

Desarrollo Web Frontend



Edición 2021

Universidad Autónoma de Entre Ríos
Facultad de Ciencia y Tecnología - Sede Oro Verde



Display

Floats

Posicionamiento

Flexbox

Grillas



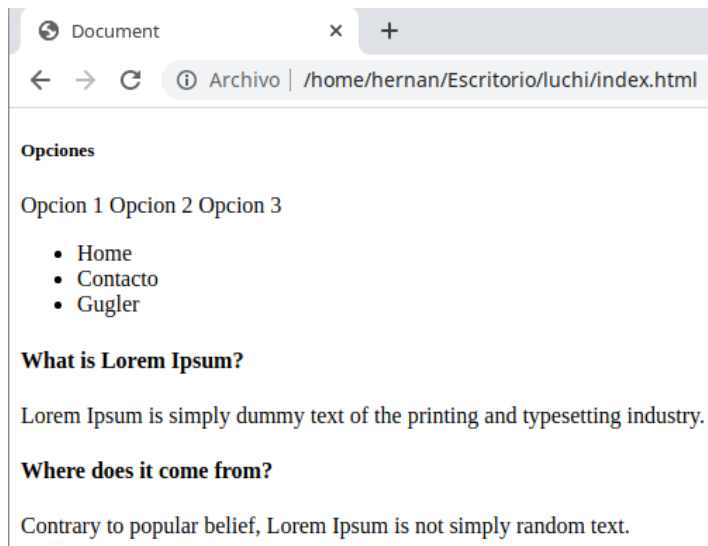


Propiedad display

Esta propiedad permite modificar el comportamiento de los elementos en el flujo normal. Es decir, cambiar la disposición natural (lineal o en bloques) que presenta cada elemento HTML.

Valores:

- **Block:** los elementos con este valor de display se muestran uno debajo de otros.
- **Inline:** los elementos con este valor de display se muestran uno a continuación del otro.
- **Inline-Block:** similar a block, con la diferencia de que admite valores en los atributos **width** y **height**



```
<h5>Opciones</h5>
  <span>Opcion 1</span>
  <span>Opcion 2</span>
  <span>Opcion 3</span>
<ul>
  <li>Home</li>
  <li>Contacto</li>
  <li>Gugler</li>
</ul>

<p>
  <h4>What is Lorem Ipsum?</h4>
  Lorem Ipsum is simply dummy text of the printing and typesetting industry.
</p>

<p>
  <h4>Where does it come from?</h4>
  Contrary to popular belief, Lorem Ipsum is not simply random text.
</p>
```



Propiedad display

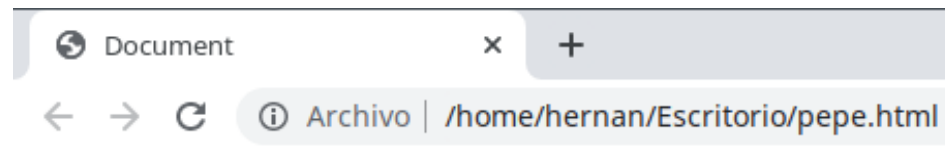
Cada elemento HTML tiene por defecto un valor definido en la propiedad display.

- **display:block** para aquellos elementos en bloque (<p>, <div>, , etc) donde se muestran uno debajo del otro.
- **display:inline** para elementos en línea (, <h1>..<h6>, <a>, etc) donde se muestran uno a continuación del otro.

Sin embargo, para modificar el valor de esta propiedad y que el elemento tome otro valor de disposición veremos el siguiente ejemplo: dos elementos en bloque y dos elementos en línea donde se mostrarán con valores de display invertidos respectivamente.

```
<style>
  p{
    display: inline; /* Valor inicial display:block */
  }
  span{
    display: block; /* Valor inicial display:inline */
  }
</style>
```

```
<body>
  <p>Esto es un elemento en bloque</p>
  <p>Otro elemento en bloque</p>
  <br/><br/>
  <span> Este es un elemento en línea</span>
  <span> Otro elemento en línea</span>
</body>
```



Esto es un elemento en bloque Otro elemento en bloque

Este es un elemento en línea
Otro elemento en línea

Elementos de párrafo en línea

Elementos de texto en línea con disposición en bloques (uno debajo del otro)



~~Display~~

Floats

Posicionamiento

Flexbox

Grillas





Floats

La propiedad **float** permite cambiar el comportamiento de un elemento y de aquellos de nivel de bloque que lo siguen en el flujo normal. Permite hacer un desplazamiento lateral, eliminando el flujo normal, donde el contenido circundante flota alrededor de este elemento.

La propiedad float admite estos cuatro atributos:

- **left:** el elemento flota hacia la izquierda.
- **right:** el elemento flota hacia la derecha.
- **none:** no especifica ningún tipo de flotación. Este es el valor por defecto.
- **inherit:** el valor de la propiedad float se hereda del elemento padre.



Floats (Continuación...)

Veamos un ejemplo:

```
<h2>Titulo de ejemplo</h2>
<div id="caja">
  
</div>

<p>
  <h4>What is Lorem Ipsum?</h4>
  Lorem Ipsum is simply dummy text of the printing and typesetting industry.
</p>

<p>
  <h4>Where does it come from?</h4>
  Contrary to popular belief, Lorem Ipsum is not simply random text.
</p>
```

```
#caja{
  float: left;
  width: 100px;
  height: 100px;
  background-color: darkcyan;
  border: 1px solid rgb(39, 37, 173);
  border-radius: 3px;
}

#foto {
  width: 90px;
  height: 90px;
}
```



Titulo de ejemplo



What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Where does it come from?

Contrary to popular belief, Lorem Ipsum is not simply random text.



Floats (Continuación...)

La propiedad **clear** permite limpiar un elemento flotante en una dirección determinada. Editando uno de los párrafos del ejemplo anterior, vemos que el mismo se situá por debajo al utilizar la propiedad **clear: left**

```
<p id="p1">
  <h4>Where does it come from?</h4>
  Contrary to popular belief, Lorem Ipsum is not simply random text.
</p>
```

```
#p1{
  clear:left;
}
```



Titulo de ejemplo



What is Lorem Ipsum?

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Where does it come from?

Contrary to popular belief, Lorem Ipsum is not simply random text.





~~Display~~

~~Floats~~

Posicionamiento

Flexbox

Grillas





Posicionamiento

El posicionamiento permite administrar y ajustar la posición de los elementos desde el flujo normal hacia otra ubicación. A continuación veremos diversas técnicas de posicionamiento:

- **Posicionamiento estático** (static): cada elemento HTML tiene establecido este valor por defecto. Coloca a los elementos en su posición normal dentro del flujo de diseño (No requiere demostración).
- **Posicionamiento relativo** (relative): permite modificar la posición de un elemento respecto a su posición en el flujo normal, lo que permite superposiciones con otros elementos de la página. Las propiedades **top**, **right**, **bottom** y **left** especifican el desplazamiento que tendrá el elemento.
- **Posicionamiento absoluto** (absolute): mueve al elemento fuera del flujo de diseño normal. Permite fijar una posición relativa respecto a su bloque contenedor. Las propiedades **top**, **right**, **bottom** y **left** especifican el desplazamiento que tendrá el elemento.
- **Posicionamiento fijo** (fixed): similar al posicionamiento absoluto, excepto que corrige la posición de un elemento relativo a la ventana del navegador y no sobre otro elemento HTML.

Veamos algunos ejemplos...





Posicionamiento (Continuación...)

Posicionamiento relativo

```
<h1>Titulo principal</h1>

<p>Este es un ejemplo de párrafo</p>
<p class="positioned">Este párrafo estará posicionado.</p>
<p>Este es un ejemplo de párrafo.</p>
```

```
body {
  width: 500px;
  margin: 0 auto;
}

p {
  background-color: ■ rgb(207,232,220);
  border: 2px solid ■ rgb(79,185,227);
  padding: 10px;
  margin: 10px;
  border-radius: 5px;
}

.positioned {
  position: relative;
  top: 30px;
  left: 30px;
  background: ■ rgba(255,84,104,.3);
  border: 2px solid ■ rgb(255,84,104);
}
```

x +

ome/hernan/Escritorio/Practica/index.html

Titulo principal

Este es un ejemplo de párrafo

Este párrafo estará posicionado.

Este es un ejemplo de párrafo.

En este ejemplo **top** y **left** desplazan al elemento con posición relativa respecto al flujo normal de diseño.

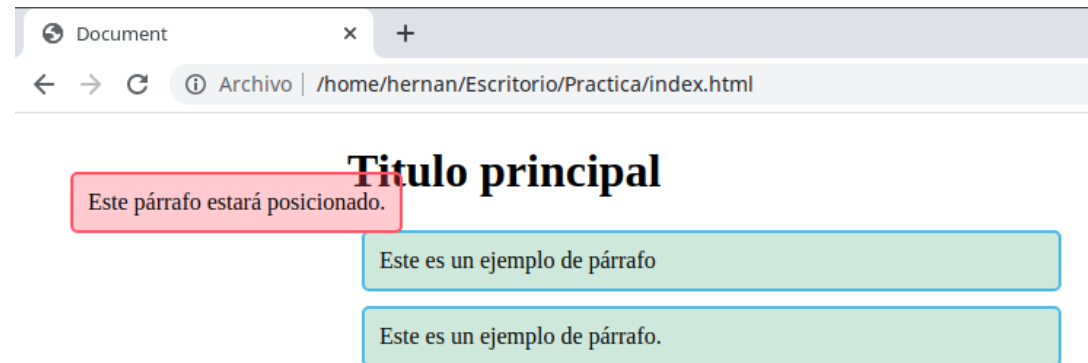


Posicionamiento (Continuación...)

Posicionamiento absoluto

```
< <h1>Titulo principal</h1>
< <p>Este es un ejemplo de párrafo</p>
< <p class="positioned">Este párrafo estará posicionado.</p>
< <p>Este es un ejemplo de párrafo.</p>
```

```
.positioned {
  position: absolute;
  top: 30px;
  left: 30px;
  background: rgba(255,84,104,.3);
  border: 2px solid rgb(255,84,104);
}
```



El elemento se ubica por fuera del flujo normal. En este caso las propiedades **top** y **left** posicionan al elemento en relación al borde de su elemento contenedor. (Para este ejemplo sería <html>)



Posicionamiento (Continuación...)

Posicionamiento fijo

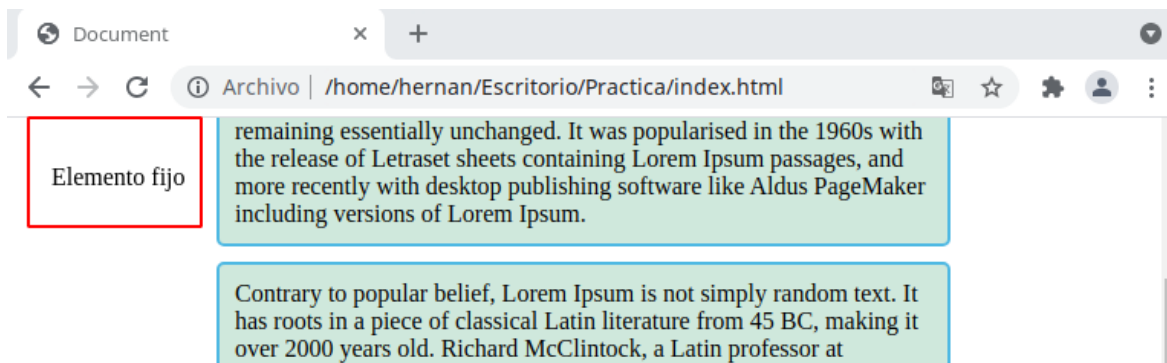
```
<h1>What is Lorem Ipsum?</h1>
<div class="positioned">Elemento fijo</div>

<p>
  Lorem Ipsum is simply dummy text of the printing and typesetting
  industry. Lorem Ipsum has been the industry's standard dummy text
  ever since the 1500s, when an unknown printer took a galley of
  type and scrambled it to make a type specimen book. It has
  survived not only five centuries, but also the leap into
  electronic typesetting, remaining essentially unchanged. It was
  popularised in the 1960s with the release of Letraset sheets
  containing Lorem Ipsum passages, and more recently with desktop
  publishing software like Aldus PageMaker including versions of
  Lorem Ipsum.
</p>
<p> ...
</p>
```

```
body {
  width: 500px;
  margin: 0 auto;
}

p {
  background-color: ■ rgb(207,232,220);
  border: 2px solid ■ rgb(79,185,227);
  padding: 10px;
  margin: 10px;
  border-radius: 5px;
}

.positioned {
  position: fixed;
  top: 30px;
  left: 30px;
}
```



En este caso el elemento se posiciona de manera relativa a la ventana del navegador, definido por **top** y **left**. Al hacer *scroll* el contenido se desplaza, pero el elemento se mantiene “fijo” en la misma ubicación.



~~Display~~

~~Floats~~

~~Posicionamiento~~

Flexbox

Grillas





Flexbox

Es el **modulo de diseño de cajas flexibles** y nos permite distribuir los elementos HTML en una dimensión, en filas o columnas.

Como se utiliza?

Debemos aplicar la propiedad **display:flex;** sobre el elemento padre de aquellos elementos que deseamos distribuir. Todos los elementos hijos se convierten en elementos flexibles.

Por que utilizar flexbox?

Resulta muy útil cuando queremos obtener un diseño de página sencillo y flexible que cumpla los siguientes requisitos:

- Centrar verticalmente un bloque de contenido dentro de su elemento padre.
- Permitir que todos los elementos secundarios (hijos) de un contenedor ocupen la misma cantidad de espacio (ancho/alto).
- Permitir que columnas múltiples adopten la misma altura independientemente su contenido.



Flexbox (Continuación...)

Si bien se tratan de elementos de bloque que se mostrarían uno por debajo del otro, al convertirse en elementos **flexibles** se ven afectados por ciertos valores iniciales del contenedor.

```
<header>
  <h1>Ejemplo flexbox</h1>
</header>
<article>
  <section>
    Lorem Ipsum is simply dummy text of the printing and
    typesetting industry. Lorem Ipsum has been the industry's
    standard dummy text ever since the 1500s
  </section>
  <section>
    It has survived not only five centuries, but also the leap
    into electronic typesetting, remaining essentially unchanged.
    It was popularised in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages
  </section>
  <section>
    It is a long established fact that a reader will be distracted
    by the readable content of a page when looking at its layout.
    The point of using Lorem Ipsum is that it has a more-or-less
    normal distribution of letters, as opposed to using 'Content
    here
  </section>
</article>
```

```
<style>
  article{
    display: flex;
    background-color: yellow;
  }
</style>
```

Ejemplo flexbox

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here

El resultado son columnas múltiples de igual tamaño y altura.





Flexbox (Continuación...)

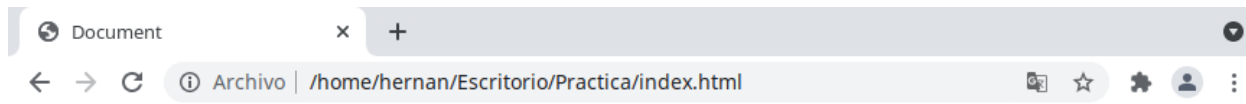
Propiedad flex-direction: Los elementos flexbox poseen esta propiedad que define en qué dirección están dispuestos los elementos hijos de un elemento flexbox.

Su valor por defecto siempre es **row** (fila), por lo que los elementos hijos se presentan en esa dirección.

Si utilizamos el mismo código HTML del ejemplo anterior, pero ahora agregando la propiedad **flex-direction: column;** los elementos hijos de <article> se mostraran nuevamente en bloques uno debajo del otro.

```
<style>
  article{
    display: flex;
    flex-direction: column;
  }

  section{
    margin-top: 5px;
    background-color: yellow;
  }
</style>
```



Ejemplo flexbox

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here





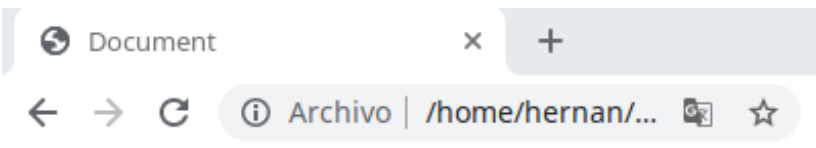
Flexbox (Continuación...)

Además de utilizar valores como **row** y **column**, flex-direction admite mostrar en filas o columnas en manera invertida, a través de **row-reverse** o **column-reverse**.

```
<style>
  article{
    display: flex;
    flex-direction: column-reverse;
  }

  section{
    margin-top: 5px;
    background-color: yellow;
  }
</style>
```

```
<header>
  <h1>Ejemplo flexbox</h1>
</header>
<article>
  <section>
    Caja 1
  </section>
  <section>
    Caja 2
  </section>
  <section>
    Caja 3
  </section>
</article>
```



Ejemplo flexbox

Caja 3
Caja 2
Caja 1





Flexbox (Continuación...)

Si tenemos definida una cantidad fija de ancho o alto puede que los hijos de un elemento flexbox desborden el contenedor. Para solucionar esto podemos hacer uso de la propiedad **flex-wrap: wrap;**

Veamos el siguiente ejemplo:

```
<header>
  <h1>Ejemplo flexbox</h1>
</header>
<article>
  <section>
    Lorem Ipsum is simply dummy text of the printing and
    typesetting industry. Lorem Ipsum has been the industry's
    standard dummy text ever since the 1500s, when an unknown
    printer took a galley of type and scrambled it to make a type
    specimen book. It has survived not only five centuries, but
    also the leap into electronic typesetting, remaining
    essentially unchanged. It was popularised in the 1960s
  </section>
  <section>...
</section>
  <section>...
</section>
  <section>
    Lorem Ipsum is simply dummy text of the printing and
    typesetting industry. Lorem Ipsum has been the industry's
    standard dummy text ever since the 1500s, when an unknown
    printer took a galley of type and scrambled it to make a type
    specimen book. It has survived not only five centuries, but
    also the leap into electronic typesetting, remaining
    essentially unchanged. It was popularised in the 1960s

    It is a long established fact that a reader will be distracted
    by the readable content of a page when looking at its layout.
  </section>
</article>
```



Obliga el desplazamiento horizontal para ver todos los elementos

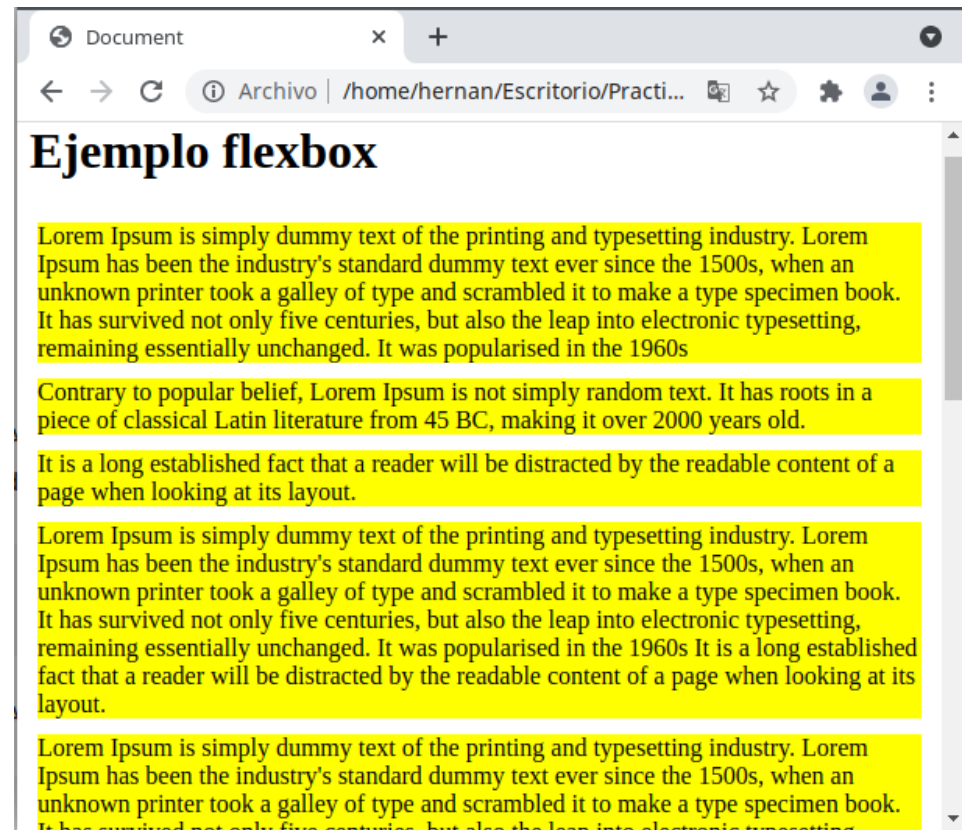


Flexbox (Continuación...)

Ejemplo continuación:

```
<style>
  article{
    display:flex;
    flex-wrap: wrap;
  }

  section{
    margin: 5px;
    background-color: yellow;
  }
</style>
```



Ahora existen varias filas y en cada fila puede haber tantos elementos hijos de un elemento flexbox como sea necesario.





Flexbox (Continuación...)

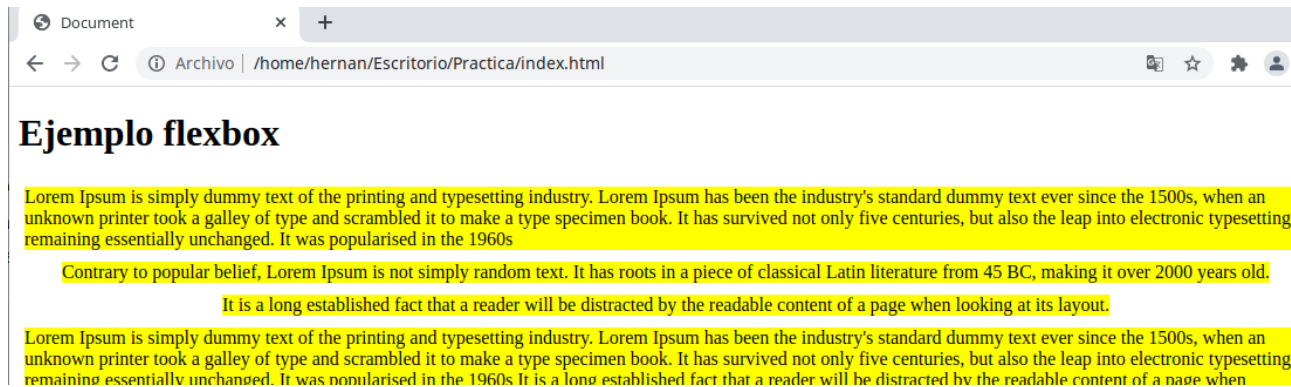
Alineación horizontal y vertical

Podemos utilizar funciones de flexbox para alinear elementos flexibles sobre el eje principal. Para esto hacemos uso de **align-items**, que admite los siguientes valores:

- **stretch**, es el valor por defecto que toma esta propiedad. Ensancha los elementos flexibles para rellenar el elemento primario.
- **center** centra el contenido del elemento flexible sobre el eje principal.
- **flex-start / flex-end**, alinean todos los elementos al inicio o final del eje transversal.

```
<style>
  article{
    display:flex;
    flex-direction: column;
    flex-wrap: wrap;
    align-items: center;
  }

  section{
    margin: 5px;
    background-color: yellow;
  }
</style>
```





~~Display~~

~~Floats~~

~~Posicionamiento~~

~~Flexbox~~

Grillas





Grillas

El diseño de cuadrícula está diseñado para distribuir elementos en dos dimensiones: alinear filas y columnas.

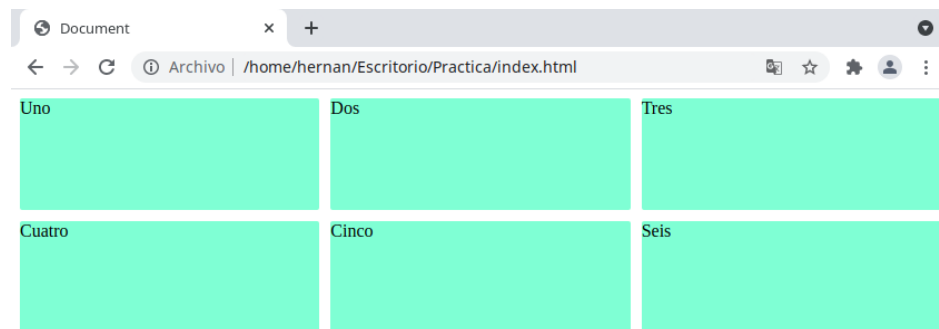
Para activar el diseño en cuadrícula definiremos un contenedor padre con el atributo **display:grid**;

Además utilizaremos propiedades como **grid-template-rows** y **grid-template-columns** que permiten definir el tamaño de la fila como la cantidad de elementos por columnas respectivamente.

```
<style>
.wrapper {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 100px 100px;
  grid-gap: 10px;
}
</style>
```

```
<div class="wrapper">
  <div class="box1">Uno</div>
  <div class="box2">Dos</div>
  <div class="box3">Tres</div>
  <div class="box4">Cuatro</div>
  <div class="box5">Cinco</div>
  <div class="box6">Seis</div>
</div>
```

(1fr equivale a una fracción del espacio disponible – máximo 12 fracciones por fila)



Hemos definido tres columnas flexibles de 1fr cada una y dos filas de 100px y una separación de celdas con la propiedad **grid-gap**.



Grillas

También podemos posicionar elementos explícitamente en nuestra cuadrícula, en lugar de confiar en el comportamiento de colocación automática que vimos anteriormente.

Veamos el siguiente ejemplo:

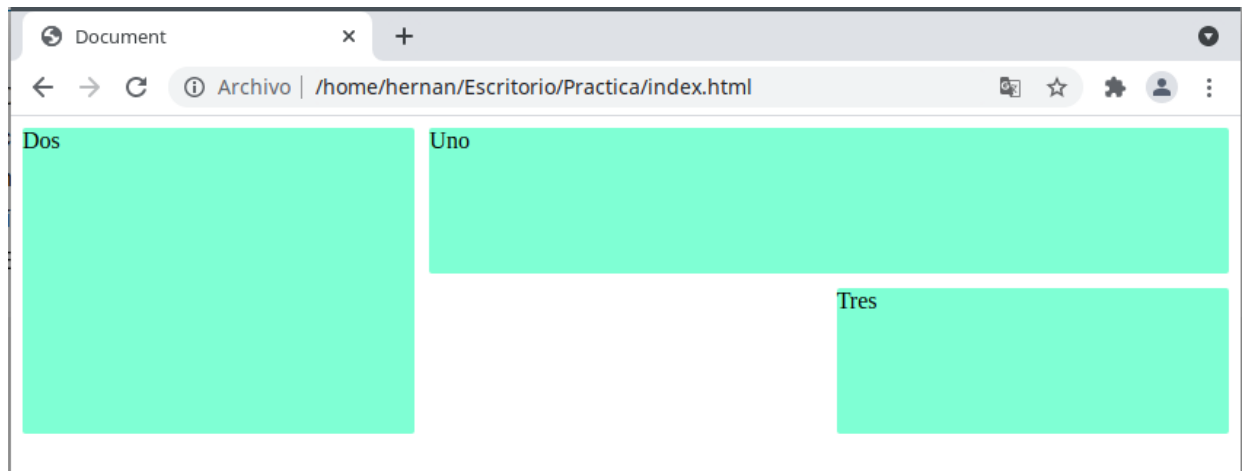
```
<div class="wrapper">
  <div class="box1">Uno</div>
  <div class="box2">Dos</div>
  <div class="box3">Tres</div>
</div>
```

```
<style>
.wrapper {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 100px 100px;
  grid-gap: 10px;
}

.box1 {
  grid-column: 2 / 4;
  grid-row: 1;
}

.box2 {
  grid-column: 1;
  grid-row: 1 / 3;
}

.box3 {
  grid-row: 2;
  grid-column: 3;
}
</style>
```



La propiedad **grid-column** permite definir que cantidad de columnas ocupara el elemento. Mientras que **grid-row** establece la cantidad de filas que serán ocupadas por el elemento.

¿Consultas?

