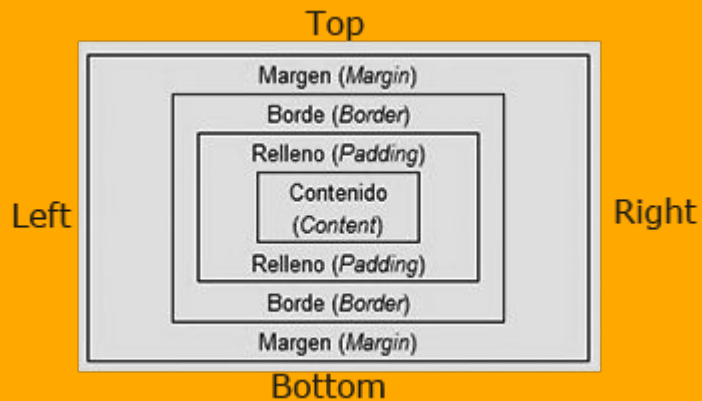




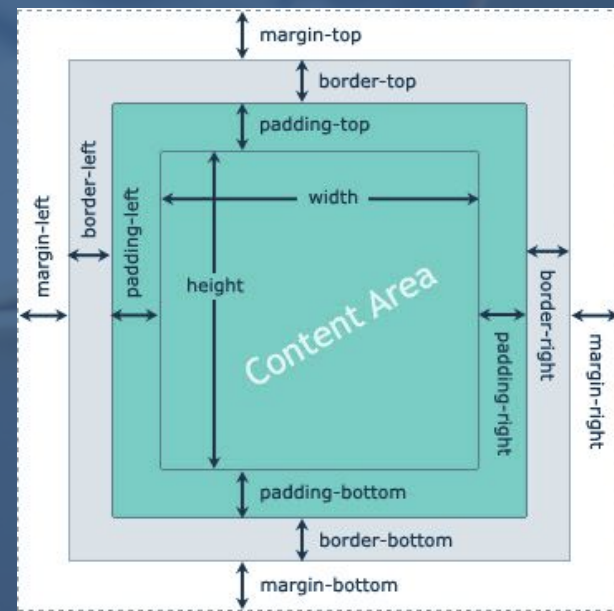
CSS Avanzado

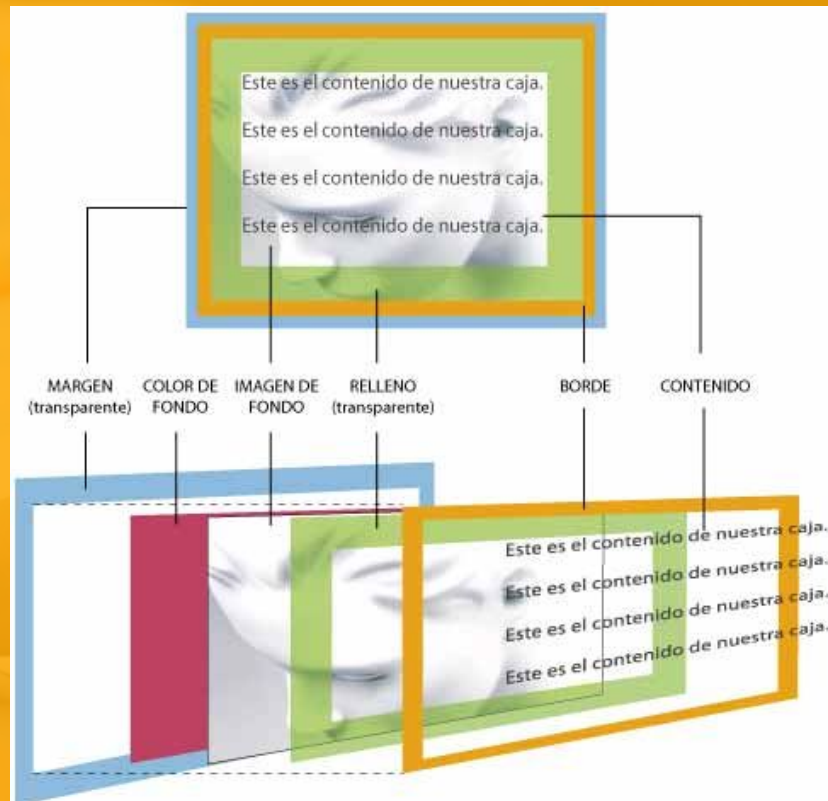
Modelo de Caja

- Todo en CSS tiene una caja alrededor.
- Controla altura, anchura, margen, etc.
- Se pueden ocultar, desplazar, fijarlas en una posición concreta dentro del documento, etc.



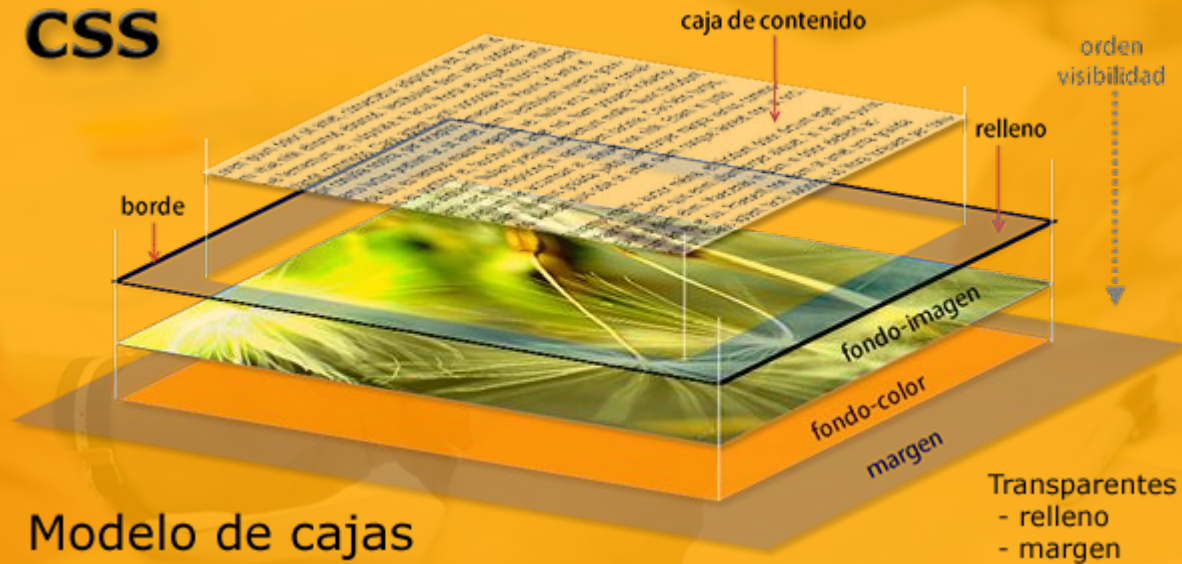
Modelo de Caja





Modelo de Caja

CSS



Modelo de cajas

Modelo de Caja

Bordes

- **border-style** especifica el tipo de borde.
- **border-width** especifica el ancho del borde. Se puede usar las unidades vistas (in px, cm, etc) o **thin**, **medium**, o **thick**.
- **border-color** setea el color.
- Todas las propiedades se pueden setear para lados específicos, como **border-top-color**.
- Con **border** se puede setear el ancho, estilo (requerido) y el color de una sola vez.
- **border-radius** permite redondear los bordes.

[Estilos de bordes](#)

[Ancho de borde](#)

[Colores de bordes](#)

[Lados del borde](#)

[Modo abreviado](#)

[Borde redondeado](#)

Margen y Relleno

- **margin** crea espacio alrededor de los elementos, fuera de los bordes definidos.
- **padding** genera espacio alrededor del contenido de un elemento, dentro de los bordes definidos.
- Posibilidad de especificar cada lado (-top, -bottom, -left y -right).
- La versión abreviada permite 4 valores (arriba, derecha, abajo e izquierda), 3 (arriba, abajo y los costados), 2 (cada eje) y uno (todos igual).

Margin

Margin (colapso)

Padding

Altura y anchura

- **height** (alto) y **width** (ancho) se utilizan para establecer altura y anchura de un elemento.
- No incluyen relleno, bordes y márgenes.
- **min-width**, **min-height**, **max-width** y **max-height** setean mínimos y máximos del ancho y alto (ideal para responsivo).
- Con **box-sizing: border-box** se especifica que el padding y el border deben ser incluidos en el total del ancho y alto del elemento.

Dimensiones

box-sizing

Centrar verticalmente

Elementos de bloque y de línea (inline)

- Conforman la principal clasificación.
- Los elementos de bloque empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de su contenedor. Pueden contener otros elementos.
Ej.: p, h1..h6, ul, ol y **div**.
- Los elementos de línea sólo ocupan el espacio necesario para mostrar sus contenidos. Solo pueden contener otros elementos en línea.
Ej.: img, a, b y **span**.
- **display** permite cambiar la forma de mostrar aunque no el tipo de elemento.

Display

div

span

Más sobre display

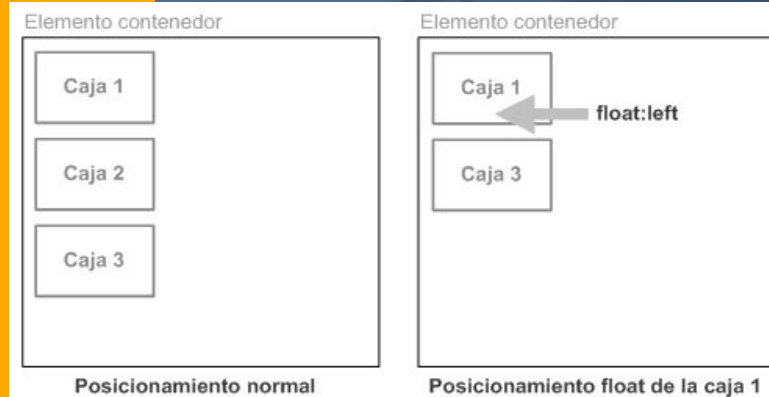
```
display: none;  
/*permite ocultar  
un elemento*/
```

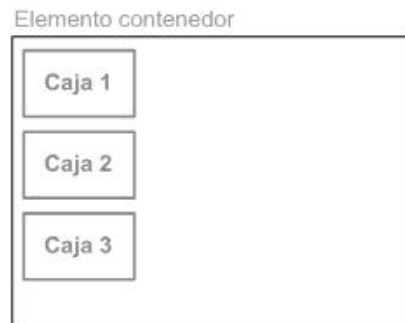

Float

- **float** permite sacar a un elemento del flujo normal del documento.
- El elemento puede ser posicionado a la izquierda o derecha de su misma línea o contenedor tanto como sea posible.

Float

Ejemplos

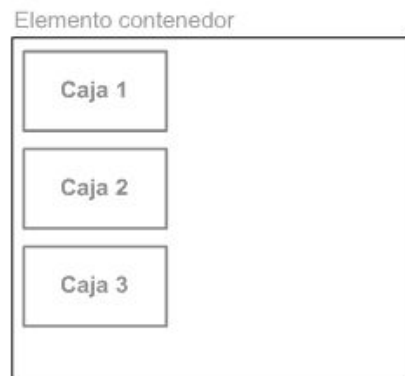




Posicionamiento normal



Posicionamiento float de las 3 cajas



Posicionamiento normal



Posicionamiento float de las 3 cajas

Flotación de divs de tamaño fijo

Float

- Admite tres valores, right, left y none.
- Fuerza la propiedad display del elemento a block.
- Los elementos que se encuentran a continuación de una caja flotantes adaptan sus contenidos para que fluyan (rodeen) alrededor del elemento.
- Los elementos anteriores al flotante no cambian su comportamiento.
- Debe aplicarse sobre elementos con un ancho definido explícitamente o implícitamente.

Float

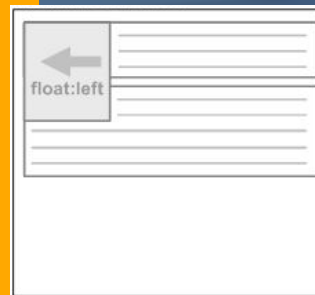
Ejemplos

Clear

- La propiedad **clear** se utiliza conjuntamente con float para forzar a que un elemento no fluya alrededor de otro elemento posicionado mediante float.
- Sus valores posibles son:
 - left: no permite flotantes a su izquierda
 - right: no permite flotantes a su derecha
 - both: no permite flotantes a ambos lados
 - none: permite otros flotantes a ambos lados

Clear

DEMO



Posicionamiento float de la imagen



Párrafo "limpio" con clear:left



```
#paginacion {  
  border: 1px solid #ccc;  
  background-color: #e0e0e0;  
  padding: 0.5em;  
}  
.derecha {  
  float: right;  
}  
.izquierda {  
  float: left;  
}  
div.clear {  
  clear: both;  
}
```

```
<div id="paginacion">  
  <span class="izquierda">&laquo; Anterior</span>  
  <span class="derecha">Siguiente &raquo;</span>  
  <div class="clear"></div>  
</div>
```

Elemento contenedor



Los elementos posicionados vacían de contenidos al <div>

Elemento contenedor



El <div> añadido "limpia" el float y fuerza la altura del <div> original



Ejemplo de flotación y clear

Position

- Especifica el tipo de posicionamiento.
- Sus posibles valores son:
 - static: valor por defecto.
 - relative: posicionado relativo al contenedor.
 - fixed: posicionado relativo al viewport*.
 - absolute: posicionado relativo al ancestro posicionado** más cercano.
 - sticky: posicionado relativo a la posición de desplazamiento del usuario.
- Luego, los elementos posicionados** se colocan utilizando las propiedades **top**, **bottom**, **left**, y **right**.

Position

*se refiere a la porción de ventana visualizada

**cualquier valor menos static

Visualización

- CSS define cuatro propiedades para controlar la visualización.
- Display, visibility, overflow y z-index.
- Con **visibility**:hidden; se puede invisibilizar un elemento.
- El navegador crea la caja del elemento pero no la muestra. El resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.
- Los valores para la propiedad son: visible, hidden y inherit.

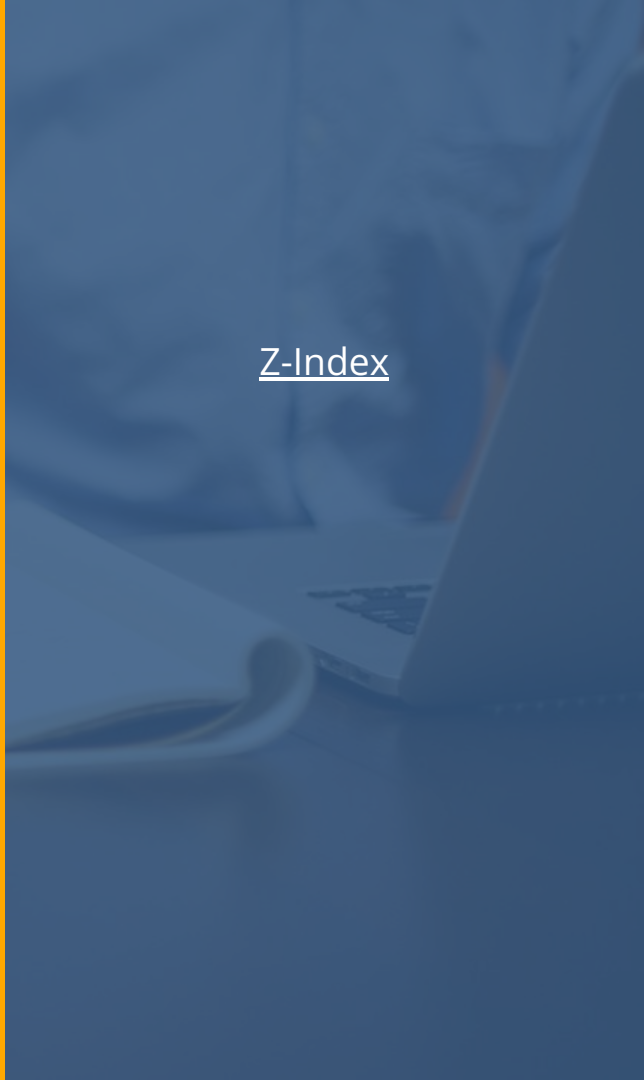
Visibility

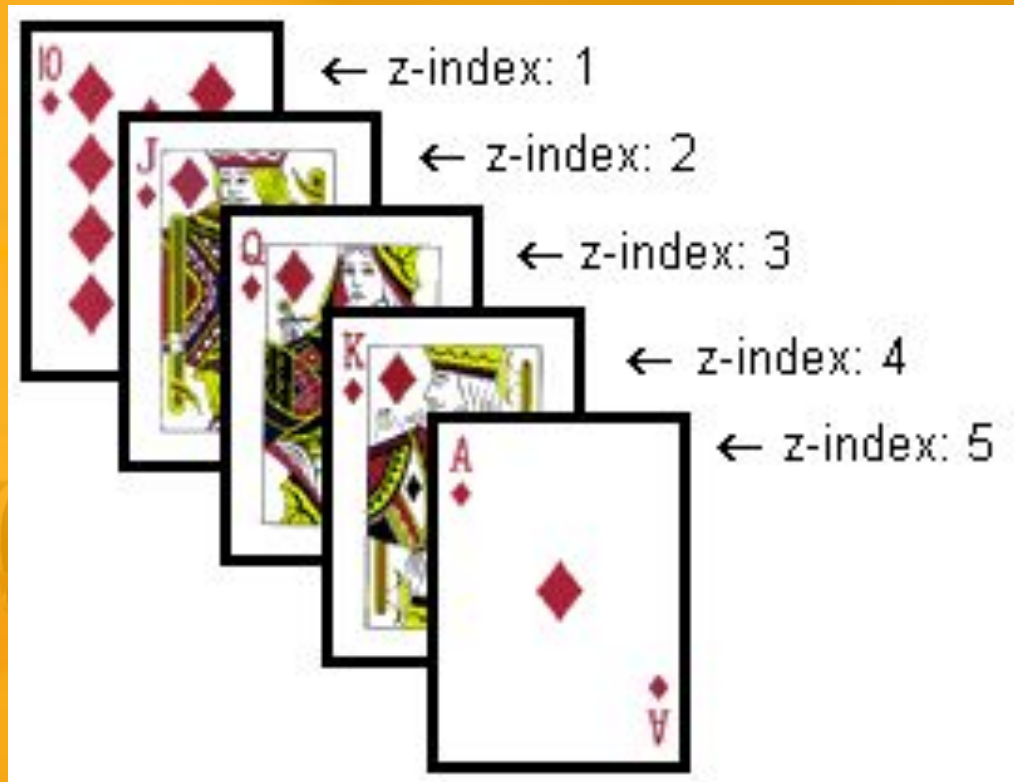
display:none vs visibility:hidden

z-index

- Cuando los elementos se colocan, pueden superponerse a otros elementos.
- z-index especifica el orden de apilamiento de un elemento (qué elemento debe colocarse delante o detrás de los demás).
- Un elemento puede tener un orden de pila positivo o negativo.
- Solo funciona en elementos posicionados o elementos flex.
- Si dos elementos tienen el mismo z-index, el último elemento definido en el código HTML se mostrará en la parte superior.

Z-Index





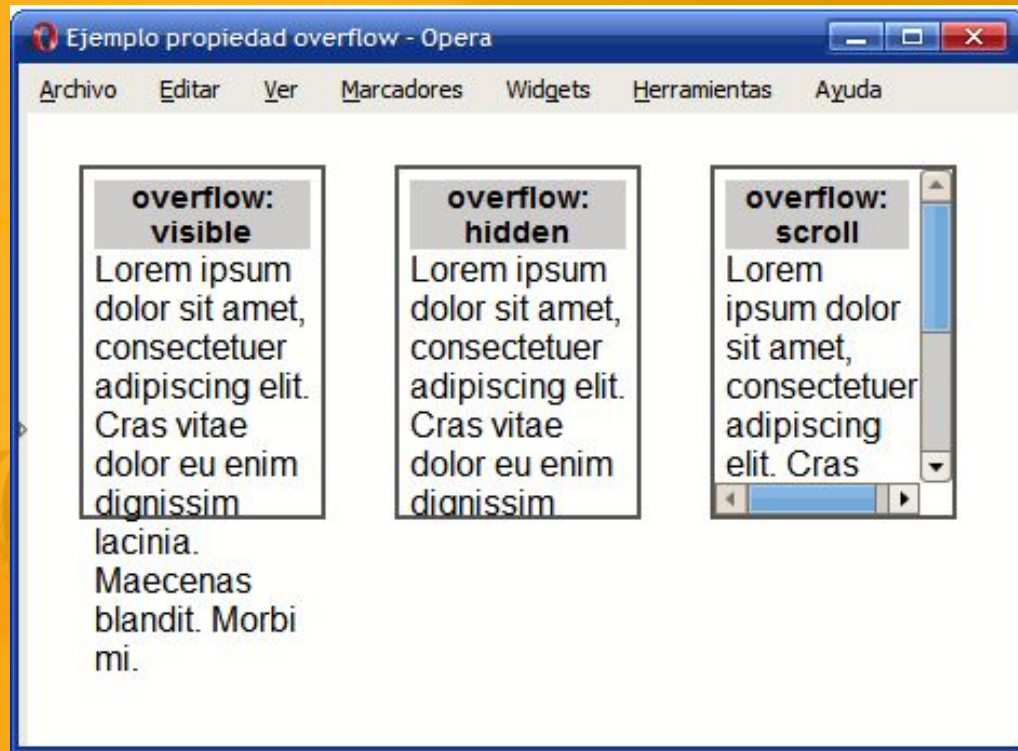
Ejemplo de z-index

Overflow

- Controla qué sucede cuando los elementos sobrepasan el límite del contenedor.
- Normalmente cuando el contenedor tiene seteada las propiedades width y/o height.
- Sus posibles valores son:
 - visible: desborde sin recortes, se renderiza fuera del contenedor. (Default).
 - hidden: desborde recortado y ocultado.
 - scroll: desborde recortado pero se muestran barras de desplazamiento.
 - auto: desborde recortado pero se muestran barras de desplazamiento solo si es necesario.

Overflow





Ejemplo de overflow

Media query

- CSS2 introdujo @media para definir diferentes estilos para diferentes tipos de medios.
- CSS3 amplió la idea de los tipos de medios. En lugar de buscar un tipo de dispositivo, analizan la capacidad del dispositivo.
- Las consultas de medios se pueden usar para verificar muchas cosas, como:
 - ancho y alto de la ventana gráfica
 - ancho y alto del dispositivo
 - orientación (horizontal o vertical)
 - resolución

[Media Queries](#)

[Referencia MDN](#)

[Ejemplos](#)

```
@media not|only mediatype and (expressions) {  
    CSS-Code;  
}
```

Sintaxis de @media

- @media consta de un tipo de medio y puede contener una o más expresiones, que se resuelven en verdadero o falso.
- Da verdadero cuando coincide el tipo y demás expresiones sobre el dispositivo. En ese caso se aplican las reglas definidas.
- Si usa **not** u **only**, se debe especificar el tipo de medio. Caso contrario **all** estará implícito.
- También aplica para vincular css para diferentes medios:
`<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">`

[Media Queries](#)

[Referencia MDN](#)

[Ejemplos](#)

Flexbox

- El módulo de diseño de Flexbox facilita el diseño de una estructura de diseño flexible y receptiva sin usar flotación o posicionamiento.
- Para comenzar a usar el modelo Flexbox, primero debe definir un contenedor flexible estableciendo su propiedad **display** con el valor flex.

```
.flex-container {  
  display: flex;  
}
```

[Flexbox](#)

[Contenedor Flexbox](#)

[Items Flex](#)

[Flexbox adaptativo](#)

Flex

- Contenedor flexible (Flex container): el elemento "padre" que contiene los elementos flexibles. Un contenedor flexible se define usando los valores flex o inline-flex en la propiedad display.
- Elemento flexible (Flex item): cada hijo de un contenedor flex se convierte en un elemento flexible. Si hay texto directamente incluido en el contenedor flexible, se envuelve automáticamente en un elemento flexible anónimo.

[Flexbox](#)

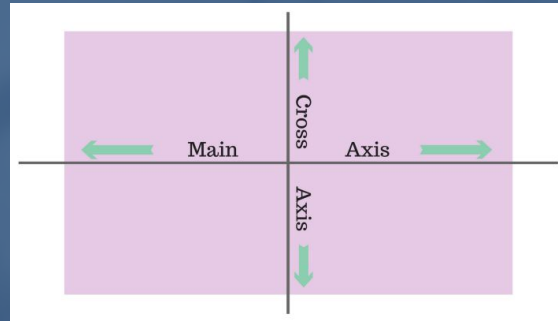
[Contenedor Flexbox](#)

[Items Flex](#)

[Flexbox adaptativo](#)

Contenedor Flexbox

- flex-direction: define en qué dirección el contenedor quiere apilar los elementos flexibles (row, column y variantes -reverse).
- flex-wrap: especifica si ajusta los elementos flexibles (wrap, wrap-reverse y nowrap).
- flex-flow: (flex-direction + flex-wrap).
- justify-content: se utiliza para alinear los elementos flexibles (eje main).
- align-items: se utiliza para alinear los elementos flexibles (eje cross).
- align-content: similar al anterior pero alinea las líneas.



Flexbox

Contenedor Flexbox

Items Flex

Flexbox adaptativo

Explicación gráfica

Ítems Flexbox

- Son los hijos directos de un contenedor flexbox.
- order: especifica el orden.
- flex-grow: especifica cuánto crecerá un elemento flexible en relación con el resto de los elementos flexibles.
- flex-shrink: similar al anterior pero al achicar.
- flex-basis: especifica la longitud inicial.
- flex: flex-grow + flex-shrink + flex-basis,
- align-self: especifica la alineación del elemento.

[Flexbox](#)

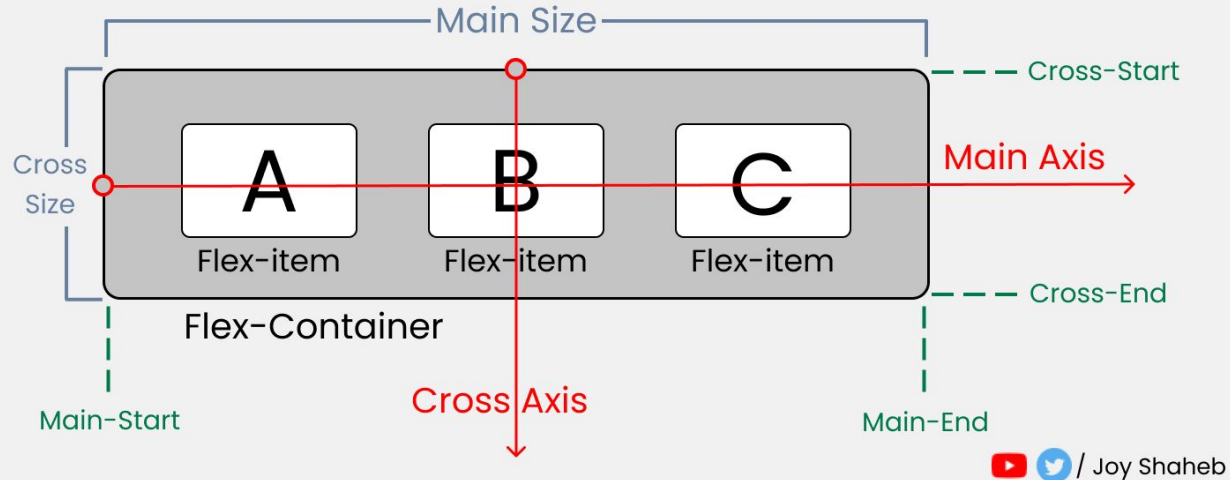
[Contenedor Flexbox](#)

[Items Flex](#)

[Flexbox adaptativo](#)

[Explicación gráfica](#)

FlexBox Architecture



Flexbox ([fuente](#))

Flex Box Chart

Property

Value(s)

#1 display	flex
#2 flex-direction	row column column-reverse row-reverse
#3 justify-content	flex-start flex-end center space-between space-around space-evenly
#4 align-items	flex-start flex-end center stretch baseline
#5 align-content	flex-start flex-end center stretch space-between space-around
#6 align-self	auto flex-start flex-end center baseline stretch
#7 Order	/* Any positive Value */
#8 flex-grow	/* Any positive Value */
#9 flex-shrink	/* Any positive Value */
#10 flex-wrap	nowrap wrap wrap-reverse

  / Joy Shaheb

Flexbox ([fuente](#))

Más recursos

- Estos apuntes no cubren todo lo que se puede lograr.
- Todo el tiempo siguen saliendo nuevas características.
- Repasar conocimientos.

[W3Schools](#)

[Referencia CSS](#)

[MDN](#)

[Acceso Rápido](#)

[How to](#)

A man with glasses is looking down at a smartphone in his hands. He is wearing a dark t-shirt and a watch. The background is a brick wall. The entire image is overlaid with a semi-transparent blue filter.

¡Gracias!

¿Preguntas?

N
ACTI