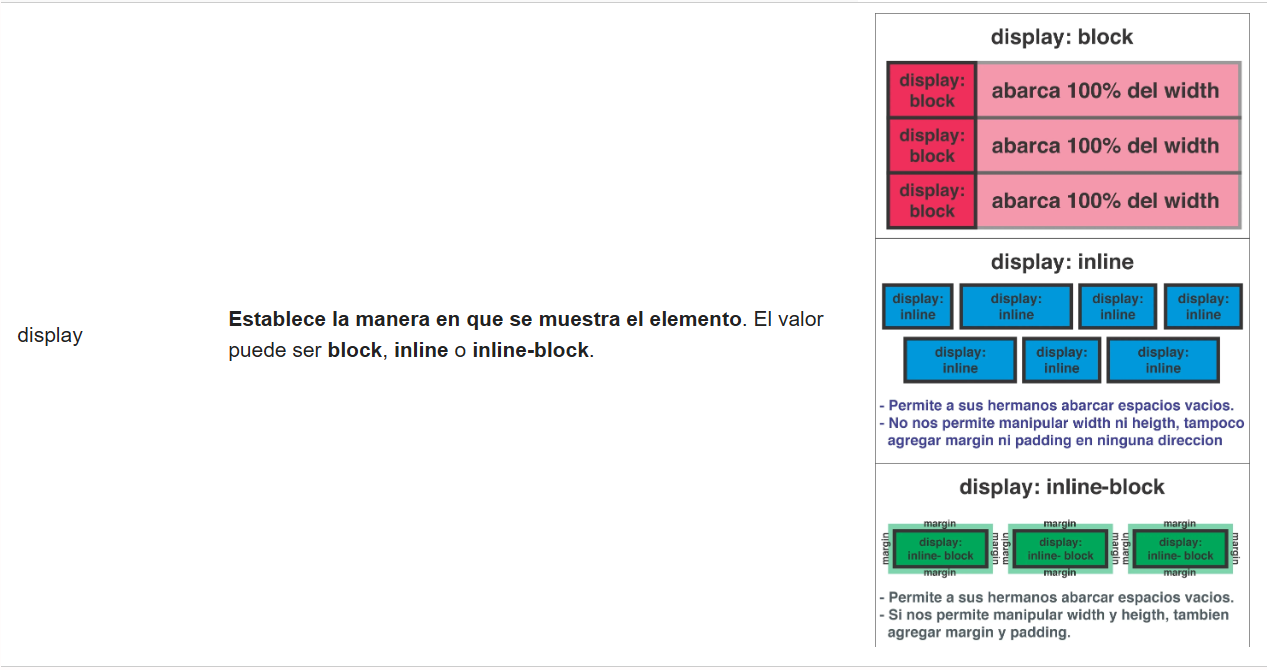
# 4.3.4. Modelos de caja











# 4.3.5. Enlaces

Podemos aplicar cualquier atributo CSS (color, font-family, background, etc.) a los enlaces y darles forma.

Además, los enlaces se pueden diseñar de manera diferente según el estado en el que se encuentren.



Al configurar el estilo para varios estados de enlace, existen algunas reglas de orden:

* **a: hover** DEBE venir después de **:link** y un **: visited**
* **a: active** DEBE venir después de **:hover**

# 4.3.6. Listas

Las propiedades de la lista CSS nos van a permitir:

* Establecer diferentes marcadores de elementos de lista para listas ordenadas
* Establecer diferentes marcadores de elementos de lista para listas desordenadas
* Establecer una imagen como marcador de elemento de lista
* Agregar colores de fondo a listas y elementos de lista



# 4.3.7. Tablas

El siguiente código refleja un ejemplo de página html de aplicación de estilos css a una tabla:

<body>

<table>

<tr>

<th> </th>

<th>Población</th>

<th>Hombres</th>

</tr>

<tr>

<th>Alemania</th>

<td>82.020.578</td>

</tr>

<tr>

<th>Francia</th>

<td>65.578.819</td>

</tr>

<tr>

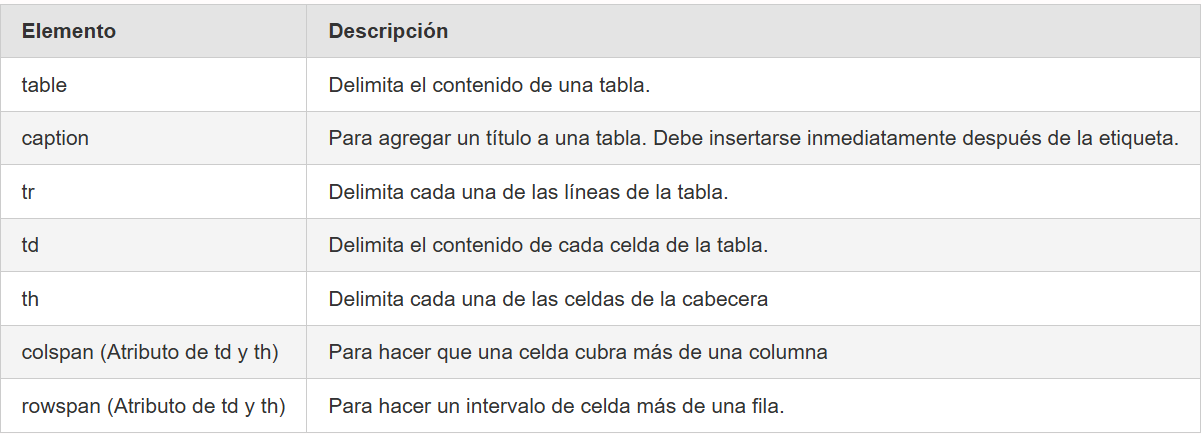
<th>Reino Unido</th>

<td>63.896.071</td>

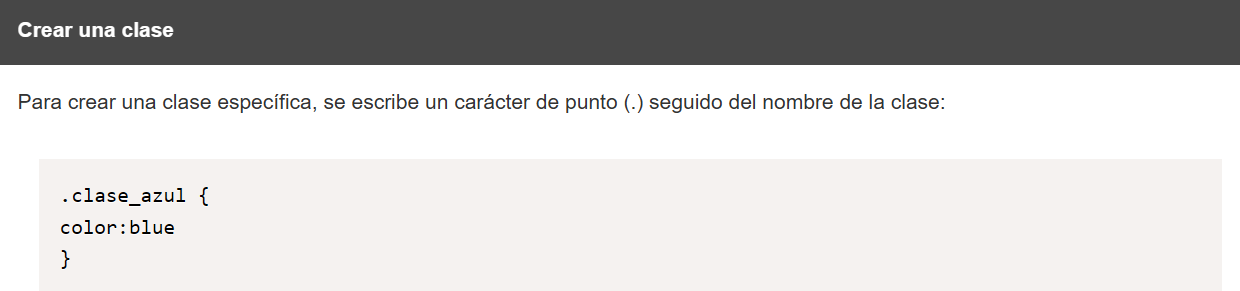
</tr>

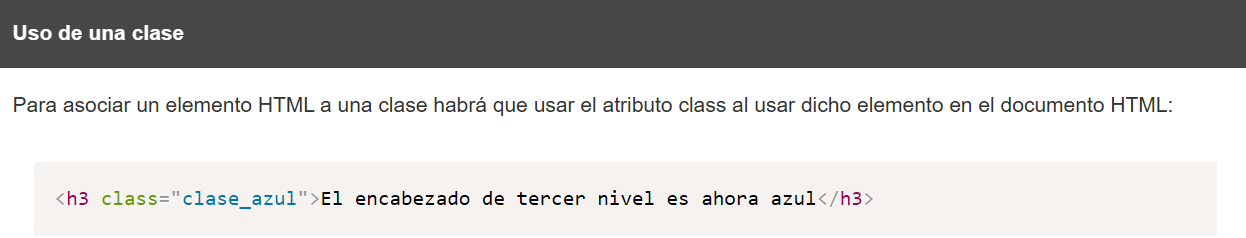
</table>

</body>



# 4.4. Definición y uso de clases

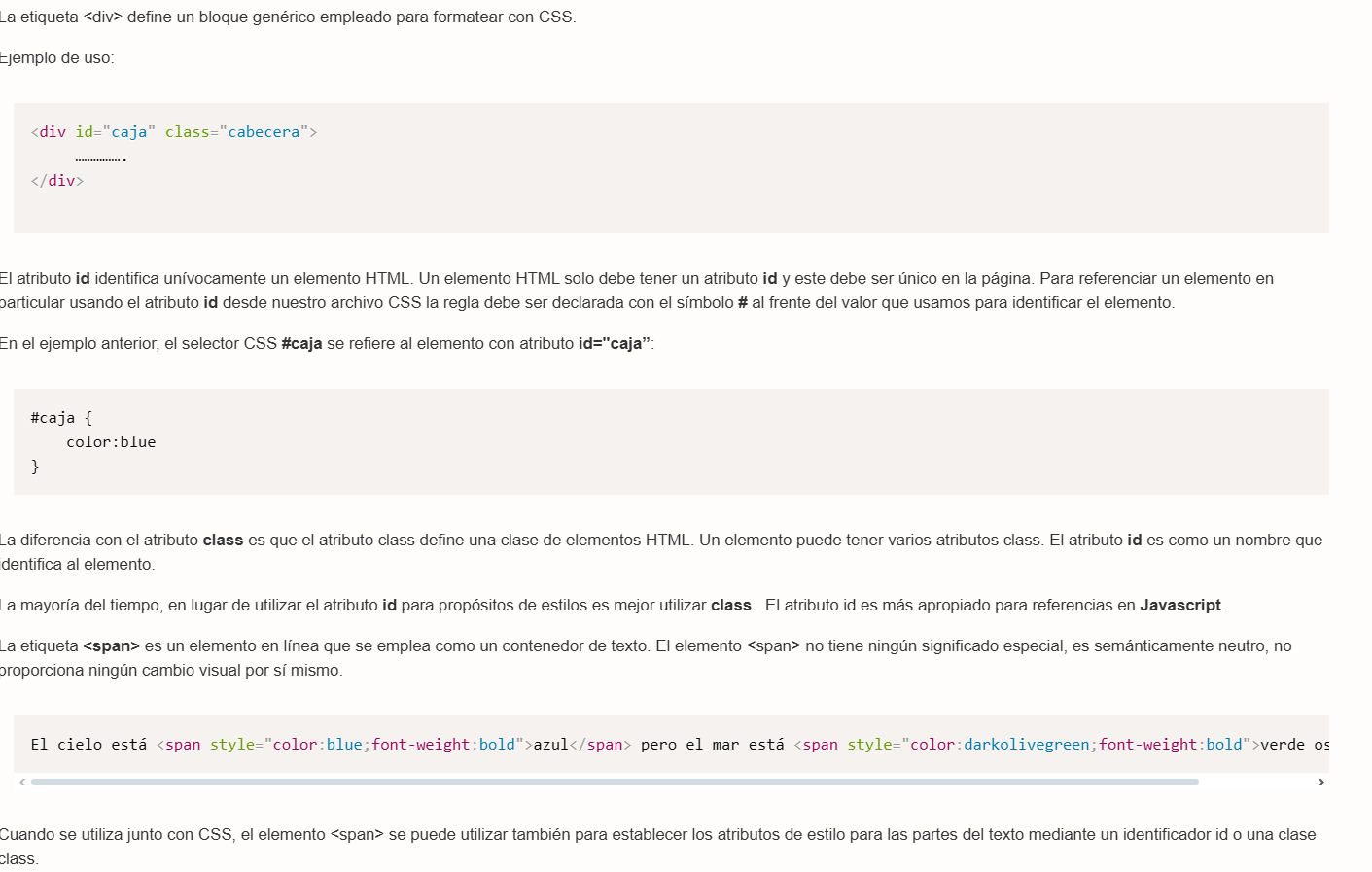






# 

# 4.5. El atributo ID y la etiqueta SPAN



# 4.6. El modelo de cajas

Como hemos visto [**anteriormente**](https://iesdavid.github.io/LM/UA04/434_modelos_de_caja.html), todo en CSS tiene una caja a su alrededor, y entender estas cajas es clave para ser capaz de crear diseños más complejos con CSS, o para alinear elementos con otros elementos.

En CSS existen varios **tipos de cajas** que generalmente encajan en las categorías de **cajas (block)** y **cajas en línea (inline)**. El tipo se refiere a cómo se comporta la caja en términos de flujo de página y en relación con otras cajas en la página. Las cajas tienen un tipo de **visualización interior** (inner) y un tipo de **visualización exterior** (outer).

En general, puede establecer varios valores para el tipo de visualización utilizando la propiedad **display**.

# 4.6. Mejoras de CSS3

* **border-radius**. Genera esquinas redondeadas para la caja formada por el elemento.
* **box-shadow**. Crea sombras para la caja formada por el elemento.
* **text-shadow** Esta propiedad es similar a box-shadow pero específica para textos.
* **@font-face** Esta regla nos permite cargar y usar cualquier fuente que necesitemos.
* **linear-gradient** (posición inicio, color inicial, color final) Esta función puede ser aplicada a las propiedades background o background-image para generar un gradiente lineal.
* **radial-gradient** (posición inicio, forma, color inicial, color final) Esta función puede ser aplicada a las propiedades background o background-image para generar un gradiente radial.
* **rgba()** Esta función es una mejora de rgb().
* **hsla()** Esta función es una mejora de hsl().
* **outline** Esta propiedad fue mejorada con la incorporación de otra propiedad llamada **outline-offset**. Ambas propiedades combinadas generan un segundo borde alejado del borde original del elemento (por ejemplo, **outline**: 1px solid #000000; outline-offset: 10px;).
* **border-image** Esta propiedad crea un borde con una imagen personalizada.
* **transform** Esta propiedad modifica la forma de un elemento.
* **translate** Mueve el objeto tantos pixeles como sean especificados por sus parámetros
* **transition** Esta propiedad puede ser aplicada para crear una transición entre dos estados de un elemento.

UA04.2

### **1. Personalización de estilos**

CSS (cascading style sheets, 'hojas de estilo en cascada') es el lenguaje utilizado para deﬁnir el modo de presentar el código HTML de una página web. Los temas que WordPress pone a disposición de la persona usuaria constituyen, fundamentalmente, una colección de archivos con código CSS, que permiten modiﬁcar el aspecto de la estructura HTML de la página.

La utilización de CSS es muy práctica, ya que permite separar la estructura de un documento (HTML) de su diseño o presentación (CSS). De esta manera, un solo archivo CSS puede modiﬁcar el aspecto de varios documentos HTML, de modo que, en caso de realizar cambios de diseño general, solo sería necesario modiﬁcar dicho archivo y no cada uno de los documentos HTML, con las ventajas de eﬁciencia y tiempo que ello supone.

#### **1.1. Colocación del código CSS**

Existen tres formas distintas de vincular código CSS a una estructura HTML, que se muestran en la siguiente ﬁgura:

  
  
➜ En un atributo HTML, con el atributo style  
El atributo style se puede incluir en cualquier etiqueta, y permite insertar directamente código CSS como valor de dicho atributo. A su vez, las diferentes instrucciones incluidas se separarán con punto y coma (;).

Excepto en aquellos casos en que se deseen realizar pruebas puntuales, no es aconsejable incluir el código CSS de esta manera. El principal motivo por el que se desaconseja tiene que ver con la repetición innecesaria de código, ya que, para establecer, por ejemplo, un aspecto común a varios párrafos, se debería copiar el atributo style con los mismos valores en todos ellos. Por otro lado, si en el futuro se quisiera modiﬁcar una propiedad, como el color del texto, se tendría que cambiar en todos ellos.

➜ Con la etiqueta <style>  
<style> permite establecer un diseño general que afectará a todo el documento HTML en el que se encuentre dicha etiqueta. Esta se suele incluir en la cabecera del documento, es decir, dentro del elemento <head>.

Aunque es mejor que usar el atributo style, ya que permite la afectación a todo el documento, tampoco es la opción más recomendable, ya que el código CSS incluido aquí solo afectará al documento actual, y no se podrá reutilizar en otros sin haberse duplicado previamente.

➜ En un archivo independiente  
La forma más correcta de incluir código CSS es mediante un archivo externo. Este tendrá una extensión .css y se situará, por lo general, en un directorio diferente al de los documentos HTML.

Para incluir un documento CSS, es preciso insertar la siguiente etiqueta en la cabecera de los documentos HTML a los que se quiera aplicar la hoja de estilo correspondiente.



#### 

#### **1.2. Selectores**

Un selector deﬁne los elementos sobre los que se van a aplicar un conjunto de reglas CSS agrupadas entre llaves ({ }).

En los archivos CSS, se establecen una serie de selectores que afectarán a determinados elementos del documento HTML, y que contendrán una serie de reglas o propiedades con las que se modiﬁcará el aspecto de dichos elementos.

El formato para escribirlos se representa:

selector {

propiedad1: valor1;

propiedad2: valor2;

}

Los tipos de selectores básicos que se pueden establecer son los que se indican a continuación:

* **Selectores por tipo.** En este caso, se escribe únicamente el nombre de una etiqueta HTML. Las propiedades modificadas en este selector afectarán a todos los elementos del mismo tipo en el documento HTML, como párrafos ( p) o enlaces (a). Esta es una manera de establecer un estilo por defecto a los elementos de una página.

a{

color: green;

}

* **Selectores de clase**. El nombre del selector, en este caso, debe ir precedido de un punto (.) y representar una clase, añadida a uno o varios elementos HTML, mediante el atributo class. Las propiedades modificadas en el interior afectarán a todos los elementos para los que se haya establecido dicha clase como atributo. En este punto, debe recordarse que el nombre de la clase únicamente irá precedido de punto en el selector CSS, pero nunca en el atributo class.

| **CSS** | **Etiqueta** |
| --- | --- |
| .clase1{  color: blue;  } | <p class="clase1"> Este texto estará de color azul.</p> |

* **Selectores de id.** Este tipo de selectores se corresponden con el nombre del identificador de un elemento, es decir, con el valor de su atributo id. Afectan solamente a un elemento, ya que los identiﬁcadores son únicos y no se pueden repetir. En el selector CSS, el nombre del identiﬁcador debe ir precedido de almohadilla (#).

| **CSS** | **Etiqueta** |
| --- | --- |
| #id2{  color: yellow;  } | <p id="id2"> Este texto estará de color amarillo.</p |

#### **1.3. Colores**

La modiﬁcación del color de un texto que incluye un elemento pasa por utilizar la propiedad color, mientras que para cambiar el color de fondo del elemento en cuestión, la propiedad correspondiente es background-color.

Los valores de un color se pueden deﬁnir en diferentes formatos, la mayoría de los cuales indican las proporciones de los tres colores básicos utilizados para representar toda la gama de colores en las pantallas, esto es, rojo, verde y azul (RGB).

Al hacer una búsqueda rápida en Internet de «colores RGB» o «colores HTML», aparecen los nombres, valores o códigos hexadecimales de los colores que se pueden emplear. Ejemplo: <https://htmlcolorcodes.com/es/>

Algunos de los métodos que permiten establecer un valor cromático son los siguientes:

* Nombre del color: blue, red, orange, green…
* Proporción de cada valor RGB, en código hexadecimal, precedido de almohadilla: #FF0000.
* Proporción de cada valor RGB en decimales, de 0 a 255: rgb(255, 0, 0).
* Proporción de cada valor RGB en porcentajes: rgb(100%, 0%, 0%).
* Proporción de cada valor RGB, más un valor de 0 a 1 (alpha), que indica la transparencia de dicho color: rgba(255, 0, 0, 0.5).

Puedes utilizar tanto la gama de colores RGB como la de CMYK para personalizar tu web, si bien la que mejor actúa en el entorno digital es la primera; se usa en todos los dispositivos capaces de emitir luz.

La gama CMYK se adecúa mejor a cualquier elemento que se vaya a imprimir.

Los diferentes valores de un color en CSS sirven para deﬁnir el color de un texto, el color de fondo de un elemento o de un borde, o, incluso, el color de la sombra que proyecta un elemento cuando se utilizan efectos visuales CSS avanzados.

Si usas un código hexadecimal para el color, este se puede abreviar, especiﬁcando solo una cifra por cada componente RGB, siempre y cuando las parejas de cifras sea idénticas. Por ejemplo, se puede abreviar #FF0033 como #F03.

Existen, además, otros formatos de color que se pueden utilizar, como el CMYK (cian, magenta, amarillo y negro). Para ello, se utiliza el texto cmyk, seguido de los cuatro valores entre paréntesis, bien en porcentaje de intensidad o usando valores numéricos entre 0 y 255. Por ejemplo: cmyk(50%, 100%, 0%, 0%);.

Algunos editores y páginas web ofrecen herramientas para conocer los valores hexadecimales y decimales de un color seleccionado. En Internet, existe una gran oferta de sitios web con este tipo de herramientas (que reciben el nombre de *color pickers*), por lo que bastará con realizar una búsqueda en Internet para localizarlos.

Por cuestiones estéticas, no es recomendable utilizar una gama de colores demasiado amplia; es mejor establecer un número limitado de colores, para diferenciar las secciones del sitio y destacar ciertos contenidos sobre otros.

#### **1.4. Texto**

Además del color, los textos tienen muchas otras propiedades que pueden modiﬁcarse, como el tamaño, el grosor o la tipografía. Cuando se establece una modiﬁcación sobre el texto de un elemento, esta afecta también a sus elementos internos. Por tanto, si se cambia una propiedad en uno más interno, la modiﬁcación llevada a cabo tendrá preferencia sobre el valor del elemento contenedor.

Algunas de las propiedades que modiﬁcan el aspecto de un texto, cuyo preﬁjo es font o text, así como sus posibles valores, son las siguientes:

* Alineación del texto (**text-align**). Define la alineación del texto dentro del elemento. Los valores más comunes son **left** (izquierda), **right** (derecha), **center** (centrada) y **justify** (justificada).
* Subrayado (**text-decoration**). Permite dibujar una línea subrayándola por encima o por debajo, o tachándola. Los valores pueden ser **underline** (subrayado), **none** (quita la propiedad de subrayado si la tuviera, como, por ejemplo, en enlaces), **overline** (línea encima del texto) y **line-through** (tachado).
* Transformación mayúsculas/minúsculas (**text-transform**). Cambia a mayúsculas o minúsculas el texto contenido. Los valores pueden ser **uppercase** (mayúsculas), **lowercase** (minúsculas) y **capitalize** (la primera letra de cada palabra en mayúscula).
* Tamaño del texto (**font-size**). Define el tamaño del texto contenido. Los valores son numéricos, para los que varía la unidad de medida utilizada. Esta puede ser: **px** (píxeles: 12px), **pt** (puntos: 12pt), **%** (porcentaje respecto a cómo se mostraría, siendo 100% ningún cambio), **em** (similar al porcentaje, pero tomando 1 como base: 120% = 1.2em). También se puede emplear como valor un tamaño relativo, como **x-small, small, medium, large o x-large**.
* Tipo de fuente (**font-family**). Establece el tipo de fuente. El valor es una lista con los nombres de las fuentes separados con comas. Los nombres compuestos por más de una palabra se deben poner entre comillas. El navegador busca, entre las fuentes de la lista, la primera que está instalada para aplicarla y, en caso de no encontrar ninguna, aplica la fuente por defecto. Existen nombres de fuentes genéricos, como serif, sans-serif o monospace, que aplican la fuente por defecto del navegador con dicho estilo. Por ejemplo: **font-family: Helvetica, Verdana, “Times New Roman”, serif;.**
* Cursiva (**font-style**). Permite aplicar cursiva al texto. El valor **italic** transforma el texto en cursiva, mientras que normal retira dicha propiedad.
* Grosor del texto (**font-weight**). Establece el grosor del trazo del texto. Aunque existen múltiples valores, muy pocas fuentes soportan diferentes grosores más allá de la negrita y del texto normal. El valor **bold** aplica la negrita al texto y **normal** la elimina.
* Sombreado (**text-shadow**). Dibuja una sombra alrededor del texto. Se especifican cuatro valores separados por espacios. El primer valor corresponde al desplazamiento hacia la derecha de la sombra (hacia la izquierda, si el valor es negativo). El segundo indica el desplazamiento hacia abajo (hacia arriba, si es negativo). El tercero es opcional, e indica el difuminado de la sombra (cuanto mayor sea el valor, más se expandirá el sombreado). Finalmente, el último indica el color de la sombra. Un ejemplo de sombreado desplazado cuatro píxeles de la izquierda, tres hacia abajo, con un difuminado de cinco píxeles y de color gris sería el siguiente: text-shadow: **-4px 3px 5px grey**;.

#### **1.7. Tamaño**

Por defecto, los elementos de bloque ocupan todo el ancho disponible, mientras que la anchura de los elementos de línea varía en función del espacio que ocupe su contenido. En ambos casos, el tamaño vertical siempre dependerá de lo que estos contienen.

La anchura y la altura de un elemento se pueden establecer con propiedades CSS, y este tamaño puede ser relativo al elemento que lo contiene, si se expresa en valores de porcentaje o en absoluto (normalmente, en píxeles).

Algunos elementos en línea, como los enlaces, no se pueden redimensionar, mientras que otros sí, como las imágenes o los botones. Para modiﬁcar la forma en la que el navegador considera un elemento, se pueden establecer las siguientes propiedades:

* **display: block**. El elemento afectado se comporta como un elemento de bloque, como un divisor (<div>) o un párrafo.
* **display: inline**. El elemento se comporta como un elemento en línea. Si se aplica a varios <div> seguidos, estos se situarían uno junto a otro.
* **display: inline-block**. El elemento se establece como un bloque, pero se comporta como un elemento en línea, al estilo de una imagen. Los cambios de tamaño tienen efecto en los elementos con esta propiedad.

El tamaño de un elemento viene deﬁnido por sus propiedades **width** (ancho) y **height** (alto), que pueden tener un valor en **píxeles** (absoluto) o en **porcentaje** (relativo al tamaño del elemento que lo contiene).

Sin embargo, no todos los elementos atienden a la propiedad height cuando esta se mide en porcentaje. Así, una forma de averiguar cuánto ocupa un elemento, con respecto al espacio disponible, pasa por establecer colores de fondo de forma temporal para los diferentes elementos, con el objetivo de visualizar mejor el espacio ocupado por cada uno.

En lugar de establecer un tamaño ﬁjo para un elemento, también existe la posibilidad de asignar un tamaño mínimo y máximo. El tamaño máximo se deﬁne mediante las propiedades max-width y max-height. Su efecto es más visible cuando se aplica a un elemento de bloque, ya que tiende a ocupar todo el ancho disponible por defecto.

El tamaño mínimo de un elemento se puede establecer mediante las propiedades min-width y min-height. Los valores se ﬁjan en píxeles o en porcentajes.

#### **1.8. Bordes**

En un elemento HTML, además de un color de fondo, se puede añadir un borde a su alrededor, el cual no tiene por qué rodearlo completamente. Se puede optar por establecer un borde independiente en la parte superior, inferior, derecha o izquierda. Otra posibilidad es insertar bordes con estilos diferentes en cada una de las zonas del elemento.

Las propiedades CSS que permiten añadir un borde a un elemento son las siguientes:

* **border-width**. Añade grosor al borde de los cuatro lados. Se suele establecer en píxeles (px) y, opcionalmente, de forma independiente a cada lado, con las propiedades border-left-width, border-right-width, border-top-width y border-bottom-width.
* **border-style**. Tipo de trazo con el que se dibuja el borde. Los valores más comunes son solid (línea continua), dashed (línea discontinua), dotted (con puntos) o double (línea continua doble). También se puede especificar el lado de manera individual, como sucede con la propiedad border-width.
* **border-color.** Permite definir el color del borde. Los valores válidos para ello son idénticos a los que se aplican para establecer un color de fondo o de texto. Tal como sucede con las anteriores propiedades, se puede indicar individualmente qué lado se desea modificar.
* **border.** Propiedad genérica que permite utilizar las tres propiedades anteriores a la vez. Para ello, se debe especificar una medida, un estilo y un color separados por un espacio, sin importar el orden. Un ejemplo sería border: 2px solid red; (borde rojo continuo de dos píxeles de grosor). También, como en el caso de las propiedades anteriores, se puede establecer cada lado de manera individual con las propiedades **border-left, border-right, border-top y border-bottom.**

• **border-radius**. Su valor se establece habitualmente en píxeles (px), y hace referencia al radio de curvatura de las esquinas de un elemento. Para aplicar valores diferentes en cada esquina, es necesario establecer cuatro medidas separadas de espacios (la primera se referirá a la esquina superior izquierda y el resto continuarán en el sentido de las agujas del reloj), por ejemplo: border-radius: **5px 8px 5px 8px**;.

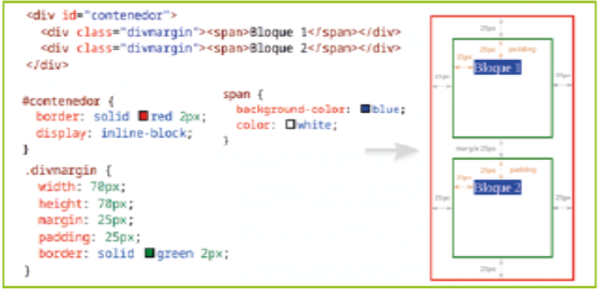


Además del borde, se pueden crear efectos de sombreado alrededor de un elemento. Esto se consigue con la propiedad box-shadow, seguida de cinco valores que definen dicho sombreado: el desplazamiento hacia la derecha, el desplazamiento hacia abajo, el radio de difuminado, el tamaño que crece la sombra respecto al elemento y el color. Por ejemplo, la propiedad box-shadow: -5px 5px 8px 0px grey; indica una sombra desplazada 5 píxeles a la izquierda (negativo), 5 píxeles hacia abajo, un radio de difuminado de 8 píxeles, mismo tamaño que el elemento (0px de diferencia) y de color gris.

#### **1.9. Márgenes**

Al dibujar varios elementos seguidos con contenido y establecer un borde o color de fondo, resulta fácil advertir que, en muchos casos, no existe ningún tipo de espaciado o margen entre dichos elementos, ni entre el borde de un elemento y su contenido. Para establecer un espacio de separación entre un elemento y los otros elementos que lo rodean (o el borde del elemento que lo contiene), o entre el borde del elemento y su contenido, CSS dispone de dos propiedades:

* **margin.** Representa el espacio entre un elemento y los elementos que lo rodean, o el borde del elemento que lo contiene. Como valor, se utiliza una unidad de medida en píxeles (px) o porcentaje (con respecto al espacio total disponible en el elemento contenedor).
* **padding.** Representa el espacio o margen interno entre un elemento y su contenido. De esta manera, es el elemento contenedor el que fuerza que los elementos interiores se despeguen de su borde. Acepta los mismos valores que la propiedad margin.



Ambas propiedades modiﬁcan, por defecto, los márgenes exteriores e interiores de los cuatro lados del elemento. Para modiﬁcar de manera individual cada uno de los lados, se pueden utilizar los suﬁjos -left, -right, -top y -bottom, o bien establecer las cuatro medidas separadas por espacios, las cuales se aplicarían en el siguiente orden: arriba, derecha, abajo e izquierda (por ejemplo, margin: 10px 5px 20px 5px;).

Cabe tener en cuenta que el navegador, por defecto, a la hora de calcular el tamaño que ocupa un elemento, además del tamaño establecido en las propiedades CSS, como width o height, también suma el tamaño del margen interno o *padding* y el grosor del borde al elemento. Por lo tanto, un elemento con un ancho de 200 píxeles, un padding de 20 píxeles y un grosor de borde de 5 píxeles ocupará un ancho total de 250 píxeles (5 píxeles de borde izquierdo, 20 píxeles de padding izquierdo, 200 píxeles de ancho, 20 píxeles de padding derecho y 5 píxeles de borde derecho).

Existe la posibilidad de que las propiedades width y height incluyan el margen interno y el grosor del borde, restándolos al espacio disponible para el contenido. En el ejemplo anterior, esto signiﬁca que el elemento ocuparía, en este caso, 200 píxeles de ancho, dejando 150 píxeles disponibles para el contenido interno. Para ello, se puede utilizar la propiedad CSS box-sizing, aplicando a esta el valor border-box, para incluir el borde y el padding al establecer el tamaño que ocupará el elemento.

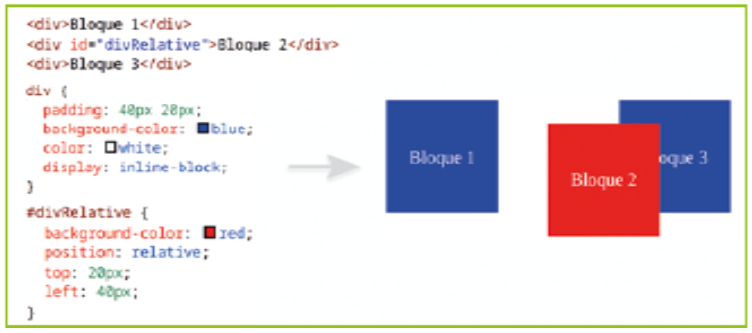
También existe un valor intermedio para la propiedad box-sizing, que sería padding-box. Con esto, incluimos el valor del padding dentro del alto y el ancho que deﬁnimos para el elemento, pero no el borde.

Suma de márgenes.

Cuando dos elementos contiguos tienen márgenes verticales (uno encima del otro), se separan entre sí con un espacio, que depende del elemento que tenga el mayor margen. Si ambos tienen un margen de 25 píxeles, se separarán 25 píxeles en total. Sin embargo, cuando se trata de márgenes horizontales (uno junto al otro), el espacio resultante será la suma de ambos márgenes. Así, en tal caso, se separarían 50 píxeles en total.

#### **1.10. Posicionamiento**

El posicionamiento de los elementos de la página web se realiza de acuerdo con la estructura HTML establecida. Así, el navegador los muestra en el orden determinado por el código: unos dentro de otros, o unos debajo de otros, si son elementos de bloque adyacentes. Sin embargo, este comportamiento se puede modiﬁcar para obligar al navegador a dibujar un elemento en un lugar distinto al original, lo que se consigue con la propiedad de CSS position:

* **position: static**. Es el valor por defecto que, después de cada elemento de bloque, realiza un salto de línea para, a continuación, añadir debajo el siguiente elemento, según el orden establecido en la estructura HTML. Los elementos con este posicionamiento no se ven afectados por las propiedades top, bottom, left y right. Es posible modificar el tamaño ocupado por un elemento añadiéndole márgenes.
* **position: relative.** Un elemento con este valor, en su propiedad position, ocupará el mismo espacio que si su valor fuera static. No obstante, se puede obligar al navegador a desplazar la posición donde lo dibujará (sin que la posición del resto de elementos se vea alterada), utilizando las propiedades top, bottom, left y right, que desplazan el elemento una cantidad de píxeles a partir del lado seleccionado. Así, por ejemplo, top: 10px; desplaza el elemento diez píxeles hacia abajo, es decir, lo aleja diez píxeles de su posición con respecto al lado superior (también es posible utilizar valores negativos para conseguir el efecto contrario).
* **position: absolute**. Al aplicar este valor, el elemento se sitúa en la esquina superior izquierda del elemento que lo contiene. Además, hay que tener en cuenta que se «despega» del documento, es decir, que, para el resto de elementos, la posición se interpreta como si no estuviera ocupando espacio en el documento, de forma similar a si se situara en una capa superior. Esto hará que, en ocasiones, los elementos con posicionamiento absoluto se superpongan a otros elementos.
* **margin-top y margin-left**. Permiten mover un elemento. Sin embargo, si se utilizan las propiedades top, left, right o bottom, dicho elemento deja de situarse con respecto al que lo contiene y emplea todo el documento HTML como referencia. En otras palabras, si se asignan los valores left: 0px; bottom: 20px;, el elemento se situará pegado al lado izquierdo de la página y a veinte píxeles de distancia de la parte inferior del documento.
* **position: fixed**. Este valor es muy similar a absolute, con la excepción de que utiliza la ventana, y no el documento, como referencia para posicionarse. De esta forma, el elemento siempre estará visible (incluso haciendo scroll sobre el documento), lo que resulta muy útil, por ejemplo, para crear barras de navegación que sean fácilmente accesibles.

#### **1.12. Imagen de fondo**

Para establecer una imagen de fondo en un elemento, se usan estas propiedades:

* **background-image.** Su valor será la ruta a la imagen desde el directorio donde se encuentra el archivo CSS. Esta debe incluirse entre comillas simples y paréntesis, precedida de la palabra url, así: background-image: url('../img.png');.
* **background-position**. En el caso de que la imagen sea más pequeña que el elemento, esta propiedad establece su alineación vertical y horizontal. Se disponen dos valores separados por espacio: la alineación horizontal, que puede ser left (izquierda), center (centro) o right (derecha), y la vertical, top (arriba), center (centro) o bottom (abajo).
* **background-repeat**. Si la imagen es más pequeña que el elemento, se repetirá, por defecto, horizontal y verticalmente hasta ocupar todo el espacio disponible. Este comportamiento se puede modificar con los valores no-repeat (no se repite), repeat-x (se repite horizontalmente) o repeat-y (se repite verticalmente).

#### **1.16. FlexBox**

Flexbox es un módulo de diseño que se introdujo el 23 de julio de 2009. Es compatible con todos los navegadores web.

Anteriormente, se usaba la propiedad float para crear diseños haciendo flotar elementos a la izquierda o a la derecha ([puedes probarlo aquí](https://developer.mozilla.org/es/docs/Web/CSS/float)).

Flexbox permite crear diseños alineando elementos a un solo eje. El eje puede ser horizontal o vertical. Se utiliza mejor para distribuir el espacio para los elementos en el mismo eje.

1. Crea un contenedor principal .flex-container, con la característica display: flex;
2. Una vez creado el contenedor primario, los elementos dentro del contenedor primario se convertirán en elementos secundarios, .flex-item
3. Los elementos flexibles se organizan en filas de forma predeterminada, pero puede cambiar esto para organizar los elementos en columnas.

La propiedad display se utiliza para establecer uno de estos dos valores: flex y inline-flex.

* Si establece display en flex, todos los elementos secundarios dentro del contenedor se convierten en un elemento flexible. Se organizarán automáticamente en una fila y cada elemento tendrá el mismo ancho que su contenido. El contenedor flexible abarca su propio ancho y cada hijo tiene que caber en el ancho del contenedor.
* Si estableces display en inline-flex, se consigue que el contenedor flexible se comporte como un elemento.

flex-direction puede cambiar la dirección de horizontal o vertical. Tiene cuatro valores que puedes usar:

1. row es el valor predeterminado en el que flex-direction se establece.
2. row-reverse
3. column Cambia la dirección verticalmente.
4. column-reverse

Por razones de accesibilidad, no debe invertir filas o columnas solo para reordenar el contenido. El HTML no ha cambiado, la dirección visual del contenido ha cambiado.

Para ajustar el tamaño de los elementos flexibles, utilizará los tres valores que aparecen en la propiedad flex

1. grow -> Esto determinará cómo se expandirán los elementos para llenar el espacio adicional (si lo hay) en el contenedor flexible.
2. shrink -> Esto determinará cómo se encogen los elementos cuando no hay suficiente espacio en el contenedor flexible.
3. basis -> establecerá el tamaño inicial de los elementos flexibles.

Úselo exactamente en ese orden. Esto se usa solo en artículos flexibles. flex: grow shrink basis

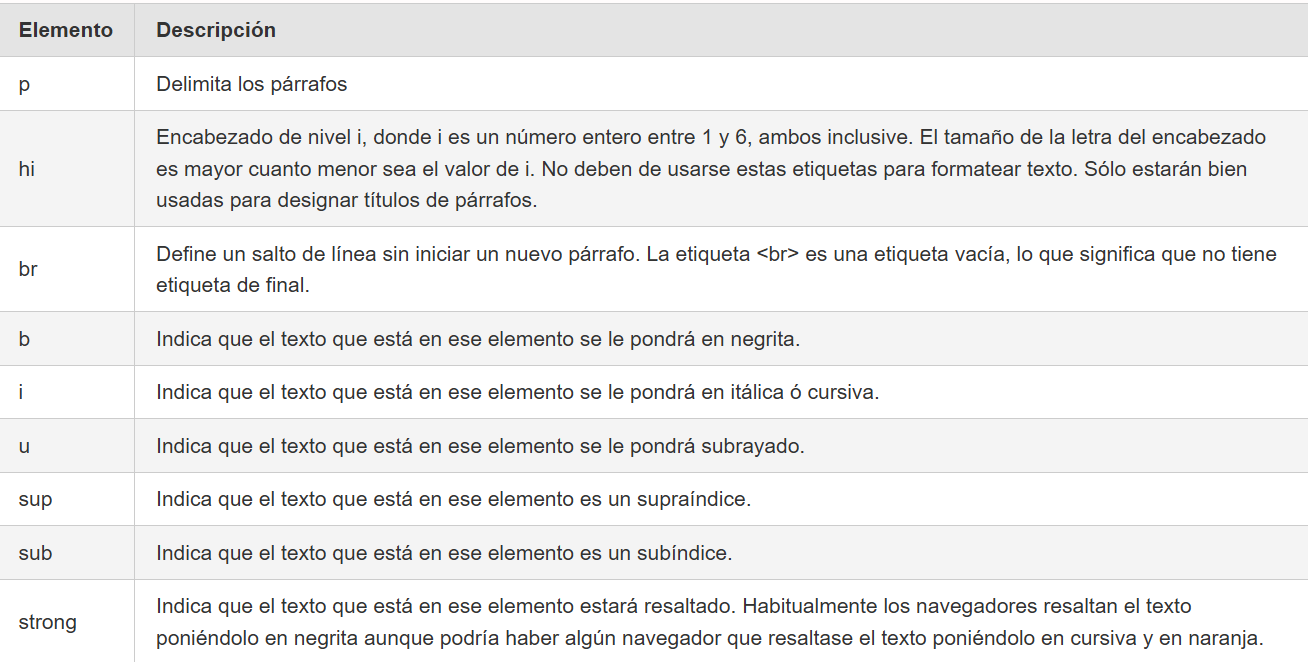
Más información: [Cajas flexibles (Flexbox). CSS. Páginas web HTML y hojas de estilo CSS. Bartolomé Sintes Marco. www.mclibre.org](https://www.mclibre.org/consultar/htmlcss/css/css-flexbox.html)

# COSITAS (HTML)

# ESTRUCTURA:



# FORMATOS:



# FORMULARIOS:

Es, en efecto, **la única forma de recibir** de vuelta la **información** que proviene directamente del usuario final, **estructurada**, además, según las necesidades del diseñador.

La etiqueta **<form>** ... **</form>** tiene como única función la de declarar al navegador que debe incluir un formulario. Esta etiqueta englobará los elementos o campos de formulario, como por  
ejemplo: casillas de texto, casillas de marcar, listas desplegables, etc.



<p>

<label>Nombre:</label>

<input type="text" name="nombre" placeholder="" >

</p>

<p>

<label>Correo electrónico</label>

<input type="email" name="correo" placeholder="">

</p>

<p>

<textarea rows="4" cols="40" >Mensaje: </textarea>

</p>

<p>

<input type="button" value="Enviar" onclick="" id="botones">

</p>

Los atributos usuales de la etiqueta

son:

* **name**: Para asignar un nombre (name) al formulario.
* **action**: Se especificará aquí el destino donde será enviada la información del formulario. El destino normalmente es una llamada a otra página, donde se procesarán los datos, alojada en un servidor. Por ejemplo: action = "http://www.servidor/procesamiento.php". También puede ser a sí mismo dejando **action** vacío.
* **target**: Al igual que en los enlaces, **target** hace que el destino se muestre en la **misma página**, una **nueva** o un **frame** (ver [**enlace**](https://iesdavid.github.io/LM/UA03/324_enlaces_y_marcadores.html) para más información).
* **method**: Especifica el método **HTTP** ( **GET** o **POST** ) que se utilizará al enviar los datos del formulario. El método predeterminado al enviar datos del formulario es **GET**, sin embargo, cuando se utiliza, los datos del formulario enviado serán visibles.
* **enctype**: Especifica en qué formato informático se transmitirán los datos del formulario. El valor por defecto es **application/x-www-form-urlencoded**. Para enviar archivos, este campo debe valer **multipart/form-data** y, por último, para el envío de datos a una dirección de correo electrónico mediante el protocolo **mailto**, el valor será **text/plain.**

## Fieldset

Las etiquetas <**fieldset**> y <**legend**> se utilizan para agrupar elementos de formulario relacionados y proporcionar un título o leyenda descriptiva para el grupo.

La etiqueta <**fieldset**> actúa como un contenedor que agrupa campos del formulario. Ejemplo:

<fieldset>

<legend>Información de contacto</legend>

<label for="name">Nombre:</label>

<input type="text" id="name"><br>

<label for="email">Email:</label>

<input type="email" id="email"><br>

<label for="message">Mensaje:</label>

<textarea id="message"></textarea>

</fieldset>

## Label

La etiqueta <**label**> ayuda a asociar una descripción de texto con un campo de entrada de formulario, lo que lo hace más accesible y fácil de usar. Ejemplo:

<label for="username">Nombre de usuario:</label>

<input type="text" id="username">

## Input

La etiqueta <**input**> en HTML se utiliza para crear campos de entrada en un formulario que permiten a los usuarios ingresar datos. El tipo de dato que se ingresa depende del valor del atributo **type** de la etiqueta <**input**>. Veamos algunos ejemplos comunes de cómo se utiliza la etiqueta <**input**> en un formulario:

**Campo de Texto:** Permite a los usuarios ingresar texto, como su nombre o dirección de correo electrónico.

<input type="text" id="nombre" name="nombre" placeholder="Tu nombre">

**Campo de Contraseña:** Similar al campo de texto, pero los caracteres ingresados se ocultan por asteriscos o puntos para proteger la privacidad de la contraseña.

<input type="password" id="contrasena" name="contrasena" placeholder="Contraseña">



**Casilla de Verificación (Checkbox):** Permite a los usuarios seleccionar o deseleccionar una opción.

<input type="checkbox" id="suscripcion" name="suscripcion" value="aceptar">

<label for="suscripcion">Acepto los términos y condiciones</label>



**Botón de Radio (Radio Button):** Permite a los usuarios seleccionar una opción de un conjunto de opciones mutuamente excluyentes.

<input type="radio" id="opcion1" name="opcion" value="opcion1">

<label for="opcion1">Opción 1</label><br>

<input type="radio" id="opcion2" name="opcion" value="opcion2">

<label for="opcion2">Opción 2</label>



**Botón (Button):** Crea un botón que puede utilizarse para enviar un formulario o realizar otras acciones mediante JavaScript.

<input type="button" value="Enviar" onclick="enviarFormulario()">

**Campo de Correo Electrónico:** Ayuda a validar que se ingrese una dirección de correo electrónico válida.

<input type="email" id="correo" name="correo" placeholder="Correo electrónico">

**Campo de Número:** Permite ingresar valores numéricos y puede incluir restricciones como mínimo y máximo.

<input type="number" id="edad" name="edad" min="18" max="99" placeholder="Edad">

## Button

La etiqueta <**button**> en HTML se utiliza para crear un botón. Puede ser utilizado para activar una acción o comportamiento específico al ser pulsado.

<button>Haz clic</button>

También puedes utilizar el atributo **type** para definir el tipo de botón. Algunos valores comunes para el atributo **type** son:

* **submit**: Un botón que envía un formulario.
* **reset**: Un botón que reinicia los valores iniciales de los campos de un formulario.

<button type="submit">Enviar</button>

## 

## Select

La etiqueta <**select**> en HTML se utiliza para crear un menú desplegable o una lista de opciones seleccionables. Permite al usuario elegir una opción de entre varias disponibles.

Ejemplo básico de cómo utilizar la etiqueta <**select**> junto con la etiqueta <**option**> para crear un menú desplegable:

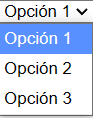
<select>

<option value="option1">Opción 1</option>

<option value="option2">Opción 2</option>

<option value="option3">Opción 3</option>

</select>



Cada <**option**> dentro de la etiqueta <**select**> representa una opción en el menú desplegable. El atributo value se utiliza para especificar el valor asociado a cada opción, que se enviará al servidor cuando se envíe el formulario.

También puedes agregar el atributo selected a una opción específica para que se seleccione automáticamente cuando se muestre el menú desplegable:

<select>

<option value="option1">Opción 1</option>

<option value="option2" selected>Opción 2</option>

<option value="option3">Opción 3</option>

</select>

En este ejemplo, **la opción «Opción 2» estará seleccionada** de forma predeterminada cuando se muestre el menú desplegable.

La etiqueta <**select**> en HTML también admite el atributo multiple, que permite al usuario seleccionar múltiples opciones del menú desplegable.

Al agregar el atributo **multiple** a la etiqueta <**select**>, se permite la selección de varias opciones manteniendo presionada la tecla Ctrl (en Windows) o Cmd (en Mac) mientras se hacen clic en las opciones deseadas. Ejemplo:

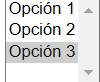
<select multiple>

<option value="option1">Opción 1</option>

<option value="option2">Opción 2</option>

<option value="option3">Opción 3</option>

</select>



La etiqueta <**optgroup**> se utiliza dentro de un elemento <**select**> para agrupar varias opciones relacionadas en una categoría o conjunto. Esto proporciona una organización visual a las opciones dentro de la lista desplegable. La etiqueta <optgroup> debe contener una o más etiquetas <option> que representan las opciones dentro de ese grupo.

<select>

<optgroup label="Frutas">

<option value="manzana">Manzana</option>

<option value="naranja">Naranja</option>

<option value="platano">Plátano</option>

</optgroup>

<optgroup label="Verduras">

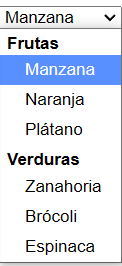
<option value="zanahoria">Zanahoria</option>

<option value="brocoli">Brócoli</option>

<option value="espinaca">Espinaca</option>

</optgroup>

</select>



# Atributos-Formularios

## Name

El atributo name se utiliza para identificar un campo de entrada en particular. Es especialmente útil cuando se procesan los datos del formulario en el servidor, ya que actúa como clave para los valores ingresados.

<input type="text" name="nombre\_usuario" placeholder="Ingrese su nombre">

## Value

El atributo value define el valor predeterminado de un campo de entrada. Este valor se envía al servidor cuando se envía el formulario.

<input type="text" name="nombre" value="John Doe">



Placeholder

El atributo placeholder muestra un texto de sugerencia en el campo de entrada cuando está vacío. Este texto desaparece cuando el usuario comienza a escribir en el campo.

<input type="text" name="correo" placeholder="Ingrese su correo electrónico">



## Required

El atributo required indica que un campo de entrada es obligatorio y debe completarse antes de enviar el formulario.

<input type="text" name="nombre" placeholder="Ingrese su nombre" required>

## Disabled

El atributo disabled desactiva un campo de entrada, lo que significa que los usuarios no pueden interactuar con él ni modificar su valor. Los campos desactivados no se envían con el formulario.

<input type="text" name="nombre" value="John Doe" disabled>

## Readonly

El atributo readonly hace que un campo de entrada sea de solo lectura. A diferencia de los campos desactivados, los campos de solo lectura sí se envían con el formulario, pero los usuarios no pueden modificar su valor.

<input type="text" name="id\_usuario" value="12345" readonly>

## ID

El atributo id asigna un identificador único al elemento . Este identificador se puede utilizar para seleccionar y manipular el elemento mediante JavaScript y para aplicar estilos específicos mediante CSS. Los valores de los atributos id deben ser únicos en todo el documento.

<input type="text" id="nombre\_usuario" name="nombre\_usuario" placeholder="Ingrese su nombre de usuario">

## CLASS

El atributo class permite asignar una o varias clases al elemento . Las clases son útiles para aplicar estilos CSS y manipular elementos que comparten características comunes mediante JavaScript. Puedes asignar múltiples clases separándolas con espacios.

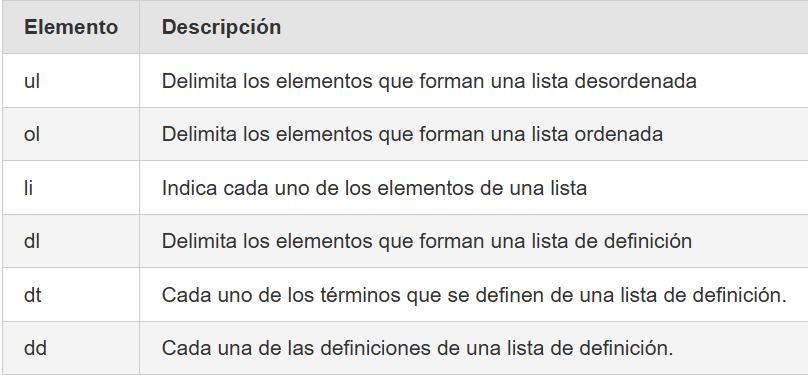
<input type="text" class="formulario-entrada entrada-nombre" name="nombre" placeholder="Ingrese su nombre">

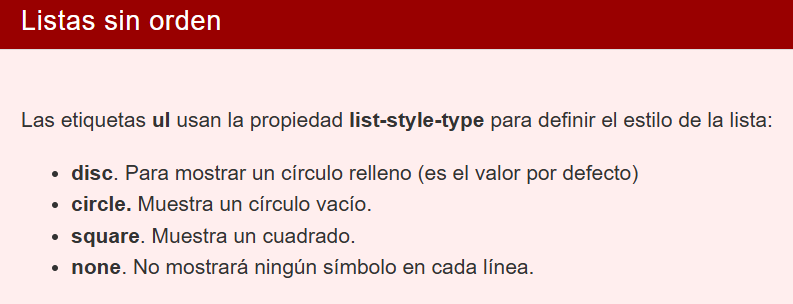
# COMENTARIOS

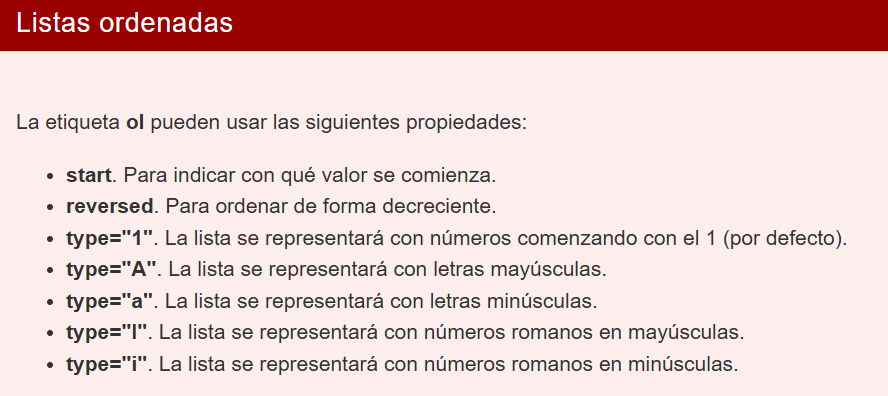
En un documento HTML, los comentarios pueden escribirse entre los caracteres “<!--” y “-->”. Por ejemplo:

**<!-- Esto es un comentario escrito en un documento HTML →**

# LISTAS:







# Enlaces y marcadores

<a href="<https://www.institutonervion.es>" target="\_blank">Visita nuestra página web</a>

El atributo **target** especifica dónde abrir el documento vinculado y pude tener los siguientes valores:

* \_blank - Abre el documento vinculado en una nueva ventana o pestaña
* \_self - Abre el documento vinculado en la misma ventana / pestaña que se hizo clic (esto es el predeterminado)
* \_parent - Abre el documento vinculado en el marco padre
* \_top - Abre el documento vinculado en todo el cuerpo de la ventana
* framename - Abre el documento vinculado en un marco con nombre

# Imágenes:

<a href="https://www.instagram.com/extraterrestrialcharlie/"><img src="../img/gato1.png"></a>

# Audio y video:

# 

# Ejemplos css y html:

### modificar el color de un enlace y hacer el texto mas grande cuando pasas el ratón por encima:

.enlace{

color: white;

text-decoration: none;

padding: 2%;

margin: 100px;

transition: color 1.5s ease, font-size 0.3s ease-in-out;

}

.enlace:hover{

color: blue;

font-size: 23px;

}

### Para cambiar el tipo de letra:

#body{

background-color: #B5F4FF;

font-family:'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;

margin: 20px;

}

#titulo{

font-family: Helvetica;

text-align: center;

}

### Cambiar tamaño y tipo de letra:

#contenido h2{

font-size: 50px;

font-family: Helvetica;

}

### Botones:

* + HTML:

<input type="button" value="Enviar" onclick="" id="b

* + CSS:

#botones{

border-radius: 40%;

border-color: #007bff;

background-color: rgb(126, 255, 255);

}

### pie de página:

<footer class="footer">

<p>© 2024 Mi Página Web. Todos los derechos reservados.</p>

</footer>

### Secciones

<section id="contenido">

<div class="div1">

<h2>Sobre mi</h2>

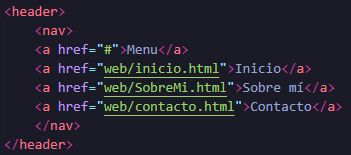
<p>Soy Dislexica</p>

<img src="img/perug.jpg">

</div>

</section>

### Barra de navegación



header{

background-color: #007bff;

border:2px solid black;

border-radius: 10px;

text-align: center;

}

header nav a{

color: white;

text-decoration: none;

padding: 2%;

transition: color 1.5s ease, font-size 0.3s ease-in-out;

}

header nav a:hover{

color: black;

font-size: 23px;

}