



# HOJAS DE ESTILO

## Cascade Style Sheets (CSS)

<b>0.- Introducción.....</b>	<b>4</b>
<b>1.- Aplicar CSS.....</b>	<b>4</b>
1.1.- De forma interna.....	4
1.2.- De forma externa.....	4
1.3.- En línea.....	5
1.4.- Jerarquía de interpretación.....	5
<b>2.- Sintaxis.....</b>	<b>5</b>
<b>3.- Selectores.....</b>	<b>6</b>
3.1.- Simples.....	6
3.1.1.- Universal.....	6
3.1.2.- De elemento.....	6
3.1.3.- De identificación.....	6
3.1.4.- De clase.....	7
3.1.5.- Agrupación.....	7
3.2.- Pseudoclasas.....	7
3.3.- Pseudoelementos.....	10
3.4.- Atributos.....	11
3.4.1.- Selector CSS [atributo].....	11
3.4.2.- Selector CSS [atributo = "valor"].....	11
3.4.3.- Selector CSS [atributo ~= "valor"].....	12
3.4.4.- Selector CSS [atributo  = "valor"].....	12
3.4.5.- Selector CSS [atributo ^= "valor"].....	12
3.4.6.- Selector CSS [atributo \$= "valor"].....	13
3.4.7.- Selector CSS [atributo *= "valor"].....	13
<b>4.- Modelo de caja CSS.....</b>	<b>14</b>
4.1.- Ancho y alto de un elemento.....	15
4.2.- Background.....	16
4.3.- Border.....	17
4.4.- Margin.....	18
4.5.- Padding.....	19
4.6.- Outline.....	21
<b>5.- Aplicación de CSS a:.....</b>	<b>22</b>
5.1.- Text.....	22
5.2.- List.....	23
5.3.- Botones.....	23



5.4.- Tables.....	24
5.3.1.- border-collapse.....	24
5.3.2.- Width y Height.....	25
5.3.3.- text-align.....	25
5.3.4.- vertical-align.....	25
5.3.5.- padding.....	25
5.3.6.- Divisores horizontales.....	25
5.3.7.- Marcado de fila con selector hover.....	26
5.3.8.- Tablas con estilo cebra.....	26
5.3.9.- Tablas responsiva.....	27
<b>6.- Propiedad display.....</b>	<b>27</b>
6.1.- Crear un menú horizontal con lista.....	28
<b>7.- Propiedad position.....</b>	<b>32</b>
<b>8.- Overflow.....</b>	<b>33</b>
<b>9.- Float.....</b>	<b>33</b>
<b>10.- Gradiente.....</b>	<b>34</b>
10.1.- Gradiente lineal.....	34
10.2.- Gradiente radial.....	35
<b>11.- Cursores.....</b>	<b>36</b>
<b>12.- Tooltips.....</b>	<b>40</b>
<b>13.- Transiciones.....</b>	<b>41</b>
<b>14.- Transformaciones.....</b>	<b>42</b>
15.1.- Escalar.....	43
15.2.- Trasladar.....	43
15.3.- Rotar.....	43
15.4.- Distorsionar.....	43
15.5.- Matrix.....	44
<b>15.- Flex.....</b>	<b>47</b>
15.1.- display: flex   inline-flex.....	48
15.2.- flex-wrap: nowrap   wrap   wrap-reverse.....	48
15.3.- justify-content: flex-start   flex-end   center   space-between  space-around.....	49
15.4.- align-items: stretch   flex-start   flex-end   center   baseline.....	50
15.5.- flex-direction: row   row-reverse   column   column-reverse.....	54
15.6.- flex-grow, flex-shrink y flex-basis.....	55
<b>16.- Grid.....</b>	<b>56</b>
16.1.- Propiedad display: grid   inline-grid.....	57
16.2.- Propiedades grid-template-columns y grid-template-rows.....	57
16.3.- Propiedades row-gap y column-gap.....	58
<b>Herramientas.....</b>	<b>59</b>
<b>Documentos de interés.....</b>	<b>59</b>
<b>Anexos.....</b>	<b>60</b>



---

Anexo I. Todas las pseudoclases.....	60
Anexo II. Todas las pseudoelementos.....	62
Anexo III. Todas las propiedades background.....	62
Anexo IV. Todas las propiedades de border.....	63
Anexo V. Lista de cursores.....	64



## 0.- Introducción

CSS, que significa Hojas de Estilo en Cascada (Cascading Style Sheets en inglés), es un lenguaje de diseño gráfico utilizado para controlar el aspecto de los elementos escritos en HTML o XML. En términos sencillos, HTML se utiliza para estructurar el contenido, mientras que CSS se utiliza para diseñar el contenido. CSS permite a los desarrolladores cambiar colores, fuentes, espaciado, y la disposición de los elementos en una página web, entre otras cosas. Es una herramienta esencial para cualquier desarrollador web, ya que proporciona la flexibilidad y el control necesarios para crear sitios web atractivos y funcionales.

## 1.- Aplicar CSS

### 1.1.- De forma interna

Haciendo uso de la etiqueta <style>.

```
<html>
  <head>
    <style type="text/css">
      body {background: white;}
      p {color: black;}
      [...]
    </style>
  </head>
  <body>
    [...]
  </body>
</html>
```

### 1.2.- De forma externa

Usando la etiqueta <link> dentro del <head>.

```
<html>
  <head>
    <link href="mi_estilo.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    [...]
  </body>
</html>
```



### 1.3.- En línea

Usar el atributo style del elemento y aplicar estilo.

```
<h1 style="color: red; background: black;">...</h1>
```

### 1.4.- Jerarquía de interpretación

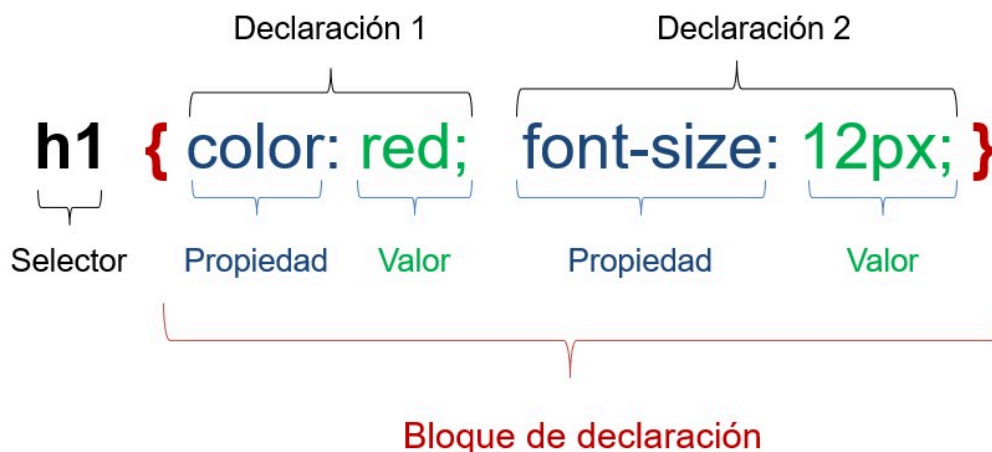
Prevalece el **estilo en línea** (style en el elemento) sobre una hoja de **estilo interna** (<style> en el <head>), que a su vez prevalece sobre una declaración de **estilo externa** (<link> en el <head>). Es decir:

1. Estilo en línea (dentro de un elemento HTML)
2. Hojas de estilo externas e internas (en la sección principal)
3. Navegador predeterminado

Por lo tanto, un estilo en línea tiene la máxima prioridad y anulará los estilos externos e internos y los valores predeterminados del navegador.

## 2.- Sintaxis

Debemos familiarizarnos con la “jerga” de CSS. Así los elementos de la sintaxis se les nombra como:





## 3.- Selectores

### 3.1.- Simples

#### 3.1.1.- Universal

El selector universal (\*) selecciona todos los elementos HTML de la página. Con lo cual el estilo aplicado afectará a todos los elementos.

```
* {  
    text-align: left;  
    color: lightblue;  
}
```

#### 3.1.2.- De elemento

El selector de elementos selecciona elementos HTML según el nombre del elemento. EN el siguiente ejemplo todos los <h1> estarán centrados y serán de color verde oscuro.

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

#### 3.1.3.- De identificación

El selector de identificación utiliza el atributo de identificación de un elemento HTML para seleccionar un elemento específico. La identificación de un elemento **es única** dentro de una página, por lo que el selector de identificación se usa para seleccionar un elemento único. Para seleccionar un elemento con una identificación específica, escriba un carácter **almohadilla** (#), seguido de la identificación del elemento.

Si tengo el código HTML de un div con id “para1” y le doy un estilo a ese id

```
<div id="para1"> <p>Hola</p> </div>
```

El código CSS sería:

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



### 3.1.4.- De clase

El selector de clases selecciona elementos HTML con un atributo de clase específico. Para seleccionar elementos con una clase específica, escriba un carácter de **punto** (.), seguido del nombre de la clase.

Si tengo el código HTML de un div con class “center” y le doy un estilo a ese id

```
<div class="center"> <p>Hola</p> </div>
```

El código CSS sería:

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

### 3.1.5.- Agrupación

Cuando hay elementos que tienen la misma definición de estilos la podemos agrupar:

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

## 3.2.- Pseudoclasas

Se utiliza para definir un estado especial de un elemento.

Por ejemplo, se puede utilizar para:

- Aplicar estilo a un elemento cuando un usuario pasa el mouse sobre él.
- Diseñar los enlaces visitados y no visitados de manera diferente.
- Aplicar estilo a un elemento cuando recibe el foco.

Tiene la sintaxis:

```
selector:pseudo-class {  
    property: value;  
}
```

Nota que después del selector solo lleva dos puntos (:).



Algunas de las pseudoclases son:

- link.
- visited.
- hover.
- active
- lang
- focus

Puedes ver todas las pseudoclases en el [Anexo I](#).

Ejemplo: el cursor sobre un elemento <div> para mostrar un elemento <p>.

```
p {
    display: none;
    background-color: yellow;
    padding: 20px;
}

div:hover p {
    display: block;
}
```

**Haga coincidir el primer elemento <i> en todos los elementos <p>.**

Tengo el HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p i:first-child{
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
    <p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
  </body>
</html>
```





Resultado:

I am a *strong* person. I am a *strong* person.  
I am a *strong* person. I am a *strong* person.

Como ves solo se aplica al primer “i” de todos los párrafos.

**Haga coincidir todos los elementos <i> en todos los primeros elementos secundarios <p>**

Tengo el HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p:first-child i {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>
    <p>I am a <i>strong</i> person. I am a <i>strong</i> person.</p>

    <div>
      <p>I am a <i>strong</i> person. I am a <i>strong</i>
      person.</p>
      <p>I am a <i>strong</i> person. I am a <i>strong</i>
      person.</p>
    </div>
  </body>
</html>
```

Resultado:

I am a *strong* person. I am a *strong* person.  
I am a *strong* person. I am a *strong* person.  
I am a *strong* person. I am a *strong* person.  
I am a *strong* person. I am a *strong* person.

Como ver sólo se aplica todos los “i” del primer “p” encontrado.



### 3.3.- Pseudoelementos

Un pseudoelemento CSS se utiliza para diseñar partes específicas de un elemento.

Por ejemplo, se puede utilizar para:

- Aplicar estilo a la primera letra o línea de un elemento
- Insertar contenido antes o después del contenido de un elemento

Su sintaxis es:

```
selector::pseudo-element {  
    property: value;  
}
```

Nota que después del selector solo lleva dos puntos (:) dos veces.

Alguno de los pseudoelementos son:

- before
- after
- selection

Puedes ver la lista completa en el [Anexo II](#).

Ejemplo: el código HTML

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      ::selection {  
        color: red;  
        background: yellow;  
      }  
    </style>  
  </head>  
  <body>  
    <h1>Select some text on this page:</h1>  
    <p>This is a paragraph.</p>  
    <div>This is some text in a div element.</div>  
  </body>  
</html>
```

Cuando hagamos una selección del texto este aparecerá de color rojo con fondo amarillo.



## 3.4.- Atributos

Es posible diseñar elementos HTML que tengan atributos o valores de atributos específicos.

### 3.4.1.- Selector CSS [atributo]

El selector [attribute] se utiliza para seleccionar elementos con un atributo específico.

Por ejemplo con:

```
<style>
  a[target] {
    background-color: yellow;
  }
</style>
```

Daremos estilo a elementos “a” que tengan el atributo “target”:

```
<a href="http://www.disney.com" target="_blank">disney.com</a>
```

El resultado sería: **disney.com**

### 3.4.2.- Selector CSS [atributo = "valor"]

El selector [attribute="value"] se utiliza para seleccionar elementos con un atributo y **valor específicos**.

Por ejemplo con:

```
<style>
  a[target="_blank"] {
    background-color: yellow;
  }
</style>
```

Daremos estilo a elementos “a” que tengan el atributo “target” con el valor “\_blank”:

```
<a href="http://www.disney.com" target="_blank">disney.com</a>
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
```

El resultado sería: **disney.com** wikipedia.org



### 3.4.3.- Selector CSS [atributo ~= "valor"]

El selector [attribute~="value"] se utiliza para seleccionar elementos con un valor de atributo que **contiene una palabra** específica.

```
<style>
  [title~="flower"] {
    border: 5px solid yellow;
  }
</style>
```

El ejemplo anterior coincidirá con elementos con título="flor", título="flor de verano" y título="flor nueva", pero no con título="mi-flor" o título="flores".

### 3.4.4.- Selector CSS [atributo |= "valor"]

El selector [attribute|= "value"] se utiliza para seleccionar elementos con el atributo especificado, cuyo valor **puede ser exactamente** el valor especificado, **o el valor especificado seguido de un guión (-)**.

Por ejemplo:

```
<style>
  [class|= "top"] {
    background: yellow;
  }
</style>
```

Y el HTML:

```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
```

Sólo dará estilo al h1 y al primer "p".

### 3.4.5.- Selector CSS [atributo ^= "valor"]

El selector [attribute^="value"] se utiliza para seleccionar elementos con el atributo especificado, cuyo **valor comienza con el valor especificado**.

El siguiente ejemplo selecciona todos los elementos con un valor de atributo de clase que comienza con "top":

Nota: ¡ El valor no tiene que ser una palabra completa!



Por ejemplo:

```
<style>
    [class^="top"] {
        background: yellow;
    }
</style>
```

Dará estilo a todos los siguientes:

```
<h1 class="top-header">Welcome</h1>
<p class="top-text">Hello world!</p>
<p class="topcontent">Are you learning CSS?</p>
```

### 3.4.6.- Selector CSS [atributo \$= "valor"]

El selector [attribute\$="value"] se utiliza para seleccionar elementos cuyo valor de **atributo termina** con un valor específico.

Ejemplo:

```
<style>
    [class$="test"] {
        background: yellow;
    }
</style>
```

Los HTML

```
<div class="first_test">The first div element.</div>
<div class="second">The second div element.</div>
<div class="my-test">The third div element.</div>
<p class="mytest">This is some text in a paragraph.</p>
```

El segundo div no tendrá estilo:

The first div element.

The second div element.

The third div element.

This is some text in a paragraph.

### 3.4.7.- Selector CSS [atributo \*= "valor"]

El selector [attribute\*="value"] se utiliza para seleccionar elementos cuyo valor de **atributo contiene** un valor específico.

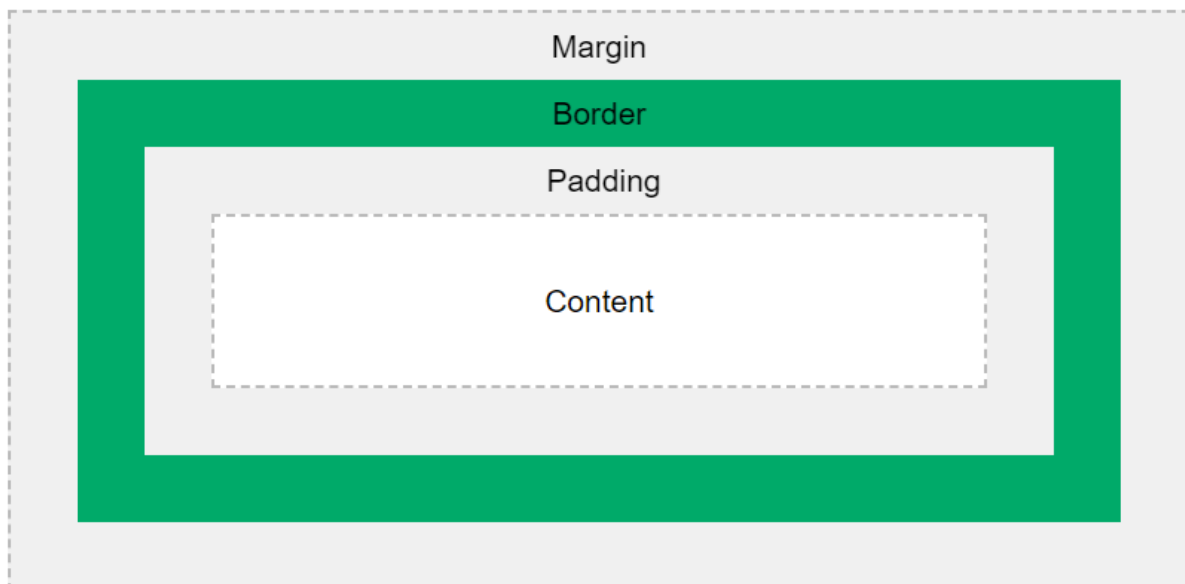


El siguiente ejemplo selecciona todos los elementos con un valor de atributo de clase que contiene "te":

Nota: ¡ El valor no tiene que ser una palabra completa!

## 4.- Modelo de caja CSS

El modelo de cuadro CSS es esencialmente un cuadro que envuelve **cada elemento HTML**. Consta de: contenido, relleno, bordes y márgenes. La siguiente imagen ilustra el modelo de caja:



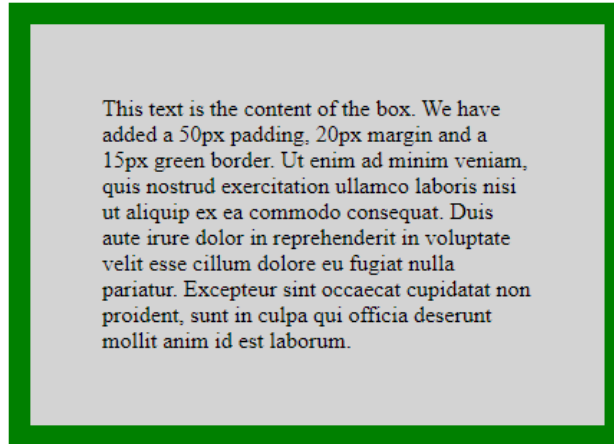
- **Contenido** : el contenido del cuadro, donde aparecen el texto y las imágenes.
- **Relleno** : borra un área alrededor del contenido. El acolchado es transparente.
- **Borde** : un borde que rodea el relleno y el contenido.
- **Margen** : borra un área fuera del borde. El margen es transparente.

Ejemplo:

```
<style>
  div {
    background-color: lightgrey;
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
  }
</style>
```



Aplicaría:



### 4.1- Ancho y alto de un elemento

Cuando configuras las propiedades de ancho y alto de un elemento con CSS, simplemente estableces el ancho y el alto del área de contenido. Para calcular el ancho y alto total de un elemento, también debes incluir el relleno y los bordes.

Por ejemplo, si tenemos:

```
div {  
  width: 320px;  
  height: 50px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

El **ancho total** se calcula como:

320px (ancho del área de contenido)  
+ 20px (relleno izquierdo + relleno derecho)  
+ 10px (borde izquierdo + borde derecho)  
= 350px (ancho total)

Y el **alto total** se calcula como:

50px (alto del área de contenido)  
+ 20px (relleno superior + relleno inferior)  
+ 10px (borde superior + borde inferior)



= 80px (alto total)

Las propiedades **height** y **width** pueden tener los siguientes valores:

- auto- Esto es lo predeterminado. El navegador calcula el alto y el ancho.
- length- Define el alto/ancho en px, cm, etc.
- %- Define la altura/ancho en porcentaje del bloque contenedor.
- initial- Establece la altura/ancho a su valor predeterminado
- inherit- La altura/ancho se heredará de su valor principal.

La propiedad **max-width** se utiliza para establecer el ancho máximo de un elemento.

## 4.2.- Background

Las propiedades de fondo de CSS se utilizan para agregar efectos de **fondo a los elementos**.

Aplicable a	Descripción
background-color	Las propiedades de fondo de CSS se utilizan para agregar efectos de fondo a los elementos.
background-image	La background-image especifica una <b>imagen</b> para usar como fondo de un elemento. De forma predeterminada, la imagen se repite para cubrir todo el elemento.
background-repeat	Usado para la repetición de <b>imágenes</b> . Valor = repeat-x => Repite sólo horizontalmente Valor = no-repeat => Muestra imagen de fondo sólo una vez.
background-position	Se utiliza para especificar la posición de la <b>imagen</b> de fondo.
background-attachment	Especifica si la <b>imagen</b> de fondo debe desplazarse o ser fija (no se desplazará con el resto de la página). Los valores pueden ser: fixed => Hará la imagen de fondo fija. scroll => La imagen de fondo debe desplazarse con el resto de la página

Puedes ver el resto de propiedades Background en el Anexo III.

Propiedad abreviada:





```
body {  
  background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

El orden es:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

### 4.3.- Border

Las propiedades del borde CSS le permiten especificar el estilo, ancho y color del borde de un elemento.

Propiedad específica	Valores
<b>border-style</b> <i>especifica qué tipo de borde mostrar</i>	<b>dotted</b> - Define un borde de puntos <b>dashed</b> - Define un borde discontinuo <b>solid</b> - Define un borde sólido <b>double</b> - Define un doble borde. <b>groove</b> - Define un borde ranurado 3D. El efecto depende del valor del color del borde. <b>ridge</b> - Define un borde estriado 3D. El efecto depende del valor del color del borde. <b>inset</b> - Define un borde insertado 3D. El efecto depende del valor del color del borde. <b>outset</b> - Define un borde inicial 3D. El efecto depende del valor del color del borde. <b>none</b> - No define fronteras <b>hidden</b> - Define un borde oculto  <b>NOTA:</b> puede tener de uno a cuatro valores (para el borde superior, el borde derecho, el borde inferior y el borde izquierdo).  <code>border-style: dotted dashed solid double;</code>
<b>border-width</b> <i>especifica el ancho de los cuatro bordes</i>	Puede tener tamaño específico (en px, pt, cm, em, etc.) o predefinido: thick y medium (por defecto es grueso).  <b>NOTA:</b> puede tener de uno a cuatro valores (para el borde superior, el borde derecho, el borde inferior y el borde izquierdo).



	<pre>border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */</pre> <p>O con dos</p> <pre>border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */</pre>
<b>border-color</b>  <i>Establece el color de los cuatro bordes.</i>	<p><b>NOTA:</b> puede tener de uno a cuatro valores (para el borde superior, el borde derecho, el borde inferior y el borde izquierdo).</p> <pre>border-color: red green blue yellow; /* red top, green right, blue bottom and yellow left */</pre>
<b>border-radius</b>  <i>Agrega bordes redondeados a un elemento</i>	<p>Ejemplo</p> <pre>border-radius: 8px;</pre> <p>Se pueden especificar el borde y lo que queremos modificar. Véase <a href="#">Anexo IV</a> para más detalle:</p>

## 4.4.- Margin

Se utilizan para crear espacio alrededor de los elementos, fuera de los bordes definidos.

Puedo especificar el margen de cada lado de un elemento:

- margin-top
- margin-right
- margin-bottom
- margin-left

Los valores que pueden tener son:

- auto - el navegador calcula el margen
- longitud : especifica un margen en px, pt, cm, etc.
- % : especifica un margen en % del ancho del elemento contenedor
- inherit: especifica que el margen debe heredarse del elemento principal

**NOTA:** se permiten valores negativos



Puede tener	Aplicación
Cuatro valores <i>margen: 25 px 50 px 75 px 100 px;</i>	el margen superior es de 25px el margen derecho es 50px el margen inferior es 75px el margen izquierdo es 100px
Tres valores: <i>margen: 25 px 50 px 75 px;</i>	el margen superior es de 25px los márgenes derecho e izquierdo son 50px el margen inferior es 75px
Dos valores: <i>margen: 25 px 50 px;</i>	Los márgenes superior e inferior son de 25 píxeles. los márgenes derecho e izquierdo son 50px
Un valor <i>margen: 25 px;</i>	los cuatro márgenes son de 25px

Para **auto**, centra horizontalmente el elemento dentro de su contenedor.

Luego, el elemento ocupará el ancho especificado y el espacio restante se dividirá en partes iguales entre los márgenes izquierdo y derecho.

## 4.5.- Padding

Se utiliza para crear espacio alrededor del contenido de un elemento, dentro de los bordes definidos.

Puedo especificar el margen de cada lado de un elemento:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Los valores que pueden tener son:

- longitud : especifica un margen en px, pt, cm, etc.
- % : especifica un margen en % del ancho del elemento contenedor
- inherit: especifica que el margen debe heredarse del elemento principal



Puede tener	Aplicación
Cuatro valores <i>margen: 25 px 50 px 75 px 100 px;</i>	el relleno superior es de 25px el relleno derecho es 50px el relleno inferior es de 75px el relleno izquierdo es 100px
Tres valores: <i>margen: 25 px 50 px 75 px;</i>	el relleno superior es de 25px los rellenos derecho e izquierdo son 50px el relleno inferior es de 75px
Dos valores: <i>margen: 25 px 50 px;</i>	los rellenos superior e inferior son de 25 px los rellenos derecho e izquierdo son 50px
Un valor <i>margen: 25 px;</i>	los cuatro rellenos son de 25px

Caso de uso:

Yo tengo:

```
div {  
  width: 300px;  
  padding: 25px;  
}
```

El elemento div tiene un ancho específico, el relleno agregado a ese elemento se agregará al ancho total del elemento. Este es a menudo un resultado indeseable.

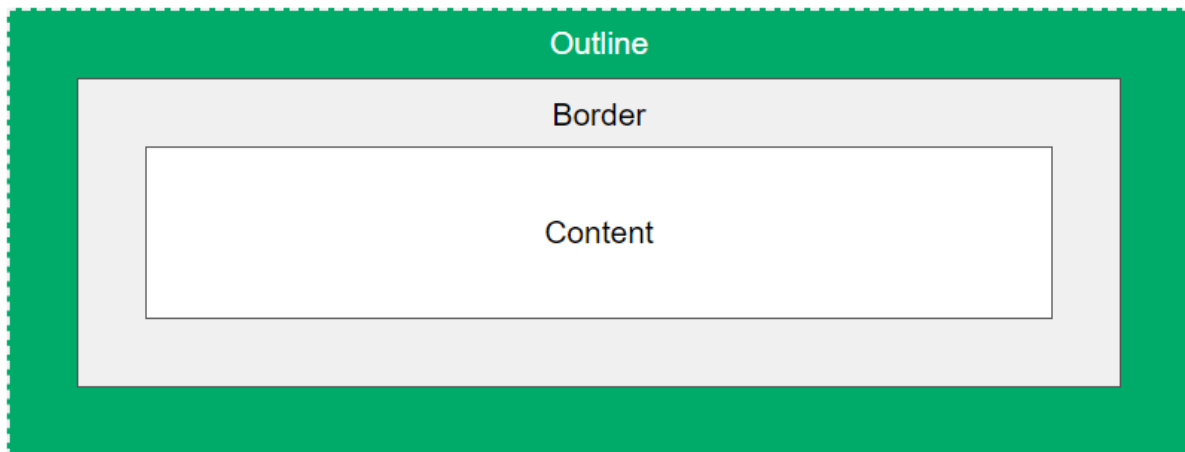
Para mantener el ancho en 300 px, sin importar la cantidad de relleno, puedes usar la propiedad box-sizing. Esto hace que el elemento mantenga su ancho real; Si aumenta el relleno, el espacio de contenido disponible disminuirá.

```
div {  
  width: 300px;  
  padding: 25px;  
  box-sizing: border-box;  
}
```



## 4.6.- Outline

Un contorno es una línea que se dibuja alrededor de los elementos, FUERA de los bordes, para que el elemento "se destaque".



**NOTA:** ¡ El contorno difiere de los bordes ! A diferencia del borde, el contorno se dibuja fuera del borde del elemento y puede superponerse a otro contenido. Además, el contorno NO forma parte de las dimensiones del elemento; El ancho y alto total del elemento no se ven afectados por el ancho del contorno.

Tiene las siguientes propiedades de esquema:

- outline-style
- outline-color
- outline-width
- outline-offset
- outline

Puede tomar los siguientes valores:

- **dotted**- Define un contorno de puntos
- **dashed**- Define un contorno discontinuo
- **solid**- Define un contorno sólido
- **double**- Define un doble contorno.
- **groove**- Define un contorno ranurado 3D
- **ridge**- Define un contorno estriado 3D
- **inset**- Define un contorno insertado en 3D
- **outset**- Define un esquema de inicio en 3D.
- **none**- No define ningún contorno
- **hidden**- Define un contorno oculto



## 5.- Aplicación de CSS a:

### 5.1.- Text

Aplicado propiedad	Ejemplo
<b>color</b>	<pre>h1 {   color: green; }</pre>
<b>text-align</b>	Puede estar alineado, centrado o justificado: <b>right, left, center, justify</b>
<b>vertical-align</b>	Establece la alineación vertical de un elemento. Valores: <b>baseline, text-top, text-bottom, sub y super</b>
<b>direction y unicode-bidi</b>	Pueden utilizar para cambiar la dirección del texto de un elemento: <pre>p.ex1 {   direction: rtl;   unicode-bidi: bidi-override; }</pre> <p>&lt;p class="ex1"&gt;This is right-to-left text direction.&lt;/p&gt;</p> <p>Salida</p> <p>.noitcerid txet tfel-ot-thgir si sihT</p>
<b>text-decoration</b>	Aplica subrayado, tachado y color
<b>text-transformation</b>	Se utiliza para especificar letras mayúsculas y minúsculas en un texto. Valores: <b>uppercase, lowercase y capitalize</b>
<b>text-indent</b>	Aplica sangría
<b>letter-spacing</b>	Espacio entre letras
<b>line-height</b>	Espacio entre líneas



<b><i>word-spacing</i></b>	Especifica cómo se manejan los espacios en blanco dentro de un elemento.  Valores: nowrap => elimina espacios
<b><i>text-shadow</i></b>	Aplica sombra al texto.

## 5.2.- List

Tenemos:

- Ordenadas => <ol>
- Desordenadas => <ul>

Marcadores <b><i>list-style-type</i></b>	Valores: circle, square, upper-roman y lower-alpha  <b>NOTA:</b> Eliminar configuración predeterminada  <code>list-style-type: none;</code>
Imagen como marcador <b><i>list-style-image</i></b>	<pre>ul {   list-style-image:   url('sqpurple.gif'); }</pre>
Eliminar configuración preseterminada	

## 5.3.- Botones

Un botón puede hacer uso de hover y active para darle vida y modificar su gama de color para cada acción y hacerlo atractivo.

Su pongamos el botón, con bordes redondeados con border-radius:

Haz clic aquí

Que cuando pase por encima el ratón cambie de color y cuando esté activo (lo presionemos también cambie de color).



El código HTML sería:

```
<button class="my-button">Haz clic aquí</button>
```

El código CSS sería:

```
.my-button {  
  padding: 10px 20px;  
  background-color: #4CAF50;  
  color: white;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
}  
  
.my-button:hover {  
  background-color: #45a049;  
}  
  
.my-button:active {  
  background-color: #3e8e41;  
}
```

## 5.4.- Tables

Veamos algunas propiedades interesantes:

### 5.3.1.- border-collapse

Por lo general una tabla tiene el siguiente aspecto:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Para eliminar el doble borde usaremos la propiedad **border-collapse** con el valor **collapse**.





```
<style>
    table, td, th {
        border: 1px solid;
    }

    table {
        width: 100%;
        border-collapse: collapse;
    }
</style>
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

### 5.3.2.- Width y Height

Aplicable a la tabla o filas.

Si lo aplicamos a table el 100% ocupará todo el ancho del elemento en el que esté.

### 5.3.3.- text-align

Establece la alineación horizontal (como izquierda, derecha o centro) del contenido en <th> o <td>.

### 5.3.4.- vertical-align

Establece la alineación vertical (como superior, inferior o media) del contenido en <th> o <td>.

### 5.3.5.- padding

Controlar el espacio entre el borde y el contenido de una tabla, use la propiedad padding en los elementos <td> y <th>.

### 5.3.6.- Divisores horizontales

Agregue propiedad a <th> y <td> para divisores horizontales:



Ejemplo:

```
th, td {  
    border-bottom: 1px solid #ddd;  
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

### 5.3.7.- Marcado de fila con selector *hover*

Para resaltar las filas de la tabla al pasar el mouse sobre la fila:

Ejemplo:

```
tr:hover {background-color: coral;}
```

### 5.3.8.- Tablas con estilo cebra

Para tablas con rayas de cebra, use el selector `nth-child()` y agregue a `background-color` a todas las filas pares (o impares) de la tabla:

```
<style>  
    table {  
        border-collapse: collapse;  
        width: 100%;  
    }  
  
    th, td {  
        text-align: left;  
        padding: 8px;  
    }  
  
    tr:nth-child(even) {  
        background-color: #f2f2f2;  
    }  
</style>
```



### 5.3.9.- Tablas responsiva

Una tabla responsiva mostrará una barra de desplazamiento horizontal si la pantalla es demasiado pequeña para mostrar el contenido completo:

Agregue un elemento contenedor (como <div>) alrededor de overflow-x:auto del elemento <table> para que responda:

Ejemplo:

```
<style>
  table {
    border-collapse: collapse;
    width: 100%;
  }

  th, td {
    text-align: left;
    padding: 8px;
  }

  tr:nth-child(even) {background-color: #f2f2f2;}
</style>
```

First Name	Last Name	Points	Points	Points	Points	Points	Points	Points	Points	Po
Jill	Smith	50	50	50	50	50	50	50	50	50
Eve	Jackson	94	94	94	94	94	94	94	94	94
Adam	Johnson	67	67	67	67	67	67	67	67	67

## 6.- Propiedad *display*

Se utiliza para especificar **cómo se muestra un elemento en una página web**.

Cada elemento HTML tiene un valor de visualización predeterminado, según el tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es block o inline.

Los valores más usados son:



block	Muestra un elemento como un bloque
inline	Muestra un elemento como elementos en línea.
flex	Muestra un elemento como un contenedor flex
grid	Muestra un elemento como un contenedor grid
inline-block	Muestra un elemento como un contenedor de bloques de nivel en línea. El elemento en sí tiene el formato de elemento en línea, pero puede aplicar valores de alto y ancho.
inline-flex	Muestra un elemento como un contenedor de flex de nivel en línea.
inline-grid	Muestra un elemento como un contenedor de grid de nivel en línea.
inline-table	Muestra un elemento como una tabla de nivel en línea.
inherit	Hereda las propiedades del elemento padre
none	El elemento es completamente movido

## 6.1.- Crear un menú horizontal con lista

### A.- Estructura HTML

Comenzamos creando la estructura básica del menú en HTML. Usaremos una lista desordenada `<ul>` con elementos de lista `<li>` para crear los ítems del menú. Cada ítem del menú será un enlace `<a>` que llevará a diferentes páginas o secciones del sitio web.

```
<header>
  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Acerca de</a></li>
      <li><a href="#">Servicios</a></li>
      <li><a href="#">Portafolio</a></li>
      <li><a href="#">Contacto</a></li>
    </ul>
  </nav>
</header>
```



### B.- Aplicamos estilo CSS

Queremos que el menú sea horizontal, quitar las viñetas y que los ítems estén alineados uno al lado del otro.


Para quitar las viñetas, usaremos la propiedad **list-style-type:none**; en la etiqueta **<ul>**. Para alinear los ítems usaremos la propiedad **display:inline**; en las etiquetas **li**.

```
nav ul {  
    list-style-type: none; /* Quitamos las viñetas de la lista */  
}  
nav li {  
    display: inline; /* Ítems del menú en una línea horizontal */  
}
```

### C.- Añadimos color y estilo personalizado

Puedes personalizar los colores y estilos del menú para que se ajusten al diseño de tu sitio web. Por ejemplo, puedes cambiar el color de fondo del encabezado, los colores de los ítems del menú y los efectos de interacción al pasar el cursor sobre ellos.

#### Ejemplo1:



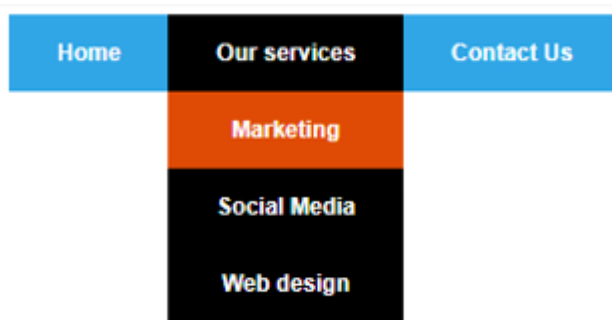
Código CSS:

```
body {  
    margin: 0;  
    font-family: Arial, sans-serif;  
}  
  
header {  
    background-color: #333;  
    color: #fff;  
    padding: 10px;  
}  
  
nav ul {  
    list-style: none;  
    padding: 0;  
    margin: 0;  
}
```



```
nav li {  
    display: inline; /* Hace que los ítems del menú estén en una línea horizontal */  
    margin-right: 20px; /* Espacio entre los ítems del menú */  
}  
  
nav a {  
    text-decoration: none;  
    color: #fff;  
    font-size: 16px;  
}  
  
nav a:hover {  
    text-decoration: underline;  
}
```

### Ejemplo 2 con dropdown:



El ítem «Our services» tiene un submenú desplegable que se oculta inicialmente y se muestra al pasar el cursor sobre él. El submenú está representado por otra lista desordenada (<ul>) con un id «submenu». Dentro del submenú, hay tres elementos de lista (<li>) que contienen enlaces (<a>) con los nombres de los servicios ofrecidos.

### Código HTML:

```
<nav class="dropdownmenu">  
  <ul>  
    <li><a href="#">Home</a></li>  
    <li><a href="#">Our services</a>  
      <ul id="submenu">  
        <li><a href="#">Marketing</a></li>  
        <li><a href="#">Social Media</a></li>  
        <li><a href="#">Web design</a></li>  
      </ul>  
    </li>  
    <li><a href="#">Contact Us</a></li>  
  </ul>  
</nav>
```



### Código CSS:

```
.dropdownmenu ul, .dropdownmenu li {
    margin: 0;
    padding: 0;
}
.dropdownmenu ul {
    background: gray;
    list-style: none;
    width: 100%;
}
.dropdownmenu li {
    float: left;
    position: relative;
    width: auto;
}
.dropdownmenu a {
    background: #30A6E6;
    color: #FFFFFF;
    display: block;
    font: bold 12px/20px sans-serif;
    padding: 10px 25px;
    text-align: center;
    text-decoration: none;
    -webkit-transition: all .25s ease;
    -moz-transition: all .25s ease;
    -ms-transition: all .25s ease;
    -o-transition: all .25s ease;
    transition: all .25s ease;
}
.dropdownmenu li:hover a {
    background: #000000;
}
#submenu {
    left: 0;
    opacity: 0;
    position: absolute;
    top: 35px;
    visibility: hidden;
    z-index: 1;
}
li:hover ul#submenu {
    opacity: 1;
```



```
top: 40px; /* adjust this as per top nav padding top & bottom
comes */
visibility: visible;
}
#submenu li {
float: none;
width: 100%;
}
#submenu a:hover {
background: #DF4B05;
}
#submenu a {
background-color: #000000;
}
```

## 7.- Propiedad *position*

Especifica el tipo de método de posicionamiento utilizado para un elemento (estático, relativo, fijo, absoluto o fijo).

Hay cinco valores de posición diferentes:

<b>static</b>	No se ven afectados por las propiedades superior, inferior, izquierda y derecha. Siempre se coloca de acuerdo con el flujo normal de la página
<b>relative</b>	Establecer las propiedades superior, derecha, inferior e izquierda de un elemento en posición relativa hará que se ajuste fuera de su posición normal.
<b>fixed</b>	Se coloca en relación con la ventana gráfica, lo que significa que siempre permanece en el mismo lugar incluso si se desplaza la página. Las propiedades superior, derecha, inferior e izquierda se utilizan para posicionar el elemento.
<b>absolute</b>	Se posiciona en relación con el ancestro posicionado más cercano (en lugar de posicionarse en relación con la ventana gráfica, como fijo).
<b>sticky</b>	Posiciona según la posición de desplazamiento del usuario.  Un elemento adhesivo alterna entre relative y fixed, según la posición de desplazamiento. Se coloca en relación hasta que se alcanza una posición de desplazamiento determinada en la ventana gráfica; luego se "pega" en su lugar (como posición: fija).





Luego, los elementos se colocan utilizando las propiedades superior, inferior, izquierda y derecha. Sin embargo, estas propiedades no funcionarán a menos que *position* se establezcan primero. También funcionan de manera diferente según el valor de la posición.

## 8.- Overflow

La propiedad **overflow** especifica si se recortará el contenido o se agregarán barras de desplazamiento cuando el contenido de un elemento sea demasiado grande para caber en el área especificada.

La overflow tiene los siguientes valores:

- **visible**- Por defecto. El desbordamiento no se recorta. El contenido se muestra fuera del cuadro del elemento.
- **hidden**- El desbordamiento se recorta y el resto del contenido será invisible.
- **scroll**- Se recorta el desbordamiento y se agrega una barra de desplazamiento para ver el resto del contenido.
- **auto**- Similar a scroll, pero agrega barras de desplazamiento solo cuando es necesario

## 9.- Float

La propiedad se utiliza para posicionar y formatear contenido, por ejemplo, dejar que una imagen flote a la izquierda del texto en un contenedor.

La propiedad puede tener uno de los siguientes valores:

- **left**- El elemento flota a la izquierda de su contenedor.
- **right**- El elemento flota a la derecha de su contenedor.
- **none**- El elemento no flota (se mostrará justo donde aparece en el texto). Esto es predeterminado
- **inherit**- El elemento hereda el valor flotante de su padre.

En su uso más simple, la floatpropiedad se puede utilizar para ajustar texto alrededor de imágenes.



## 10.- Gradiente

Los gradientes nos permiten añadir efectos de colores degradados en nuestros diseños. Se configuran como fondos, por lo que tendremos que usar la propiedad “background”. Disponemos de dos tipos de gradientes: lineal y radial.

### 10.1.- Gradiente lineal

El gradiente lineal se define mediante la función linear-gradient() y permite crear fondos degradados en una dirección específica. El formato es el siguiente:

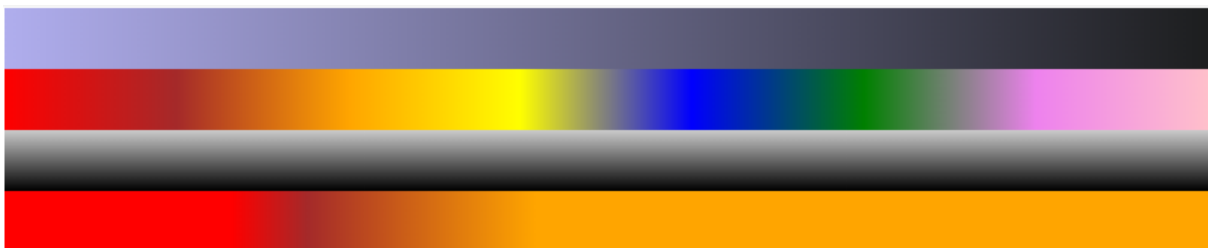
*background: linear-gradient(dirección|ángulo, color, color, color...);*  
*background: linear-gradient(dirección|ángulo, color posición, color posición, color posición...);*  
*background: linear-gradient(color, color, color...);*

Significado de las propiedades:

- **Dirección | ángulo:** indica la dirección o ángulo del gradiente. Puede ser especificado usando las palabras clave to top, to bottom, to left y to right. También puede ser indicado por un ángulo para declarar una dirección específica del gradiente.
- **Color:** permite crear una lista con varias paradas para ir cambiando el color del gradiente. Se pueden definir tantos colores como se necesiten. Se puede utilizar el nombre de color (en inglés), valor hexadecimal, color en RGB o HSL.
- **Posición:** altura a la que comienza a cambiar el color del gradiente.

Ejemplo:

Cuatro franjas (cada una es un <div>) con aplicación lineal diferente.:



El código CSS es:

```
div:nth-child(1){ background: linear-gradient(90deg, rgba(176,174,238,1) 0%, rgba(29,30,31,1));}  
div:nth-child(2){ background: linear-gradient(to right,red,brown,orange,yellow,blue,green,violet,pink);}  
div:nth-child(3){ background: linear-gradient(to top, #000, #ccc);}
```



```
div{padding:40px;}  
div:nth-child(4){ background: linear-gradient(to right,red 300px,brown 400px,orange 700px);}
```

## 10.2.- Gradiente radial

El gradiente radial nos permite crear degradados con formas circulares.  
Debemos usar la función `radial-gradient()` y un nuevo atributo para la forma.

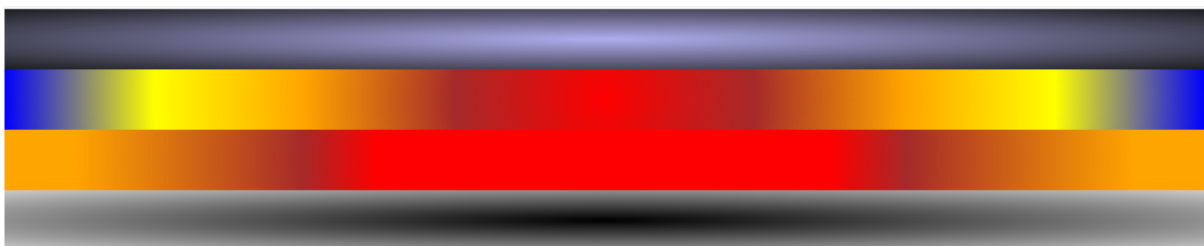
```
background: radial-gradient(forma, color, color, color...);  
background: radial-gradient(forma posición, color, color, color...);  
background: radial-gradient(forma, color posición, color posición, color posición...);  
background: radial-gradient(color, color, color...);
```

Significado de las propiedades:

- **Forma:** existen dos tipos de forma: `circle` y `ellipse` (el valor por defecto será `ellipse`). De forma optativa se puede indicar el tamaño de la forma.
- **Color:** permite crear una lista con varias paradas para ir cambiando el color del gradiente. Se pueden definir tantos colores como se necesiten.
- **Posición:** para cada color se puede indicar la posición en la que comienzan las transiciones.

Ejemplo:

Cuatro franjas (cada una es un `<div>`) con aplicación radial diferente.:



El código CSS es:

```
div:nth-child(1){ background: radial-gradient(ellipse, rgba(176,174,238,1) 0%, rgba(29,30,31,1));}  
div:nth-child(2){ background: radial-gradient(circle,red,brown,orange,yellow,blue);}  
div:nth-child(3){ background:radial-gradient(circle,red 300px,brown 400px,orange 700px);}  
div:nth-child(4){ background: radial-gradient(#000, #ccc);}  
div{padding:40px;}
```



## 11.- Cursores

Se utilizan para cambiar la apariencia del cursor del ratón cuando se encuentra sobre un elemento interactivo. Esto permite personalizar la forma del cursor y proporcionar retroalimentación visual al usuario.

CSS proporciona una variedad de valores para la propiedad **cursor**, que se utiliza para especificar el estilo del cursor. Puedes ver la lista completa en el [Anexo V](#).

Veamos alguno de ellos: crea el código html siguiente y verás que al pasar el ratón por encima cambiará el cursor.

### Código HTML:

```
<div class="container">
  <div class="caja caja1">crosshair</div>
  <div class="caja caja2">help</div>
  <div class="caja caja3">move</div>
  <div class="caja caja4">pointer</div>
  <div class="caja caja5">progress</div>
  <div class="caja caja6">text</div>
  <div class="caja caja7">wait</div>
  <div class="caja caja8">vertical-text</div>
  <div class="caja caja9">alias</div>
  <div class="caja caja10">no-drop</div>
  <div class="caja caja11">grab</div>
  <div class="caja caja12">grabbing</div>
  <div class="caja caja13">zoom-in</div>
  <div class="caja caja14">zoom-out</div>
  <div class="caja caja15">not-allowed</div>
  <div class="caja caja16">no-drop</div>
  <div class="caja caja17">copy</div>
  <div class="caja caja18">ew-resize</div>
  <div class="caja caja19">ns-resize</div>
  <div class="caja caja20">nwse-resize</div>
  <div class="caja caja21">nesw-resize</div>
  <div class="caja caja22">col-resize</div>
  <div class="caja caja23">row-resize</div>
  <div class="caja caja24">context-menu</div>
  <div class="caja caja25">none</div>
  <div class="caja caja26">Personalized</div>
</div>
```

### Código CSS:



```
.container{
  position: relative;
}
.caja{
  width: 180px;
  height: 180px;
  color: #FFF;
  text-align: center;
  line-height: 180px;
  margin: 15px;
  float: left;
  font-size: 18px;
  font-family: Arial;
}
.caja1{
  background-color: #C0392B;
  cursor: crosshair;
}
.caja2{
  background-color: #9B59B6;
  cursor: help;
}
.caja3{
  background-color: #2980B9;
  cursor: move;
}
.caja4{
  background-color: #3498DB;
  cursor: pointer;
}
.caja5{
  background-color: #17A589;
  cursor: progress;
}
.caja6{
  background-color: #F1C40F;
  cursor: text;
}
.caja7{
  background-color: #E67E22;
  cursor: wait;
}
.caja8{
  background-color: #95A5A6;
```



```
    cursor: vertical-text;
}
.caja9{
    background-color: #2C3E50;
    cursor: alias;
}
.caja10{
    background-color: grey;
    cursor: no-drop;
}
.caja11{
    background-color: black;
    cursor: grab;
}
.caja12{
    background-color: #C0392B;
    cursor: grabbing;
}
.caja13{
    background-color: #9B59B6;
    cursor: zoom-in;
}
.caja14{
    background-color: #2980B9;
    cursor: zoom-out
}
.caja15{
    background-color: #3498DB;
    cursor: not-allowed;
}
.caja16{
    background-color: #17A589;
    cursor: progress;
}
.caja17{
    background-color: #F1C40F;
    cursor: copy
}
.caja18{
    background-color: #E67E22;
    cursor: ew-resize;
}
.caja19{
    background-color: #95A5A6;
```



```
        cursor: ns-resize;
    }
    .caja20{
        background-color: #2C3E50;
        cursor: nwse-resize;
    }
    .caja21{
        background-color: grey;
        cursor: nesw-resize;
    }
    .caja21{
        background-color: black;
        cursor: nesw-resize;
    }
    .caja26{
        background-color: #C0392B;
        cursor: url(https://cdn-icons-png.flaticon.com/128/2662/2662839.png),
pointer;
    }
    .caja21{
        background-color: #9B59B6;
        cursor: nesw-resize;
    }
    .caja22{
        background-color: #2980B9;
        cursor: col-resize;
    }
    .caja23{
        background-color: #3498DB;
        cursor: row-resize;
    }
    .caja24{
        background-color: #17A589;
        cursor: context-menu;
    }
    .caja25{
        background-color: #E67E22;
        cursor: none;
    }
    .caja26{
        background-color: #95A5A6;
        cursor: url(https://cdn-icons-png.flaticon.com/128/2662/2662839.png),
pointer;
    }
```



## 12.- Tooltips

Es una pequeña ventana emergente que se muestra cuando el usuario coloca el cursor sobre un elemento. Proporciona información adicional o aclaratoria sobre el elemento al que está asociado, sin ocupar mucho espacio en la página. El tooltip suele aparecer cerca del elemento objetivo y desaparece cuando el cursor se mueve fuera de él.

Los tooltips son una forma efectiva de **mejorar la usabilidad y la experiencia del usuario al proporcionar información adicional** de manera contextual y discreta.

Pasos para crear un tooltip:

**A.-** Crea la estructura HTML para el elemento que tendrá el tooltip. Por ejemplo, puedes usar un elemento `<span>` dentro de un elemento con una clase específica.

Por ejemplo:

```
<div class="tooltip-container">  
  <span class="tooltip-text">Este es un tooltip</span>  
  Mi elemento  
</div>
```

**B.-** Estiliza el contenedor del tooltip y el texto del tooltip en tu archivo CSS. Puedes ajustar los estilos según tus preferencias.

```
.tooltip-container {  
  position: relative;  
  display: inline-block;  
}  
  
.tooltip-text {  
  visibility: hidden;  
  width: 120px;  
  background-color: #000;  
  color: #fff;  
  text-align: center;  
  border-radius: 6px;  
  padding: 5px;  
  position: absolute;  
  z-index: 1;  
  bottom: 125%; /* Posiciona el tooltip encima del elemento */  
  left: 50%;  
  transform: translateX(-50%);  
  opacity: 0;
```





```
transition: opacity 0.3s;  
}
```

C.- Agrega una regla CSS para mostrar el tooltip cuando se pasa el cursor sobre el elemento. Puedes utilizar el selector :hover para lograrlo.

```
.tooltip-container:hover .tooltip-text {  
  visibility: visible;  
  opacity: 1;  
}
```

## 13.- Transiciones

Nos permiten aplicar un **cambio de estilo gradual a los elementos** del documento HTML. Además, nos ofrecen la ventaja de poder **especificar el tiempo** para que se produzca la transformación entre estilos, de esta forma podríamos utilizarlas para **dar un efecto de animación**.

Los parámetros se pueden aplicar en una sola línea o por separado.

Formato:

**transition:**[propiedad a modificar] [Duración] [Tipo de animación] [Retardo];

Significado de las propiedades:

**[Propiedades a modificar]:** algunas de las propiedades que podemos modificar utilizando transiciones son las que se muestran en la siguiente lista:

background-color	border	border-radius	color	top
bottom	left	right	box-shadow	width
height	line-height	margin	opacity	word-spacing
letter-spacing	fill	padding	stroke	text-shadow
vertical-align	visibility	z-index		

**[Duración en segundos]:** se debe especificar el número de segundos que va a durar la animación. Por ejemplo: 3s (3 segundos).



**[Tipo de animación]:** esta propiedad es opcional y sirve para indicar la velocidad de la animación. Algunas de las posibilidades son las siguientes:

- **linear:** la velocidad de la animación es uniforme en todo el recorrido.
- **ease:** la velocidad se acelera al inicio, luego se retarda un poco y se acelera al final de nuevo.
- **ease-in:** la animación empieza lentamente y va aumentando progresivamente.
- **ease-out:** la animación empieza muy rápida y va descendiendo progresivamente.
- **ease-in-out:** la animación empieza y acaba lentamente, y es en la parte central del recorrido donde la velocidad es más rápida.

**[Retardo]:** tiempo (en segundos) que el navegador esperará antes de poner en marcha la animación. Se especifica el número de segundos a esperar seguido de la «s» (ejemplo: 1s).

Ejemplo:

Crea un <div> con la clase “caja1” y aplica el código CSS:

```
.caja1{
    background-color: #C0392B;
    transition: background-color 1s linear;
}
.caja1:hover{
    background-color: #3F51B5;
}
```

Funcionamiento:

En la clase “caja” yo defino un fondo y una transición. Esta transición se activará cuando se cambie el color de fondo, ¿y cuándo sucede eso? Pues, en este ejemplo, cuando se activa “hover” (al pasar el ratón por encima del elemento) se cambia el color de fondo, pero no inmediatamente sino usando lo especificado en **transition**, es decir, un retardo de 1s y de forma lineal.

## 14.- Transformaciones

**Permiten rotar, torcer, escalar o desplazar los elementos** de nuestra página web. Este tipo de propiedades de CSS3 son muy interesantes para convertir el lenguaje de hojas de estilo en un sistema capaz de realizar todo tipo de efectos visuales.

Las dos propiedades que nos sirven para definir las transformaciones son:

- **transform:** La posición de origen para realizar la transformación es el eje central del elemento.



- **transform-origin**: La posición de origen que se utilizará para la transformación es por defecto el lado superior izquierdo del elemento.

## 15.1.- Escalar

Modifica el tamaño de los elementos. La función **scale ()** se establece con uno o dos valores, que representan la cantidad de escala que se aplica en cada dirección: **scale (x)** o **scale(x,y)**. Se define mediante un valor numérico de manera que cuando un valor de coordenadas está fuera del rango [-1, 1], el elemento crece a lo largo de esa dimensión. Cuando está dentro del rango el elemento se encoge.

Ejemplo:

```
/* Escala el elemento a la mitad */  
transform: scale(0.5);
```

## 15.2.- Trasladar

Cambia la posición del elemento hacia la izquierda, derecha, arriba o abajo. La función **translate()** se establece con uno o dos valores: **translate(x)** o **translate(x,y)**. Los valores x e y son los vectores de translación en las coordenadas x e y. Sus valores pueden estar definidos en píxeles, porcentajes,...

Ejemplo:

```
/* Traslada el elemento 10px hacia la derecha */  
transform: translate(10px);
```

## 15.3.- Rotar

Gira o rota los elementos en grados: **rotate(v)**.

Ejemplo:

```
/* Rota el elemento 45 grados */  
transform: rotate(45deg);
```

## 15.4.- Distorsionar

Distorsiona los elementos según el ángulo en grados. La función **skew()** se establece con uno o dos valores: **skew(x)** o **skew(x,y)**.



Ejemplo:

```
/* Distorsiona el elemento 45 grados en el eje x */  
transform: skew(45deg);
```

## 15.5.- Matrix

Mueve o transforma los elementos con precisión de píxel. La función `matrix()` se establece con seis valores numéricos: **`matrix(a,b,c,d,x,y)`**. Los dos últimos valores representan la translación y los primeros la transformación lineal.

Ejemplo:

```
transform: matrix(0.5, 0.1, 0.5, 1, 10, -2);
```

Para que veas todos los efectos creamos varios `<div>` y aplicaremos las transformaciones para ver su efecto.

**Código HTML:**

```
<div class="container">  
  <div class="caja caja1">Scale</div>  
  <div class="caja caja2">Rotate</div>  
  <div class="caja caja3">Translate</div>  
  <div class="caja caja4">Skew</div>  
  <div class="caja caja5">SkewY</div>  
  <div class="caja caja6">SkewX</div>  
  <div class="caja caja7">TranslateX</div>  
  <div class="caja caja8">TranslateY</div>  
  <div class="caja caja9">Perspective rotate</div>  
  <div class="caja caja10">Perspective rotate</div>  
  <div class="caja caja11">Matrix</div>  
</div>
```

**Código CSS:**

```
.container{  
  position: relative;  
}  
.caja{  
  width: 180px;  
  height: 180px;  
  color: #FFF;  
  text-align: center;
```



```
line-height: 180px;
margin: 15px;
float: left;
font-size: 18px;
font-family: Arial;
}
.caja1{
background-color: #C0392B;
-webkit-transition: 1s linear;
transition: 1s linear;
}
.caja1:hover{
-webkit-transform: scale(.5);
transform: scale(.5);
}
.caja2{
background-color: #9B59B6;
-webkit-transition: 1s linear;
transition: 1s linear;
}
.caja2:hover{
-webkit-transform: rotate(360deg);
transform: rotate(360deg);
}
.caja3{
-webkit-transition: 1s linear;
transition: 1s linear;
background-color: #2980B9;
}
.caja3:hover{
-webkit-transform: translate(10%);
transform: translate(10%);
}
.caja4{
background-color: #3498DB;
-webkit-transition: 1s linear;
transition: 1s linear;
}
.caja4:hover{
-webkit-transform: skew(10deg);
transform: skew(10deg);
}
.caja5{
background-color: #17A589;
```



```
-webkit-transition: 1s linear;
transition: 1s linear;
}
.caja5:hover{
  -webkit-transform: skewY(10deg);
  transform: skewY(10deg);
}
.caja6{
  background-color: #F1C40F;
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja6:hover{
  -webkit-transform: skewX(-5deg);
  transform: skewX(-5deg);
}
.caja7{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #E67E22;
}
.caja7:hover{
  -webkit-transform: translateX(20px);
  transform: translateX(20px);
}
.caja8{

  -webkit-transition: 1s linear;

  transition: 1s linear;
  background-color: #95A5A6;
}
.caja8:hover{
  -webkit-transform: translateY(20px);
  transform: translateY(20px);
}
.caja9{
  -webkit-transition: 1s linear;
  transition: 1s linear;
  background-color: #2C3E50;
}
.caja9:hover{
  -webkit-transform: perspective(150px) rotateX(45deg);
  transform: perspective(150px) rotateX(45deg);
```

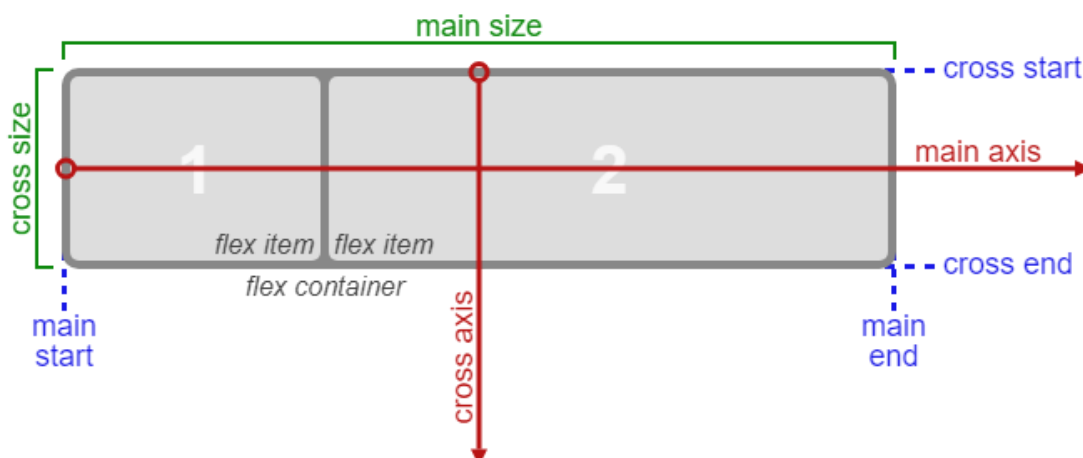


```
}  
.caja10{  
  -webkit-transition: 1s linear;  
  transition: 1s linear;  
  background-color: grey;  
}  
.caja10:hover{  
  -webkit-transform: perspective(150px) rotateY(45deg);  
  transform: perspective(150px) rotateY(45deg);  
}  
.caja11{  
  -webkit-transition: 1s linear;  
  transition: 1s linear;  
  background-color: black;  
}  
.caja11:hover{  
  -webkit-transform: matrix(0.5, 0.1, 0.5, 1, 10, -2);  
  transform: matrix(0.5, 0.1, 0.5, 1, 10, -2);  
}
```

## 15.- Flex

Flexbox, también conocido como **modelo de caja flexible**, es un diseño unidimensional que permite crear un diseño más eficiente y predecible en situaciones donde **el tamaño de los elementos es desconocido o dinámico**.

Flexbox trabaja en una dimensión a la vez, ya sea en fila o en columna. Esto permite una fácil alineación y distribución de elementos a lo largo de un eje principal (main axis) y un eje transversal (cross axis).

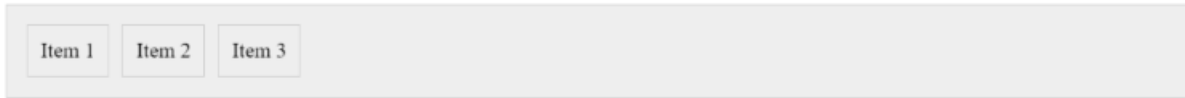


Para usarlo debemos declarar un contenedor y aplicar: flex o inline-flex



## 15.1.- display: flex | inline-flex

- **display: flex**



Al aplicarlo este se convierte en un contenedor flexible que distribuye su contenido de manera flexible a lo largo del eje principal.

Por defecto, el eje principal es horizontal (de izquierda a derecha).

Los elementos secundarios dentro de este contenedor flex se comportarán como bloques y se ajustarán automáticamente para llenar el contenedor en la dirección principal.

Ejemplo de aplicación:

```
.mi-contenedor-flex {  
  display: flex;  
}
```

- **display: inline-flex.**



Al aplicarlo el contenedor se comporta como un elemento en línea en lugar de un bloque. Esto significa que ocupará solo el ancho necesario y permitirá que otros elementos en línea coexistan en la misma línea.

## 15.2.- flex-wrap: nowrap | wrap | wrap-reverse

Podemos indicar qué elementos podemos desplazar indicando a flex-wrap uno de los siguientes valores:

- **nowrap:** los elementos no pasan a la siguiente fila y se reduce su anchura para mostrarlos. Este es el valor por defecto.

- 







- **wrap**: los elementos pasan a la siguiente fila y conservan su anchura.



- **wrap-reverse**: los elementos pasan a la siguiente fila, pero en sentido inverso al de su declaración.



### 15.3.- justify-content: flex-start | flex-end | center | space-between | space-around

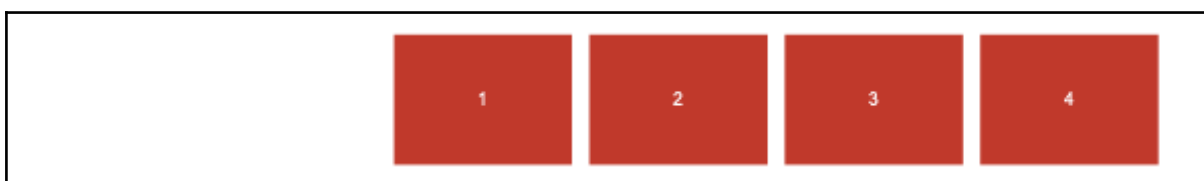
La propiedad **justify-content** es la que define la justificación horizontal de los elementos hijos de un contenedor flexible.

Los valores posibles son:

- **flex-start**: posiciona los elementos a la izquierda del contenedor. Valor por defecto.

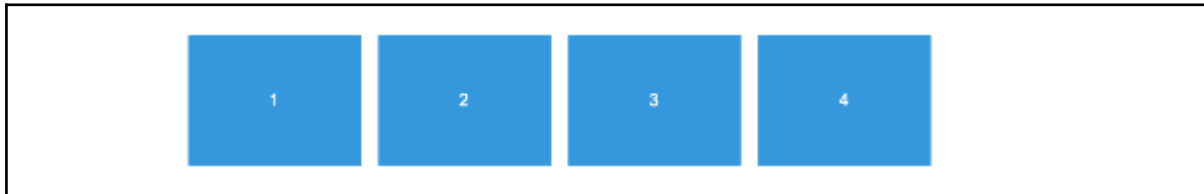


- **flex-end**: posiciona los elementos a la derecha del contenedor.





- **center**: centra los elementos en el contenedor



- **space-between**: añade un espacio idéntico entre los elementos. El primer elemento está alineado a la izquierda del contenedor, y el último, a la derecha.



- **space-around**: espacia de forma regular los elementos que no están alineados a la izquierda y a la derecha del contenedor.



## 15.4.- align-items: stretch | flex-start | flex-end | center | baseline

Actúa sobre el eje vertical. Para utilizar esta propiedad primero se debe definir la propiedad **flex-direction: row**.

La propiedad align-items acepta cinco valores:

- **stretch**: los elementos se amplían verticalmente para ocupar toda la altura disponible en el contenedor. Este es el valor por defecto.
- **flex-start**: los elementos se colocan en la parte superior del contenedor.
- **flex-end**: los elementos se colocan en la parte inferior del contenedor.
- **center**: los elementos se colocan en el centro del contenedor.



- **baseline:** los elementos se alinean sobre la línea de base del texto.

La sintaxis:

```
#container{  
  display: flex;  
  flex-direction: row;  
  align-items: stretch;  
}
```

Con el siguiente código veremos en acción su aplicación:

### Código HTML:

```
<h2>align-items: stretch;</h2>  
<div id="flex-container1">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
</div>  
  
<h2>align-items: flex-start;</h2>  
<div id="flex-container2">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
</div>  
  
<h2>align-items: flex-end;</h2>  
<div id="flex-container3">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
</div>  
  
<h2>align-items: center;</h2>  
<div id="flex-container4">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>
```



```
</div>

<h2>align-items: baseline;</h2>
<div id="flex-container5">
  <div><h3>1</h3></div>
  <div>2</div>
  <div><h4>3</h4></div>
  <div>4</div>
</div>
```

#### Código CSS:

```
#flex-container1{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: stretch;
  height: 250px;
}
#flex-container1 div{
  width: 200px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
  font-family: 'Roboto', sans-serif;
  background-color: #9B59B6;
}
#flex-container2{
  display: flex;
  justify-content: center;
  flex-direction: row;
  align-items: flex-start;
  height: 250px;
}
#flex-container2 div{
  width: 200px;
  height: 150px;
  color: #FFF;
  text-align: center;
  line-height: 150px;
  margin: 10px;
  font-size: 20px;
```



```
font-family: Arial;
background-color: #C0392B;
}
#flex-container3{
display: flex;
justify-content: center;
flex-direction: row;
align-items: flex-end;
height: 250px;
}
#flex-container3 div{
width: 200px;
height: 150px;
color: #FFF;
text-align: center;
line-height: 150px;
margin: 10px;
font-size: 20px;
font-family: Arial;
background-color: #3498DB;
}
#flex-container4{
display: flex;
justify-content: center;
flex-direction: row;
align-items: center;
height: 250px;
}
#flex-container4 div{
width: 200px;
height: 150px;
color: #FFF;
text-align: center;
line-height: 150px;
margin: 10px;
font-size: 20px;
font-family: Arial;
background-color: #17A589;
}
#flex-container5{
display: flex;
justify-content: center;
flex-direction: row;
align-items: baseline;
```



```
height: 250px;
}
#flex-container5 div{
width: 200px;
height: 150px;
color: #FFF;
text-align: center;
line-height: 120px;
margin: 10px;
font-size: 20px;
font-family: Arial;
background-color: #E67E22;
}
```

### 15.5.- flex-direction: row | row-reverse | column | column-reverse

Determina la dirección en la que los elementos flexibles se colocan dentro de un contenedor flex. Puede establecerse para organizar los elementos en una fila (horizontal) o en una columna (vertical).

La propiedad flex-direction acepta cuatro valores:

- **row** (valor por defecto): Coloca los elementos en una fila horizontal. El inicio está a la izquierda y el final a la derecha.



- **row-reverse**: Coloca los elementos en una fila horizontal, pero en orden inverso al valor por defecto. El inicio estará a la derecha y el final a la izquierda.

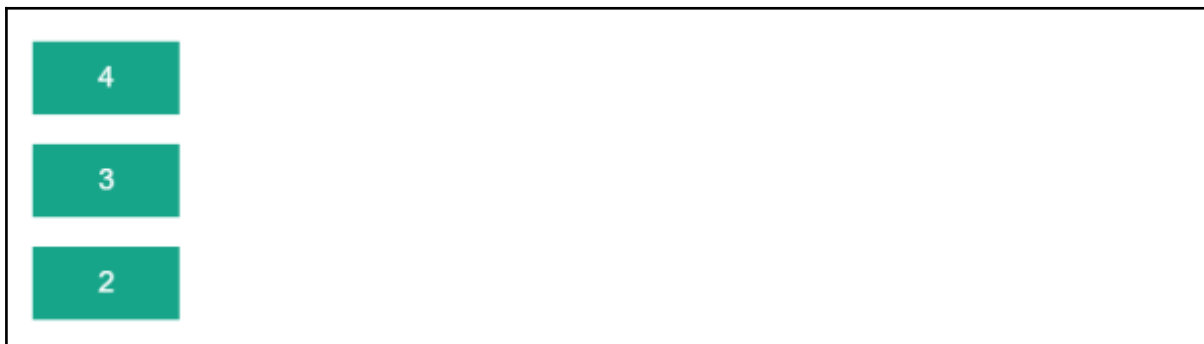




- **column:** Coloca los elementos en una columna vertical. El inicio está arriba y el final abajo.

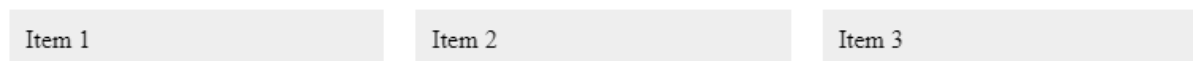


- **column-reverse:** Coloca los elementos en una columna vertical, pero en orden inverso al valor por defecto. El inicio estará abajo y el final arriba.



## 15.6.- flex-grow, flex-shrink y flex-basis

**flex-grow** define cómo un elemento debe crecer en relación con el resto de los elementos flexibles en el contenedor Flexbox.



Ejemplo de sintaxis:

```
.flex-item {  
  flex-grow: 1; /* Todos los elementos crecen de igual manera */  
}
```



**flex-shrink** define cómo los elementos se encogen en relación con los demás cuando no hay suficiente espacio en el contenedor.

Ejemplo de código CSS

```
.flex-container {  
  display: flex;  
  width: 500px; /* Ancho total menor que la suma de los flex-basis */  
}  
.flex-item {  
  flex-basis: 200px;  
}  
/* Primer elemento */  
.flex-container .flex-item:first-child {  
  flex-shrink: 0; /* Este elemento no se encogerá */  
}  
/* Segundo elemento */  
.flex-container .flex-item:nth-child(2) {  
  flex-shrink: 2; /* Este elemento se encogerá el doble que el normal */  
}  
/* Tercer elemento */  
.flex-container .flex-item:nth-child(3) {  
  flex-shrink: 1; /* Este elemento se encogerá normalmente */  
}
```

**flex-basis** establece el tamaño inicial de un elemento antes de que se distribuya el espacio

Sintaxis:

```
.flex-item {  
  flex-basis: 100px; /* Cada elemento comienza con 100px de ancho */  
}
```

## 16.- Grid

La **diferencia básica entre Flex y Grid** es que el primero se pensó para diseño en una dimensión, en una fila o columna. Mientras que Grid fue pensado para trabajar bidimensionalmente, en varias filas o columnas al mismo tiempo.

Para utilizar CSS Grid definiremos un contenedor padre y en su interior los ítems que se necesiten crear. Además, se precisan definir una serie de propiedades, tanto para el





contenedor padre como para las rejillas que se definen en su interior. Veamos en detalle las propiedades más importantes.

## 16.1.- Propiedad display: grid | inline-grid

Indican cómo se comporta el contenedor con respecto al contenido exterior. Con el valor inline-grid el contenedor aparece en línea con respecto al contenido exterior. Con el valor grid, el contenedor aparece en bloque con respecto al contenido exterior.

- inline-grid Cuadrícula en línea con respecto al contenido exterior
- grid Cuadrícula en bloque con respecto al contenido exterior

## 16.2.- Propiedades grid-template-columns y grid-template-rows

Para definir el espaciado o tamaño de las columnas y las filas usamos las propiedades:

Propiedad	Valor	Descripción
grid-template-columns	[col1] [col2] ...	Tamaño de cada columna
grid-template-rows	[fila1] [fila2] ...	Tamaño de cada fila

Ejemplos:

Código	Resultado
<pre>.contenedor{   display: grid;   grid-template-rows: 200px; 200px   grid-template-columns: 200px;   200px; 200px; }</pre>	Se crea una cuadrícula de 2 filas por 3 columnas.
<pre>.contenedor{   display: grid;   grid-template-columns: auto auto   auto auto; }</pre>	<div>Item 1      Item 2      Item 3      Item 4</div>
<pre>.contenedor{   display: grid;   grid-template-columns: 50px 200px   300px 500px; }</pre>	<div>Item 1   Item 2                      Item 3                      Item 4</div>



<pre>.contenedor{   display: grid;   grid-template-columns: 1fr 2fr   1fr 3fr;}</pre>	Item 1	Item 2	Item 3	Item 4
---	--------	--------	--------	--------

## 16.3.- Propiedades row-gap y column-gap

Podemos definir el espaciado entre las rejillas mediante las propiedades **row-gap** y **column-gap**. El espaciado solo se crea entre las columnas/filas, no en los bordes exteriores.

Propiedad	Descripción
column-gap	Espaciado entre columnas
row-gap	Espaciado entre filas

Ejemplo:

```
.contenedor{  
  display: grid;  
  grid-template-rows: 200px; 200px  
  grid-template-columns: 200px; 200px; 200px;  
  column-gap: 20px;  
  row-gap: 10px;  
}
```

1	2	3
4	5	6



## Herramientas

[caniuse](#): para comprobar qué navegadores soportan una determinada propiedad CSS.

[Validador CSS](#).

[Browserling](#): ermite probar cualquier web en los diferentes navegadores: Internet Explorer, Firefox, Chrome, Opera y Safari, en distintas versiones y resoluciones de pantalla.

[Autoprefixer](#): extensión de Visual Studio Code para facilitarnos la tarea de incluir los prefijos de las propiedades CSS

[Fuentes de Google](#): repositorio de fuentes de google.

[FontAweson](#): repositorios de iconos para representar cualquier parte de nuestra interfaz.

[cssgradient](#): generador de gradiente de color

[Patrones](#): generador de patrones con CSS

## Documentos de interés

[50 buenas prácticas en CSS. por @davidvalverde](#):

Este artículo ofrece 50 buenas prácticas en CSS, incluyendo consejos sobre cómo evitar la redundancia, la importancia de no modificar directamente los archivos del tema, y más.

<https://webcode.tools/css-generator>

Generador de código CSS desde sombras hasta transiciones pasando por transformaciones.

<https://css-tricks.com/>

Web con multitud de ejemplos y curiosidades

<https://www.cssmatic.com/>

Editor de gradiente, sombra, color de ruido, de borde

<https://cssgrid-generator.netlify.app/>

Generador de Grid



## Anexos

### Anexo I. Todas las pseudoclases

Selector	Example	Example description
<a href="#">:active</a>	a:active	Selects the active link
<a href="#">:checked</a>	input:checked	Selects every checked <input> element
<a href="#">:disabled</a>	input:disabled	Selects every disabled <input> element
<a href="#">:empty</a>	p:empty	Selects every <p> element that has no children
<a href="#">:enabled</a>	input:enabled	Selects every enabled <input> element
<a href="#">:first-child</a>	p:first-child	Selects every <p> elements that is the first child of its parent
<a href="#">:first-of-type</a>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<a href="#">:focus</a>	input:focus	Selects the <input> element that has focus
<a href="#">:hover</a>	a:hover	Selects links on mouse over
<a href="#">:in-range</a>	input:in-range	Selects <input> elements with a value within a specified range
<a href="#">:invalid</a>	input:invalid	Selects all <input> elements with an invalid value
<a href="#">:lang(language)</a>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<a href="#">:last-child</a>	p:last-child	Selects every <p> elements that is the last child of its parent
<a href="#">:last-of-type</a>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<a href="#">:link</a>	a:link	Selects all unvisited links
<a href="#">:not(selector)</a>	:not(p)	Selects every element that is not a <p> element



---

<a href="#">:nth-child(n)</a>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<a href="#">:nth-last-child(n)</a>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<a href="#">:nth-last-of-type(n)</a>	p:nth-last-of-type(2) )	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<a href="#">:nth-of-type(n)</a>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<a href="#">:only-of-type</a>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<a href="#">:only-child</a>	p:only-child	Selects every <p> element that is the only child of its parent
<a href="#">:optional</a>	input:optional	Selects <input> elements with no "required" attribute
<a href="#">:out-of-range</a>	input:out-of-range	Selects <input> elements with a value outside a specified range
<a href="#">:read-only</a>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<a href="#">:read-write</a>	input:read-write	Selects <input> elements with no "readonly" attribute
<a href="#">:required</a>	input:required	Selects <input> elements with a "required" attribute specified
<a href="#">:root</a>	root	Selects the document's root element
<a href="#">:target</a>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
<a href="#">:valid</a>	input:valid	Selects all <input> elements with a valid value
<a href="#">:visited</a>	a:visited	Selects all visited links



## Anexo II. Todas las pseudoelementos

Selector	Example	Example description
<a href="#"><u>::after</u></a>	p::after	Insert something after the content of each <p> element
<a href="#"><u>::before</u></a>	p::before	Insert something before the content of each <p> element
<a href="#"><u>::first-letter</u></a>	p::first-letter	Selects the first letter of each <p> element
<a href="#"><u>::first-line</u></a>	p::first-line	Selects the first line of each <p> element
<a href="#"><u>::marker</u></a>	::marker	Selects the markers of list items
<a href="#"><u>::selection</u></a>	p::selection	Selects the portion of an element that is selected by a user

## Anexo III. Todas las propiedades background

Property	Description
<a href="#"><u>background</u></a>	Sets all the background properties in one declaration
<a href="#"><u>background-attachment</u></a>	Sets whether a background image is fixed or scrolls with the rest of the page
<a href="#"><u>background-clip</u></a>	Specifies the painting area of the background
<a href="#"><u>background-color</u></a>	Sets the background color of an element
<a href="#"><u>background-image</u></a>	Sets the background image for an element
<a href="#"><u>background-origin</u></a>	Specifies where the background image(s) is/are positioned
<a href="#"><u>background-position</u></a>	Sets the starting position of a background image
<a href="#"><u>background-repeat</u></a>	Sets how a background image will be repeated
<a href="#"><u>background-size</u></a>	Specifies the size of the background image(s)



## Anexo IV. Todas las propiedades de border

Property	Description
<a href="#"><u>border</u></a>	Sets all the border properties in one declaration
<a href="#"><u>border-bottom</u></a>	Sets all the bottom border properties in one declaration
<a href="#"><u>border-bottom-color</u></a>	Sets the color of the bottom border
<a href="#"><u>border-bottom-style</u></a>	Sets the style of the bottom border
<a href="#"><u>border-bottom-width</u></a>	Sets the width of the bottom border
<a href="#"><u>border-color</u></a>	Sets the color of the four borders
<a href="#"><u>border-left</u></a>	Sets all the left border properties in one declaration
<a href="#"><u>border-left-color</u></a>	Sets the color of the left border
<a href="#"><u>border-left-style</u></a>	Sets the style of the left border
<a href="#"><u>border-left-width</u></a>	Sets the width of the left border
<a href="#"><u>border-radius</u></a>	Sets all the four border-*-radius properties for rounded corners
<a href="#"><u>border-right</u></a>	Sets all the right border properties in one declaration
<a href="#"><u>border-right-color</u></a>	Sets the color of the right border
<a href="#"><u>border-right-style</u></a>	Sets the style of the right border
<a href="#"><u>border-right-width</u></a>	Sets the width of the right border
<a href="#"><u>border-style</u></a>	Sets the style of the four borders
<a href="#"><u>border-top</u></a>	Sets all the top border properties in one declaration
<a href="#"><u>border-top-color</u></a>	Sets the color of the top border
<a href="#"><u>border-top-style</u></a>	Sets the style of the top border
<a href="#"><u>border-top-width</u></a>	Sets the width of the top border
<a href="#"><u>border-width</u></a>	Sets the width of the four borders



## Anexo V. Lista de cursores

Valor	Descripción
crosshair	Cursor tipo cruz para tareas en las que se requiere precisión.
help	Interrogación. Cursor de ayuda.
move	Flechas hacia todos lados. Cursor para mover elementos.
pointer	Mano o apuntador. Cursor para hacer clic.
progress	Barra progreso o similar. Cursor que indica que se está trabajando en segundo plano.
text	Cursor que permite seleccionar texto.
wait	Cursor que indica que se debe esperar.
vertical-text	Flecha vertical, utilizado para texto en orientación vertical.
alias	Flecha diagonal, utilizado para indicar que se puede acceder a un enlace o recurso externo.
no-drop	Círculo con una línea diagonal, utilizado para indicar que no se permite soltar el elemento arrastrado.
grab	Mano cerrada, utilizado para indicar que el elemento se puede agarrar
grabbing	Mano abierta, utilizado para indicar que el elemento está siendo arrastrado o agarrado.
zoom-in	Símbolo de lupa con un signo de más, utilizado para indicar que se puede hacer zoom in.
zoom-out	Símbolo de lupa con un signo de menos, utilizado para indicar que se puede hacer zoom out.





not-allowed	Círculo con una línea diagonal, utilizado para indicar que la interacción no está permitida.
cell	Cursor tipo cruz, utilizado en hojas de cálculo para ajustar celdas.
copy	Puntero con un símbolo de copia, utilizado para indicar que se puede realizar una copia de un elemento.
ew-resize	Flecha bidireccional horizontal, utilizado para indicar que se puede ajustar el tamaño horizontalmente.
ns-resize	Flecha bidireccional vertical, utilizado para indicar que se puede ajustar el tamaño verticalmente.
nwse-resize	Flecha bidireccional diagonal, utilizado para indicar que se puede ajustar el tamaño diagonalmente en dirección noroeste-sureste.
nesw-resize	Flecha bidireccional diagonal, utilizado para indicar que se puede ajustar el tamaño diagonalmente en dirección noreste-suroeste.
col-resize	Flecha bidireccional horizontal con una línea vertical, utilizado para indicar que se puede ajustar el tamaño de una columna.
row-resize	Flecha bidireccional vertical con una línea horizontal, utilizado para indicar que se puede ajustar el tamaño de una fila.
context-menu	Puntero con un símbolo de menú, utilizado para indicar que se puede abrir un menú contextual.
none	No se muestra ningún cursor.
Personalized	Imagen o valor personalizado en el cursor.