

## POSICIONAMIENTO

- Tipos:
  - o Estático: por defecto. La posición que toca.
  - o Relativo: Desplazar la caja desde su posición original.
  - o Absoluto: Desplazar la caja desde su elemento contenedor. El resto de elementos ignoran esta caja y ocupan su hueco.
  - o Fijo: Convierte la caja en un elemento inamovible en una posición independientemente del resto de elementos.
- Propiedad:
  1. Especificar position: static|relative|absolute|fixed
  2. Especificar el desplazamiento:
    - o top
    - o right
    - o bottom            longitud | porcentaje | auto
    - o left

### A) Posicionamiento estático

EJEMPLO: 1 POSICIONAMIENTO

- Por defecto.
- Teniendo en cuenta las características del elemento:
  - o Línea: se muestra detrás del anterior elemento de forma horizontal desde la posición más a la izquierda de su elemento contenedor.  
Márgenes laterales delimitan el espacio con el elemento contiguo  
Listado:  
[https://developer.mozilla.org/es/docs/Web/HTML/Elementos\\_en\\_l%C3%A1nea](https://developer.mozilla.org/es/docs/Web/HTML/Elementos_en_l%C3%A1nea)
  - o Bloque: se muestra debajo del anterior y el que va detrás suyo se pone debajo.  
Márgenes verticales delimitan el espacio con el elemento de encima/debajo.  
Listado: [https://developer.mozilla.org/es/docs/Web/HTML/Block-level\\_elements](https://developer.mozilla.org/es/docs/Web/HTML/Block-level_elements)
- El posicionamiento del elemento siempre se referencia desde su contenedor padre.
  - o Si los elementos línea no caben en el espacio disponible en una línea, se desplazan a la siguiente.
  - o Si ocuparan menos del espacio disponible, se pueden alinear con text-align.

### B) Posicionamiento relativo

EJEMPLO: POSICIONAMIENTO ABSOLUTO Y RELATIVO - PADRE CONTENEDOR - MUCHOS ELEMENTOS

- Desplaza la caja respecto su posición original. No afecta a las cajas de al lado (respetan su lugar, aunque puede haber solapamientos al desplazarse ella)
- Propiedad
  - position: relative
  - top|right|left|bottom: unidad

## C) Posicionamiento absoluto

[EJEMPLO: POSICIONAMIENTO ABSOLUTO Y RELATIVO - PADRE CONTENEDOR - MUCHOS ELEMENTOS](#)

- El resto de elementos se desplazan para ocupar su lugar. Puede haber solapamientos.
- Origen de coordenadas: esquina superior izquierda de su primer elemento contenedor posicionado. (se puede añadir position: relative() aunque no se mueva el elemento)

## D) Posicionamiento fijo

- Su posición es inamovible en la pantalla. Siempre está en pantalla

[EJEMPLO: 5POSICIONAMIENTO FIXED - 6POSICIONAMIENTO FIXED RESPECTO DIV RELATIVE](#)

## E) Prioridad de Capas: Z-INDEX

- Solapamiento de capas
- A > z-index, + cerca del usuario
- Sólo tiene efecto en los elementos posicionados (deben llevar asociado un position)

## FLOAT

- Permite flotar las cajas a la derecha o a la izquierda.
- La caja flotada se coloca lo más a la izquierda o a la derecha y el resto de elementos se colocan a su alrededor, en el sitio disponible. (como posicionamiento absoluto)
- Es apropiado para colocar imágenes alrededor de texto.
- También tenemos la propiedad clear para romper el flujo de flotación. (las cajas siguientes no ocupan su espacio, como posicionamiento relativo)
- Propiedad
  - o float: left | right | none
  - o clear: none | left | right | both

[EJEMPLO: ./POSICIONAMIENTO/7POSICIONAMIENTO FLOAT](#)

## DISPLAY

- Modifica el modo natural de comportarse del elemento (bloque o línea)

### EJEMPLO: 1 VISUALIZACIÓN

- Propiedad:

display: inline | block | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none | flex

- o **inline:** visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate. Sólo ocupa el espacio necesario para mostrar sus contenidos.
  - o **block:** muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. Siempre empiezan en una nueva línea y siempre ocupan todo el espacio disponible en la línea, aunque sus contenidos no ocupen todo el sitio.
  - o **inline-block:** Una caja de tipo inline-block se comporta como si fuera de bloque (conserva anchura y altura), pero respecto a los elementos que la rodean es una caja en línea.
- EJEMPLO: 2 VISUALIZACION DISPLAY BLOCK Y INLINE

- o **none** oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página ocupan el espacio en el que se debería visualizar el elemento ocultado.
- o **list-item:** Hace que cualquier elemento de cualquier tipo se muestre como si fuera un elemento de una lista (elemento <li>).  
Se pueden utilizar las propiedades de listas como list-style-type, list-style-image, list-style-position y list-style
- o **run-in:** genera una caja de bloque o una caja en línea dependiendo del contexto, es decir, dependiendo de sus elementos adyacentes. El comportamiento de las cajas de tipo run-in se rige por las siguientes reglas:
  - Si la caja run-in contiene una caja de bloque, la caja run-in se convierte en una caja de bloque.
  - Si después de la caja run-in se encuentra una caja de bloque (que no esté posicionada de forma absoluta y tampoco esté posicionada de forma flotante), la caja run-in se convierte en una caja en línea en el interior de la caja de bloque.
  - En cualquier otro caso, la caja run-in se convierte en una caja de bloque.

#### EJEMPLO: DISPLAY LIST-ITEM Y RUN-IN

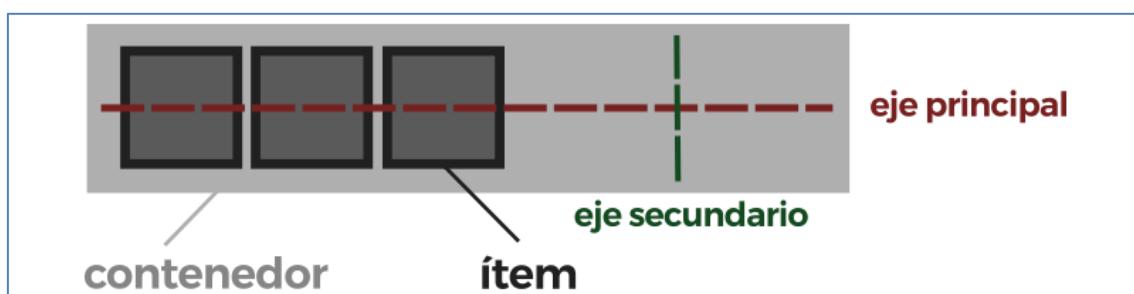
- **table:** Permite asignar un comportamiento de tabla a los elementos que haya dentro (anidados a la etiqueta donde estás colocando el display: table).
- **table-row:** Actúa como un elemento fila, etiqueta TR.
- **table-cell:** Este valor se lo colocarías a las etiquetas que deben trabajar como una celda de tabla, etiqueta TD.
- **table-caption:** se comportan como la etiqueta CAPTION
- **table-column:** se comportan como la etiqueta COL
- **table-column-group:** se comportan como la etiqueta COLGROUP
- **table-footer-group:** se comportan como la etiqueta TFOOT
- **table-header-group:** comportamiento como la etiqueta THEAD
- **table-row-group:** comportamiento como la etiqueta TBODY

#### EJEMPLO: DISPLAY TABLE-ROW-CELL

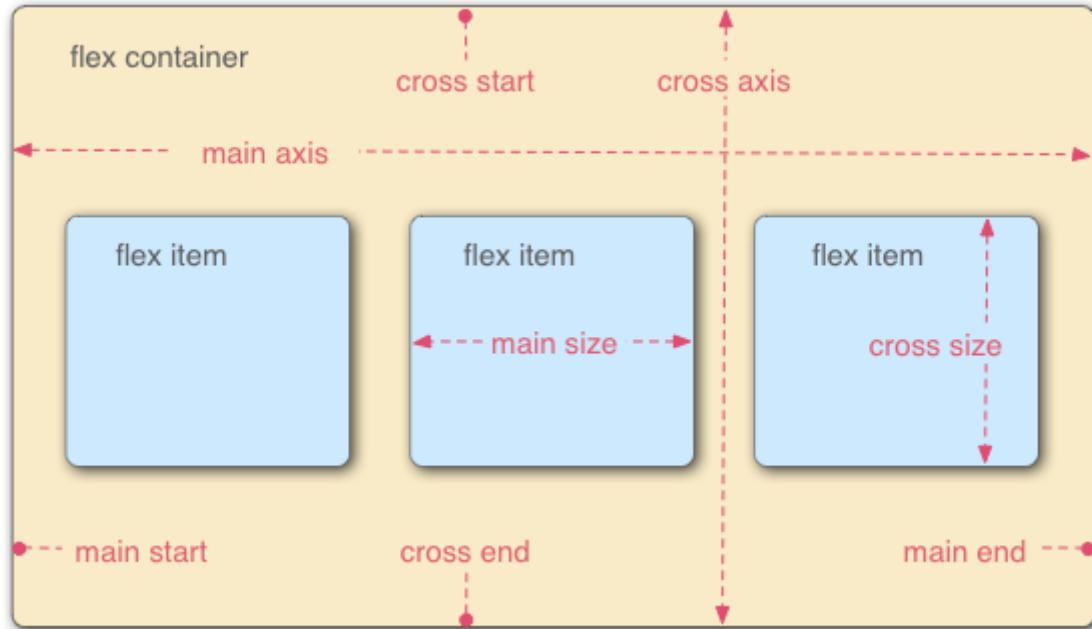
- **display: inline-table:** Sería igual que un display: table con la diferencia que en este caso se comportaría como un elemento "inline".  
Es decir, mientras que una tabla normal actúa como un bloque, con este elemento puedes hacer que se comporte como un elemento "inline".

## DISPLAY: flex

Lo que caracteriza un diseño flexible es su habilidad para alterar el ancho y alto de sus elementos para ajustarse lo mejor posible al espacio disponible en cualquier dispositivo. Un contenedor flexible expande sus elementos para llenar el espacio libre, o los comprime para evitar que rebasen el área prevista.



- Cada hijo de un contenedor flex se convierte en un elemento flexible. (no hereda la propiedad flex, pero se ajustará a lo que toque, es flexible)
- Propiedades column-\*, float y clear no tienen ningún efecto en un elemento flexible.
- vertical-align no tiene efecto en la alineación de los elementos flexibles.
- Ejes: Cada diseño de "caja flexible" sigue dos ejes:
  - El **eje principal** es el eje a lo largo del cual los elementos flexibles se suceden unos a otros.
  - El **eje secundario** es el eje perpendicular al **eje principal**.

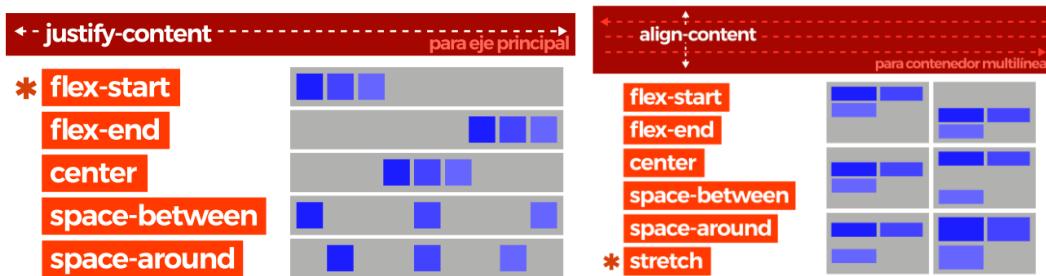


- La propiedad **flex-direction** establece el eje principal: `row | row-reverse | column | column-reverse`

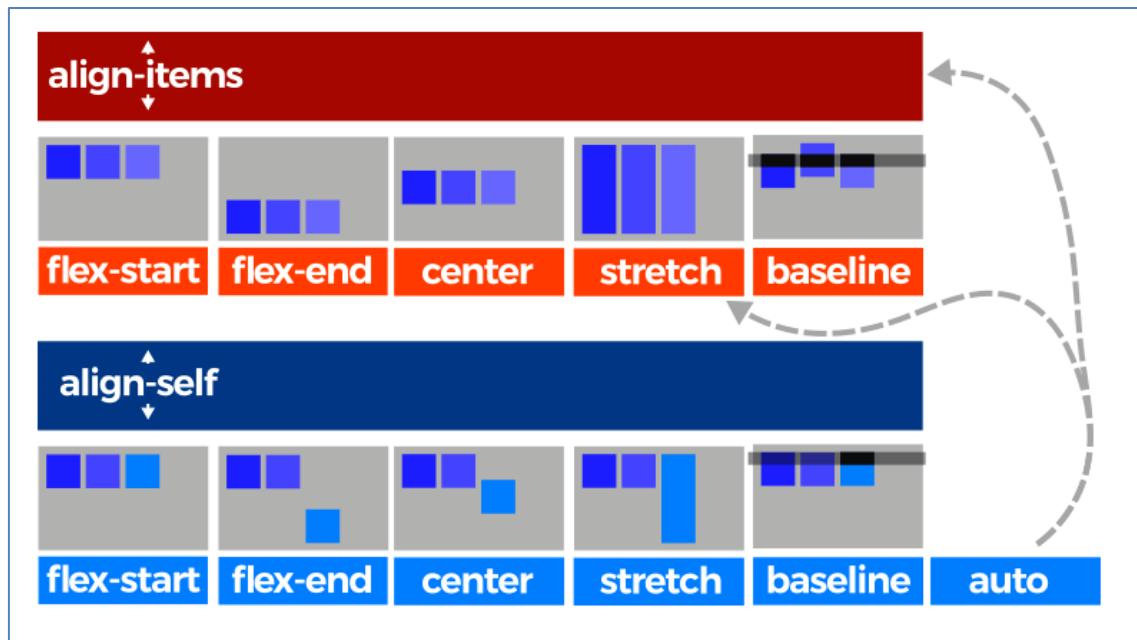
#### EJEMPLO: 7 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-FLEXDIRECTIONY ORDER

- Alinear en el eje ppal:
  - La propiedad **justify-content** define cómo los elementos flexibles se disponen a lo largo del eje principal en la línea en curso.  
`flex-start | flex-end | center | space-between | space-around`

#### EJEMPLO: 9 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-WRAP



- Alinear en el eje secundario:
  - La propiedad **align-items** define cómo los elementos flexibles se disponen a lo largo del eje secundario de la línea en curso: `flex-start | flex-end | center | baseline | stretch`
  - La propiedad **align-self** sustituye al valor por defecto establecido por align-items. Es para hijos de un contenedor flexible que se quieren alinear diferente a los demás hijos (definidos por align-items)



#### EJEMPLO: 8 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-ALIGN-ITEMSY ALIGN-SELF

- c) La propiedad `align-content` define cómo los elementos flexibles se disponen a lo largo del eje secundario en la línea en curso. Contenedor de varias líneas. Debe llevar `flex-wrap: wrap`.
   
`flex-start | flex-end | center | space-between | space-around | stretch`

#### EJEMPLO: 14 VISUALIZACIÓN ALIGN-CONTENT Y ALIGN-ITEMS

- La propiedad `order` asigna elementos a grupos ordinales y determina qué elementos aparecen primero, independientemente de si aparecen antes o después en el html.

#### EJEMPLO: 7 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-FLEXDIRECTIONY ORDER

- La propiedad `flex-wrap` especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas `wrap | no-wrap | wrap-reverse`

#### EJEMPLO: 9 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-WRAP

- La propiedad `flex-flow` combina las propiedades `flex-direction` y `flex-wrap` para colocar los elementos flexibles.
   
`flex-flow: <'flex-direction'> <'flex-wrap'>`
- La propiedad `flex` hace que los hijos ignoren su tamaño y se ajusten al espacio disponible. Acepta 3 valores:
  - 1º cuánto crece si sobra espacio: `flex-grow`
  - 2º cuánto encoge si falta espacio `flex-shrink`

- 3º tamaño inicial: flex-basis

Esos valores serán: (0, fija su tamaño y no varía,>1 se establece relación de aspecto entre ellos).

#### EJEMPLO: 6 VISUALIZACIÓN DISPLAY FLEX-ELEMENTOFLEX-FLEX-SHRINK-FLEX-GROW

- Las propiedades row-gap y column-gap permiten establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor, y sin necesidad de estar utilizando padding o margin en los elementos hijos.

#### EJEMPLO: 13 ROWGAP-COLUMN-GAP

## RESUMEN DISPLAY

PROPIEDAD	APLICA A....	CONDICIONES	VALORES POSIBLES
DISPLAY: Flex	Contenedor		Ninguno. Solo indica flexibilidad.
FLEX-DIRECTION	Contenedor		row, column
JUSTIFY-CONTENT	Contenedor		Eje main: flex-start, flex-end, center, space-between, space-around
ALIGN-CONTENT	Contenedor	flex-wrap: wrap Contenedor de varias líneas	Eje cross: flex-start, flex-end, center, space-between, space-around, stretch. (Es decir, cuando la propiedad flex-wrap es wrap y los elementos no caben en una sola línea). La clave está en si el conedor tiene o no altura
ALIGN-ITEMS	Contenedor		Eje cross: flex-end, flex-start, center, baseline, stretch
ALIGN-SELF	HIJO	Padre debe ser flex	Eje cross: flex-end, flex-start, center, baseline, stretch
ROW-GAP	Contenedor	Varias líneas	Establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor (no utilizar padding o margin en hijos)
COLUMN-GAP	Contenedor	Varias líneas	Establecer el tamaño de un «hueco» entre ítems desde el elemento padre contenedor (no utilizar padding o margin en hijos)
FLEX-WRAP	Contenedor		wrap, no-wrap, wrap-reverse
FLEX-FLOW	Contenedor		Abreviatura <flex-direction><flex-wrap>
FLEX	HIJO	Padre debe ser flex	0-mantiene   relación de aspecto (1.grow, 2.shrink, 3.basics )
ORDER	HIJO	Padre debe ser flex	Orden de aparición.

## OTRAS PROPIEDADES:

1. **VISIBILITY:** Contenedor, Hijo

Visualizar o no una caja, su hueco se queda, no es ocupado  
Hidden, visible, collapse (filas, columnas)

2. **OVERFLOW:** Contenedor, hijo

Qué hacer con el contenido sobrante de la caja.  
Visible, hidden, scroll  
Overflow-x y overflow-y

#### EJEMPLO: 10 VISUALIZACIÓN VISIBILITY Y OVERFLOW

3. **Z-INDEX:** Contenedor, hijo

Orden de aparición en cajas superpuestas.

Mayor número, más cerca de la visión.

EJEMPLO: 87\_ZINDEX

Fuentes:

- <https://uniwebsidad.com/libros/css-avanzado/capitulo-4/propiedad-display>
- [https://developer.mozilla.org/es/docs/Web/CSS/CSS Flexible Box Layout/Usando las cajas flexibles CSS](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Usando_las_cajas_flexibles_CSS)
- [https://developer.mozilla.org/es/docs/Web/CSS/CSS Flexible Box Layout/Conceptos Basicos de Flexbox](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceptos_Basicos_de_Flexbox)
- <https://3con14.biz/css/layout-responsive/flex-box/21-propiedades-flex-container.html>
- <http://w3.unpocodetodo.info/css3/flex-align-content.php>