



Argentina
programa
4.0

CSS

“Introducción a la Programación Web”

Introducción a CSS

CSS: Cascading Style Sheets, es decir, Hojas de Estilo en Cascada

Es un lenguaje creado para controlar el aspecto o presentación de los elementos de HTML.

Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es .css. Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

Formas de aplicar CSS

- **Interno**

- **Inline CSS**

Se crea utilizando la propiedad style en los elementos del HTML.

Por ejemplo:

```
<body style="background-color: bisque;">  
<p style="text-align: center;"> </p>  
<p style="text-align: justify;"> </p>
```

- **Internal CSS**

Reconoce la etiqueta como un elemento. Aplica propiedades a todos aquellos que

Tengan ese elemento. Por ejemplo:

Agrego en el HEAD:

```
<style>  
p{  
  text-align:justify ;  
  font-weight: bold ;  
}  
</style>
```

Etiqueta de enlace – External CSS

Para enlazar un archivo .css se utiliza la etiqueta <link> de la siguiente manera:

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

rel: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`

type: indica el tipo de recurso enlazado. Para los archivos CSS su valor siempre es `text/css`

href: indica la URL del archivo CSS que contiene los estilos.

Esta etiqueta se debe colocar en el head de nuestro html.

Estructura básica

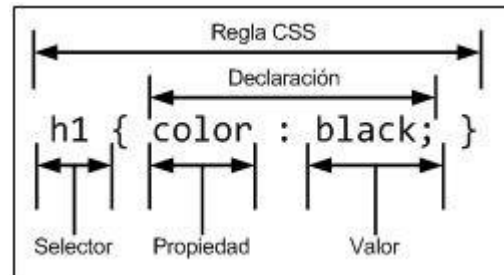
Regla: cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte llamada "selectores", un símbolo de "llave de apertura" ({), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" (}).

Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS. El selector indica "a quién hay que hacérselo"

Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS. La declaración indica "qué hay que hacer"

Propiedad: característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.

Valor: establece el nuevo valor de la característica modificada en el elemento.



Tipos de selectores

Selector universal:

Sintaxis: `* { propiedad:valor; }`

Por ejemplo: ` { color:orange; }`*

Selector de etiqueta:

Sintaxis: `etiqueta { propiedad:valor ; }`

Por ejemplo: `p { text-align:center; font-weight: bold; }`

Tipos de selectores

Selector descendente:

Selecciona los elementos que se encuentran dentro de otros elementos. Están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el cual se aplican los estilos y todos los selectores anteriores indican el lugar en el que se encuentra el elemento.

Su sintaxis es: selector1 selector2... selectorN { propiedad:valor }. **Siendo el selector N el elemento sobre el que se aplica el estilo.**

Por ejemplo:

```
ul li { background-color: #d7c2a7 }
```

Tipos de selectores

Selector de clase:

Sintaxis: `.clase { propiedad:valor ; }`

or ejemplo:

```
<ul class="lista">  
  .lista { font-style: italic;}
```

Selector de identificación:

Utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento con un id dado en un documento.

Sintaxis: `id { propiedad:valor ; }`

Por ejemplo:

```
<h3 id="enlaces">  
#enlaces{ font-style: italic; }
```


Tipos de selectores

Selectores combinados:

Nos permite dar un estilo a todos los selectores indicados.

Sintaxis: selector1, selector2, selector3 { propiedad: valor; }

Por ejemplo:

```
h1, h2, h3 { background-color: #d7c2a7 }
```

Herencia

Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

```
body {color: blue; }
```

Todos los elementos de la página tendrían color de letra azul.

Cascada

El orden de las reglas de CSS importa: cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS.

```
h1 {  
  color: red;  
}  
h1 {  
  color: blue;  
}
```

El h1 va a ser de color azul, ya que es la última regla que se aplicó.

Modelo de cajas

En CSS todo elemento está enmarcado en una caja.

Hay dos tipos de cajas: cajas en bloque y cajas en línea. Estas características se refieren al modo como se comporta la caja con otras.

Las cajas **en bloque** siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta el final de la línea.

Las cajas **en línea** sólo ocupan el espacio necesario para mostrar su contenido. Los párrafos (p) por ejemplo son elementos de bloque y los links (a) son de línea.

Este comportamiento se puede modificar con la propiedad [display de css.](#)

Modelo de cajas

Los valores más comunes que puede tomar la propiedad `display` son:

block: hace que el comportamiento del elemento sea como un bloque.

inline: hace que los elementos se muestren en la misma línea, ajustándose al contenido y sin tener en cuenta el ancho, alto o márgenes de la caja.

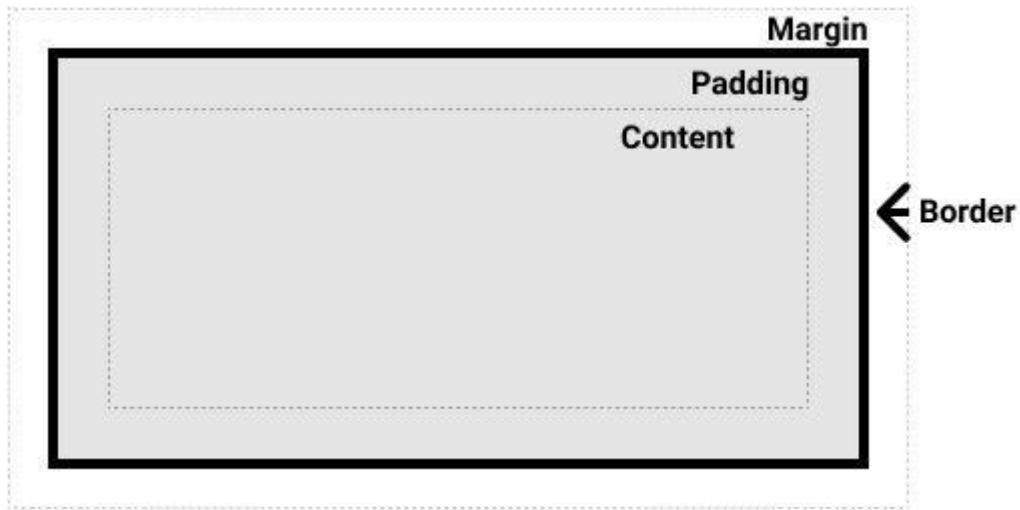
inline-block: hace que los elementos se muestren en la misma línea respetando el ancho, el alto y los márgenes.

flex: hace que los elementos hijos se comporten de manera flexible.

none: oculta un elemento.

Modelo de cajas

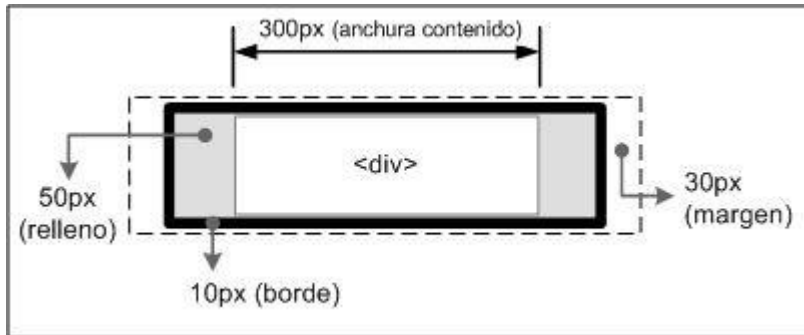
Una caja esta conformada de la siguiente manera:



Modelo de cajas

Por defecto, el ancho y el alto de un elemento no solamente se calculan teniendo en cuenta sus propiedades `width` y `height`. El `margin`, el `padding` y `border` también suman al cálculo final.

```
div {  
  width: 300px;  
  padding-left: 50px;  
  padding-right: 50px;  
  margin-left: 30px;  
  margin-right: 30px;  
  border: 10px solid black;  
}
```



Ancho total de la caja: $30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480$ píxel

Modelo de cajas

Esto se puede modificar a través de la propiedad **box-sizing** que acepta los siguientes 2 valores:

content-box: comportamiento por defecto para el tamaño de la caja.

border-box: el padding y border de ese elemento no incrementan su ancho. Si se define un elemento con un ancho de 100 pixeles, estos incluirán cualquier border o padding que se añadan, y la caja de contenido se encogerá para absorber ese ancho extra.

Propiedades de texto

font-family

Establece la familia tipográfica. Hay dos grandes grupos, sans-serif y serif

Ejemplo de serif

```
font-family: "Times New Roman", Times, serif;
```

Ejemplo de sans serif

```
font-family: Arial, Helvetica, sans-serif;
```

font-size

Establece el tamaño de fuente (px, em o %)

font-weight

Establece el peso de la fuente

Valores: normal | bold | light

Valor por defecto: normal

font-style

Se utiliza para especificar si la fuente es normal o itálica.

Valores: normal | italic

Propiedades de texto

text-decoration

Establece la decoración del texto (subrayado, tachado).

Valores: none | underline | line-through | overline

Valor por defecto: none

text-transform

Transforma el texto original, a mayúsculas, minúsculas, etc.

Valores: none | capitalize | uppercase | lowercase

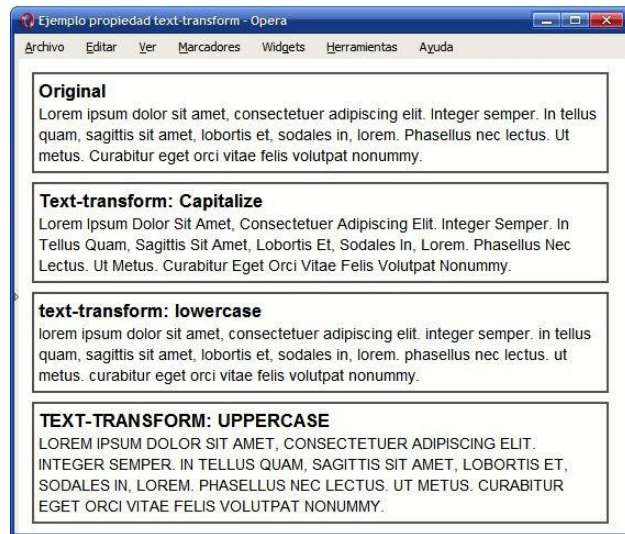
Valor por defecto: none

letter-spacing

Permite establecer el espacio entre las letras que forman las palabras del texto (interletrado)

Valores: normal | unidad de medida | inherit

Valor por defecto: normal



Propiedades color y background

color

Establece el color del texto

background-color

Establece el color de fondo

Valores: color | transparent

Valor por defecto: transparent

background-image

Establece la imagen de fondo

Valores: url | none

Valor por defecto: none

background-repeat

Establece la imagen de fondo

Valores: repeat | repeat-x

(horizontalmente) | repeat-y

(verticalmente) | no-repeat

Valor por defecto: repeat

background-attachment

Establece la fijación del fondo

Valores: scroll | fixed

Valor por defecto: scroll

background-position

Establece la imagen de fondo

Valores: valor | top | center |

bottom || left | center | right

Valor por defecto: 0% 0%

Otras propiedades

width

Establece el ancho del elemento

Valores: medida | inherit

Valor inicial: 0

height

Establece el alto del elemento

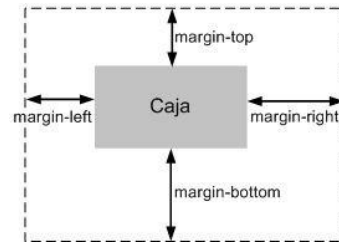
Valores: medida | inherit

Valor inicial: 0

margin

Permite especificar el espacio entre los elementos

Valores: medida | auto

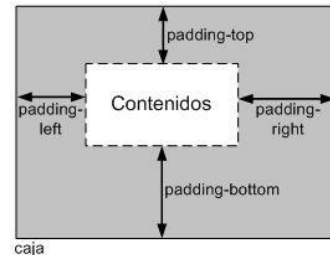


padding

Establece cada uno de los rellenos horizontales y verticales de un elemento

Valores: medida | inherit

Valor inicial: 0



Propiedad position

Existe una propiedad llamada **position** que sirve para posicionar un elemento dentro de la página. Dependiendo de cual sea la propiedad que usemos, el elemento tomará una referencia u otra para posicionarse respecto a ella.

Los posibles valores que puede adoptar la propiedad position son: **static** | **relative** | **absolute** | **fixed**

Propiedad position

position: static

Es el valor que toma un elemento por defecto para posicionarse. No tendrá en cuenta los valores para las propiedades top, left, right y bottom.

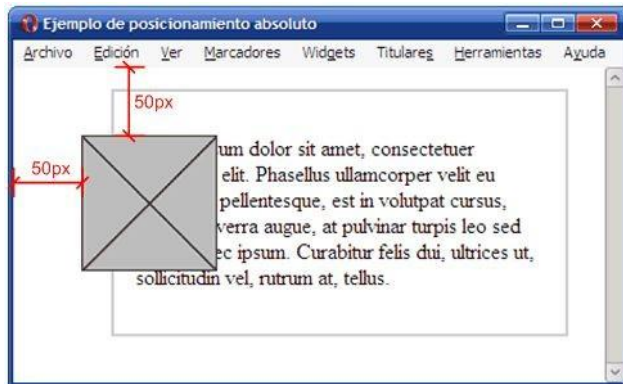
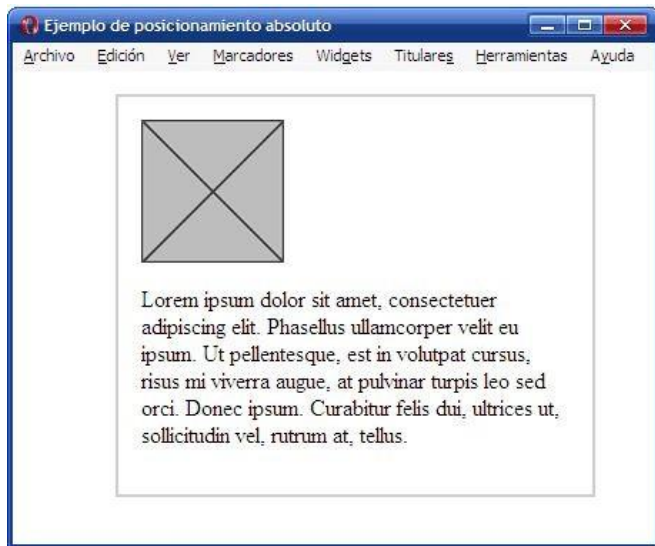
position: relative

Se comporta de la misma manera que static a menos que le agreguemos las propiedades top, left, right y bottom. Esto causará que su posición normal se reajuste, pudiendo causar solapamiento entre los distintos elementos.

position: absolute

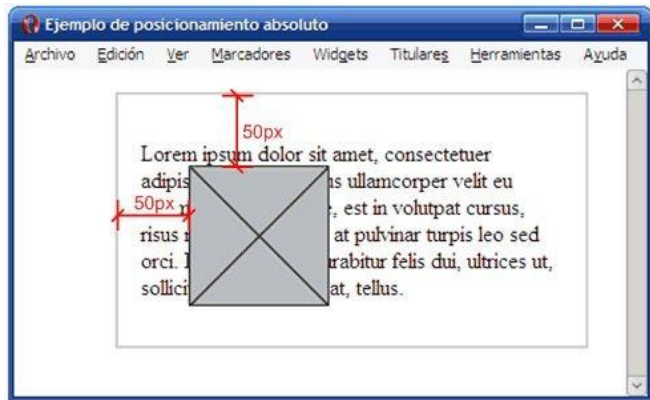
Cuando se posiciona un elemento de manera absoluta, se hace en base a su elemento contenedor (por lo general el cuerpo del documento, o un elemento posicionado relativamente que lo contenga).

Propiedad position



En este caso la imagen tiene position absolute y toma como referencia la ventana del navegador.

Propiedad position



Al ponerle al div contenedor position relative, la imagen lo toma como referencia para su posicionamiento.

Propiedad position

position: fixed

Este atributo sirve para posicionar un elemento como si tuviera posicionamiento absoluto, pero su posición final será siempre fija, es decir, aunque se desplace el documento con las barras de desplazamiento del navegador, siempre aparecerá en la misma posición.

position: sticky

Es una posición nueva que todavía no está soportada por todos los navegadores. Su comportamiento va a ser relativo hasta que al hacer scroll llegamos al elemento con position sticky y en ese momento pasa a tener un comportamiento fijo.

Propiedad z-index

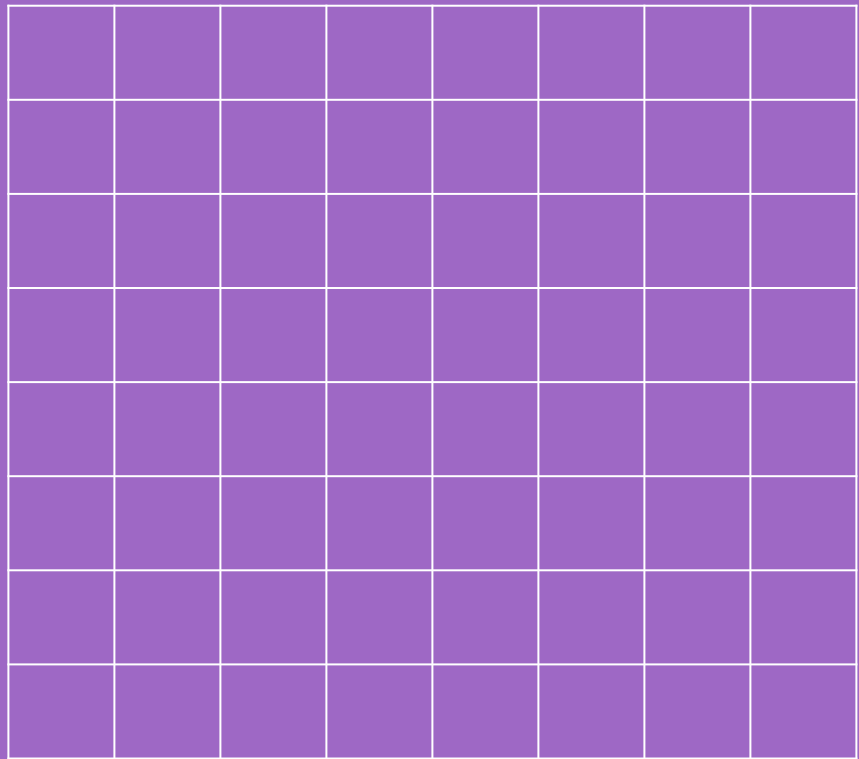
Cuando varios elementos se superponen, z-index determina cual está por encima del otro. Un elemento con mayor z-index se ubica por encima de uno con z-index menor. Esta propiedad se puede aplicar solamente en los elementos que poseen algún tipo de posición que no sea static.

Referencias

- <https://code.tutsplus.com/es/tutorials/the-30-css-selectors-you-must-memorize-net-16048>
- <https://cssreference.io/>
- <https://css-tricks.com/almanac/properties/>
- <https://www.quackit.com/css/properties/>
- <https://developer.mozilla.org/es/docs/Web/CSS/position#Examples>
- <https://www.w3schools.com/css/exercise.asp>



Flexbox



¿Qué es flexbox?

Es un sistema de elementos flexibles que se caracterizan por su habilidad de alterar su ancho y alto para ajustarse lo mejor posible al espacio disponible en cualquier dispositivo.

Este tipo de modelo flexbox, nos permite controlar además parámetros tales como la alineación, dirección de los elementos (horizontal/vertical), entre otros.

Conceptos básicos

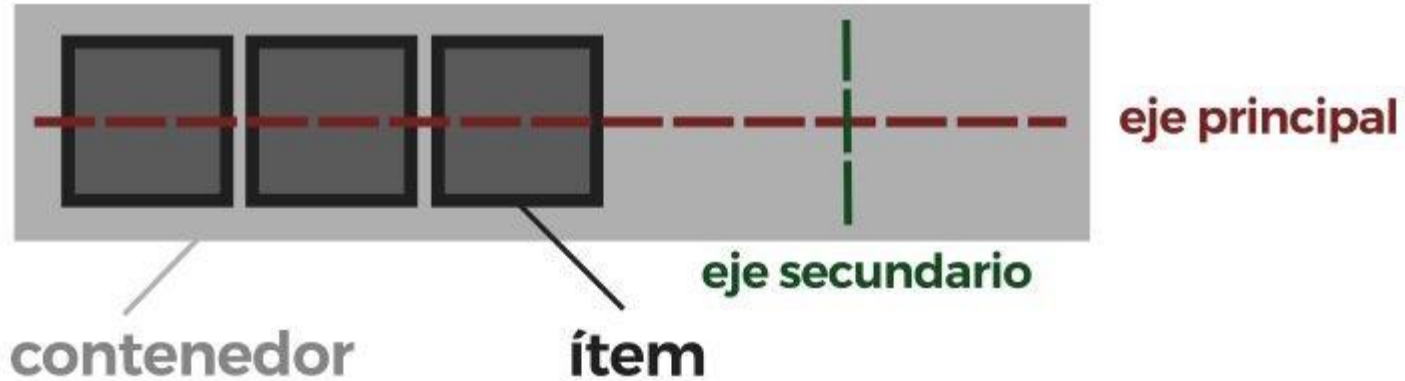
Para empezar a utilizar flexbox lo primero que debemos hacer es conocer algunos de los elementos básicos:

Contenedor flexible (Flex container)

El elemento "padre" que contiene cada uno de los ítems flexibles. Un contenedor flexible se define usando los **valores flex o inline-flex en la propiedad display**.

- **Eje principal:** Los contenedores flexibles tendrán una orientación principal específica. Por defecto, es en horizontal (en fila).
- **Eje secundario:** De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical, y viceversa.

Conceptos básicos



Conceptos básicos

Ejemplo en código:

```
1 <div class="contenedor">
2   <div class="elemento">1</div>
3   <div class="elemento">2</div>
4   <div class="elemento">3</div>
5   <div class="elemento">4</div>
6   <div class="elemento">5</div>
7   <div class="elemento">6</div>
8   <div class="elemento">7</div>
9 </div>
```

```
1 .contenedor{
2   display: flex;
3 }
4 .elemento{
5   width: 25%;
6 }
```

1

2

3

4

5

6

7

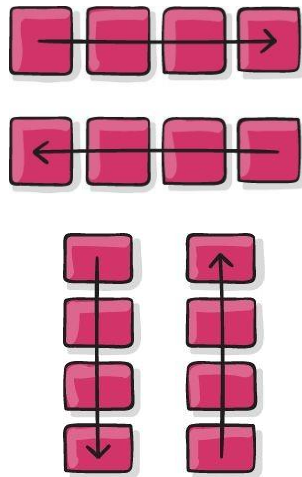
Al no haber definido aún el comportamiento de dirección y tamaño que tendrán los elementos de nuestro contenedor, aunque hayamos definido un ancho de 25%, éstos se adaptan a su padre ocupando el 100% de ancho entre la suma de todos.

Dirección de los ejes

Propiedad flex-direction

Permite modificar la dirección del eje principal del contenedor para que se oriente en horizontal (por defecto) o en vertical. Además, también podemos incluir el sufijo `-reverse` para indicar que coloque los ítems en orden inverso.

- **flex-direction: row;** | Los elementos se visualizan de izquierda a derecha (valor por defecto)
- **flex-direction: row-reverse;** | Los elementos se visualizan de derecha a izquierda.
- **flex-direction: column;** | Los elementos se visualizan de arriba hacia abajo.
- **flex-direction: column-reverse;** | Los elementos se visualizan de abajo hacia arriba.

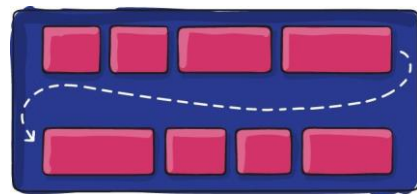


Dirección de los ejes

Propiedad flex-wrap

Permite especificar el comportamiento del contenedor respecto a evitar que se desborde (nowrap, valor por defecto) o permitir que lo haga, en cuyo caso, estaríamos hablando de un contenedor flexbox multilínea, es decir los elementos se distribuyen en varias filas.

- **flex-wrap: nowrap;** | Establece los elementos en una sola línea (no permite que se desborde el contenedor).
- **flex-wrap: wrap;** | Los elementos se muestran en una sola línea, pero si su ancho supera la del contenedor, se distribuyen en varias filas.
- **flex-wrap: wrap-reverse;** | Su comportamiento es igual al flex-wrap: wrap pero en dirección inversa.



Dirección de los ejes

Atajo / short hand

Existe una propiedad llamada **flex-flow** que permite resumir los valores de las propiedades flex-direction y flex-wrap en una sola propiedad:

```
.container {  
  /* flex-flow: <flex-direction> <flex-wrap>; */  
  flex-flow: row wrap;  
}
```

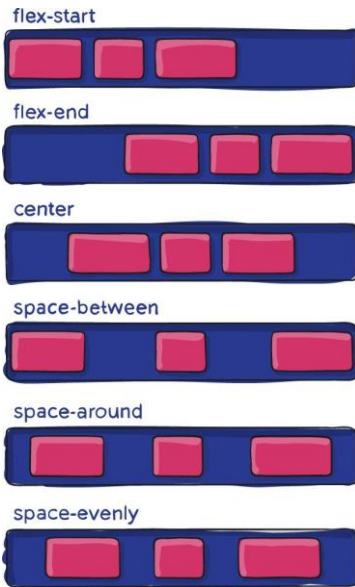
Propiedades de alineación

Sobre el eje principal

Existe distintas propiedades dentro de flexbox para disponer los ítems dependiendo de nuestro objetivo.

justify-content: Se utiliza para alinear los ítems del eje principal (por defecto, el horizontal). Y puede tomar los siguiente valores:

- **flex-start:** Agrupa los ítems al principio del eje principal.
- **flex-end:** Agrupa los ítems al final del eje principal.
- **center:** Agrupa los ítems al centro del eje principal.
- **space-between:** Distribuye los ítems dejando el máximo espacio para separarlos.
- **space-around:** Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).
- **space-evenly:** Distribuye los ítems dejando el mismo espacio (solapado) a izquierda y derecha.



Propiedades de alineación

Sobre el eje principal

La propiedad **align-content** actúa sobre cada una de las líneas de un contenedor multilínea (no tiene efecto sobre contenedores de una sola línea). Los valores que puede tomar son los siguientes:

- **flex-start:** Agrupa los ítems al principio del eje principal.
- **flex-end:** Agrupa los ítems al final del eje principal.
- **center:** Agrupa los ítems al centro del eje principal.
- **stretch:** Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
- **space-between:** Distribuye los ítems desde el inicio hasta el final.
- **space-around:** Distribuye los ítems dejando el mismo espacio a los lados de cada uno.

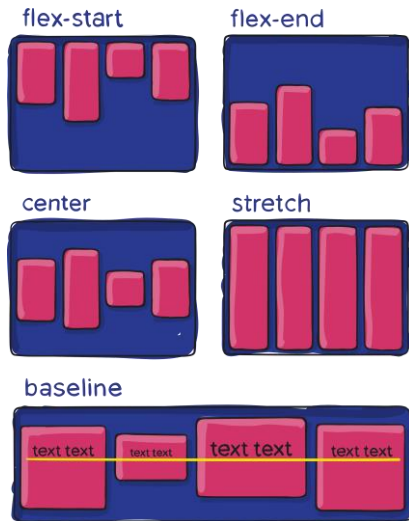


Propiedades de alineación

Sobre el eje secundario

La propiedad **align-items**, se encarga de alinear los ítems en el eje secundario del contenedor. Los valores que puede tomar son los siguientes:

- **flex-start:** Alinea los ítems al principio del eje secundario.
- **flex-end:** Alinea los ítems al final del eje secundario.
- **center:** Alinea los ítems al centro del eje secundario.
- **stretch:** Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
- **baseline:** Alinea los ítems en el contenedor según la base del contenido de los ítems del contenedor.

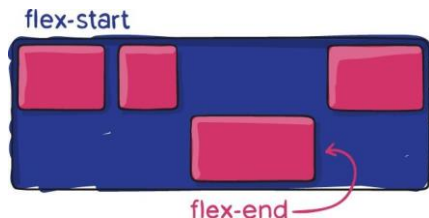


Propiedades de alineación

Sobre el eje secundario

La propiedad **align-self** actúa exactamente igual que align-items, sin embargo se utiliza sobre un ítem hijo específico y no sobre el elemento contenedor. Los valores que puede tomar son los mismos que align-items:

- **flex-start:** Alinea los ítems al principio del eje secundario.
- **flex-end:** Alinea los ítems al final del eje secundario.
- **center:** Alinea los ítems al centro del eje secundario.
- **stretch:** Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
- **baseline:** Alinea los ítems en el contenedor según la base del contenido de los ítems del contenedor.
- **auto:** Hereda el valor de align-items del padre (si no se ha definido, es stretch).

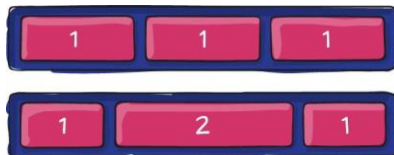


Propiedades de hijos

Todas las propiedades vistas se aplican sobre el elemento contenedor. Las siguientes propiedades, se aplican sobre los ítems hijos.

- **flex-grow:** 0 | Número que indica el factor de crecimiento del ítem respecto al resto.
- **flex-shrink:** 1 | Número que indica el factor de decrecimiento del ítem respecto al resto.
- **flex-basis:** size - content | tamaño por defecto que tendrán los ítems antes de aplicarle la distribución de espacio. Generalmente, se aplica un tamaño (unidades, porcentajes, etc...), pero también se puede aplicar la palabra clave content que ajusta automáticamente el tamaño al contenido del ítem, que es su valor por defecto.
- **order:** 0 | Número que indica el orden de aparición de los ítems.

flex-grow



Referencias

- <https://medium.com/@alexcamachogz/una-gu%C3%ADa-completa-de-flexbox-768b038de5e9>
- https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceptos_Basicos_de_Flexbox
- <https://flexboxfroggy.com/#es>
- <https://developer.mozilla.org/es/docs/Web/CSS/position#Examples>



**Argentina
programa
4.0**

Gracias!
