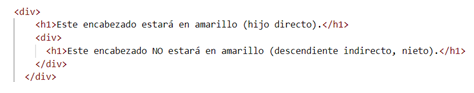
**SELECTORES**

* **Hijo ( > )**

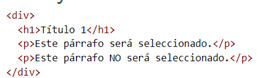


El elemento hijo directo

* **Adyacente ( + )**



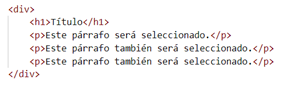
Hermanos y directamente adyacentes



* **Hermano ( ~ )**



Hermanos directamente adyancentes o no

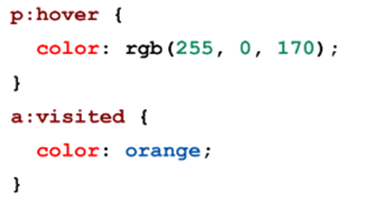


* **Selector con clase**

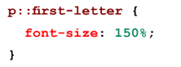


Selecciona los párrafos con clase “destacado”

* **Pseudoclases**

****

* + Hover: cuando se pasa el raton por encima
  + Visited: para enlaces ya visitados
  + Active: cuando el usuario activa el elemento
  + Focus: cuando el elemento recibe el foco
* **Pseudoelementos**

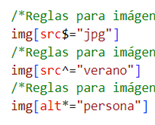
****

* + **first-line:** la primera línea de un texto
  + **before:** inserta contenido antes del elemento
  + **after:** inserta contenido despues del elemento
  + **first-letter**: la primera letra de un texto
  + **placeholder:** estilos al place holder de un input
  + **selection:** parte del contenido seleccionado por el usuario
* **Selectores de atributo**

****

Afecta a los elementos a que tengan un atributo target=”\_blank”

Mas Ejemplos:



**PROPIEDADES ABREVIADAS**

* **margin**

1. margin-top
2. margin-right
3. margin-bottom
4. margin-left

o

1. margin-top y margin-bottom
2. margin-right y margin-left



* **padding**

1. padding-top
2. padding-right
3. padding-bottom
4. padding-left



o

1. padding-top y padding-bottom
2. padding-right y padding-left

* **font**

1. font-style
2. font-variant
3. font-weight
4. font-size
5. line-height
6. font-family



* **background**

1. background-color
2. background-image
3. background-position
4. background-size
5. background-repeat
6. background-attachment
7. background-clip
8. background-origin



* **border**

1. border-width
2. border-style
3. border-color



* **border-radius**

1. top-left
2. top-right
3. bottom-right
4. bottom-left



* **outline:** borde que no afecta al tamaño total del elemento

1. outline-width
2. outline-style
3. outline-color



* **box-shadow**

1. desplazamiento-horizontal
2. desplazamiento-vertical
3. desenfoque (opcional)
4. color



* **flex**

1. flex-grow
2. flex-shrink
3. flex-basis



* **grid**

1. grid-template-rows
2. grid-template-columns
3. grid-template-areas
4. grid-auto-rows
5. grid-auto-columns
6. grid-auto-flow



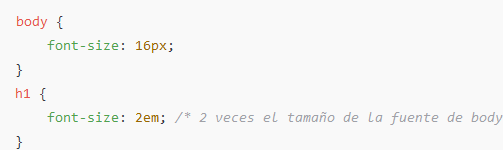
* **grid-area**

1. grid-row-start
2. grid-column-start
3. grid-row-end
4. grid-column-end

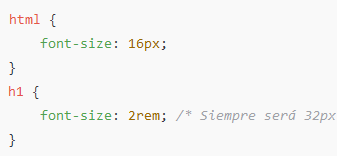


**UNIDADES RELATIVAS**

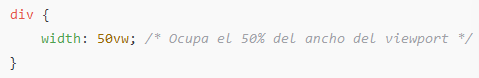
* **em:** relativo al tamaño de la fuente del elemento padre



* **rem:** relativo al tamaño de la fuente del elemento raíz (html)



* **vw:** relativo al 1% del ancho de la ventana visible del navegador



**MODELO DE CAJAS**

**BOX-SIZING:** Como se calculan las dimensiones de un elemento, si el padding y el borde se incluyen o no del tamaño total del elemento (width y height)

* **content-box -> width y height solo incluyen el contenido del elemento**
* **border-box -> width y height incluyen el contenido, el padding, y el border**

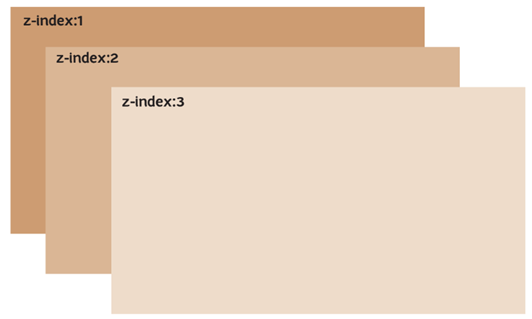
**DISPLAY**

* **none ->** elimina el elemento, queda invisible y se comporta como si no existiese en el flujo del diseño
* **block ->** se comporta como un bloque, Inicia en una nueva línea, ocupa el ancho completo del contenedor padre
* **inline ->** se comporta como contenido en línea, no inicia en una nueva línea, solo ocupa el ancho necesario para su contenido
* **inline-block ->** se comportan como bloques dentro de esa línea, no inicia en una nueva línea,
* **flex ->** se convierte en un contenedor flexible, tiene control de alineación y distribución con distintas propiedades
* **inline-flex ->** el contenedor flexible se comporta como un elemento en línea
* **grid ->** se comporta como un contenedor de cuadrícula, tiene control de distribución con distintas propiedades
* **inline-grid ->** el contenedor de cuadrícula se comporta como un elemento en línea

**POSICIONAMIENTO**

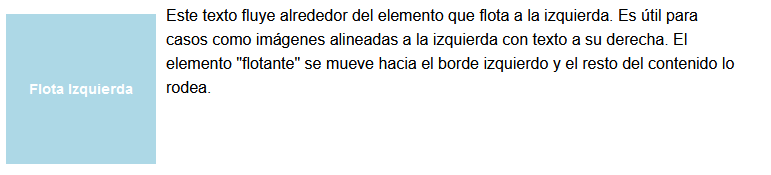
**POSITION**

* **static:** se colocan de acuerdo al orden normal del documento (código)
* **relative:** se colocan de acuerdo al orden normal, pero se pueden ,over respecto a su posición original (top, right, bottom…)
* **absolute:** se posiciona en relación con el contenedor mas cercano que tenga una posición distinta de static, en su defecto se posiciona en relación con <html>
* **fixed:** se posiciona en relación con la ventana del navegador, permanecerá en la misma posición incluso cuando el usuario haga scroll
* **sticky:** se comporta como relative hasta que llega a un cierto punto en la pagina
* **z-index:** controla que elementos se muestran por encima de otros

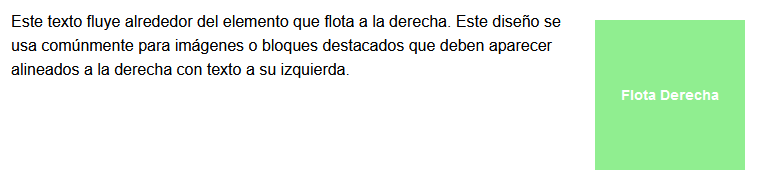
****

**FLOAT**

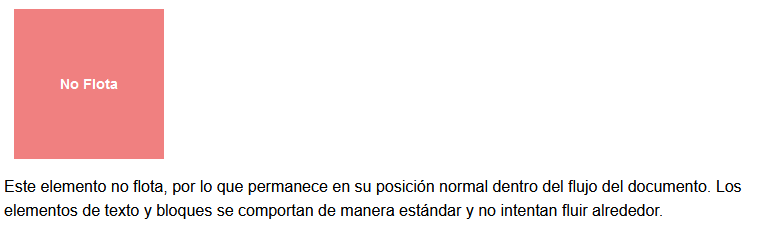
* **left:** flota hacia la izquierda, los demás elementos rodean al elemento por el lado derecho

****

* **right:** flota hacia la derecha, los demás elementos rodean al elemento por el lado izquierdo



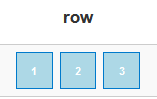
* **none:** permanece en su posición normal



**FLEXBOX**

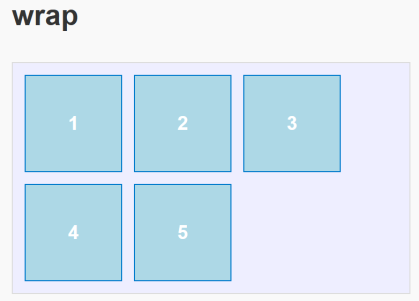
**Propiedades que se aplican al contenedor con display: flex**

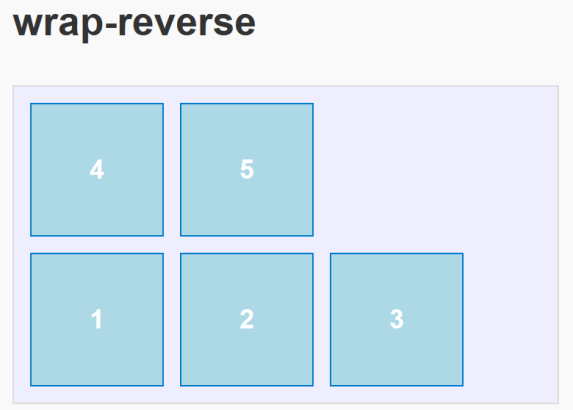
* + **flex-direction**: define la dirección de los contenedores
    - column: columna vertical
    - column-reverse: columna vertical con el orden invertido
    - row: fila horizontal
    - row-reverse: fila horizontal con el orden invertido

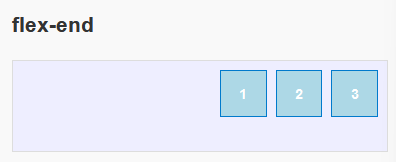
 

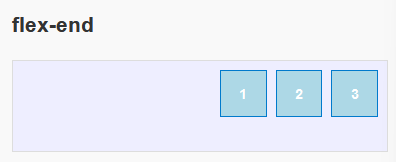
* + **flex-wrap**: Determina si al acabar una línea los elementos deben pasarse a una nueva
    - wrap: se ajustan a una nueva línea
    - non-wrap: se colocan en una sola línea
    - wrap-reverse: se ajustan una nueva línea en orden inverso



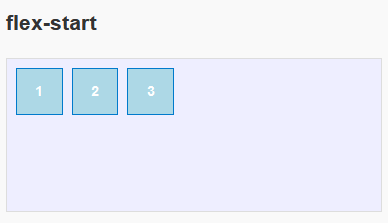
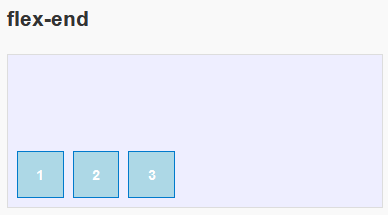
* + **justify-content:** la alineación en el eje horizontal
    - flex-start: alineados al inicio del eje principal
    - flex-end: alineados al final del eje principal.
    - center: alineados al centro
    - space-between: espacio igual entre los elementos pero el primero y el ultimo se ajustan a los extremos
    - space-around: espacio igual entre los elementos

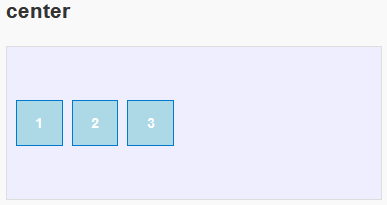
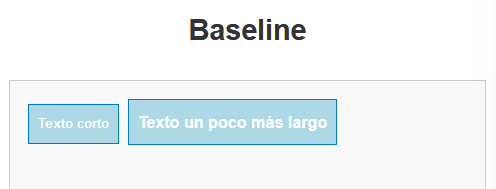
 

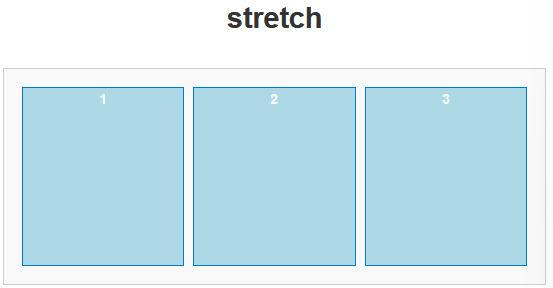
 



* + **align-items:** la alineación en el eje vertical
    - flex-start: alineados al inicio del eje principal
    - flex-end: alineados al final del eje principal.
    - center: alineados al centro
    - baseline: alineados según sus líneas base de texto
    - stretch: elemento estirados verticalmente para llenar el contenedor
  + **gap:** espacio en blanco entre los elementos flexibles y las filas

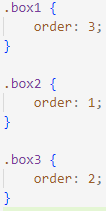
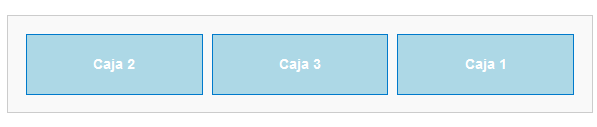
 

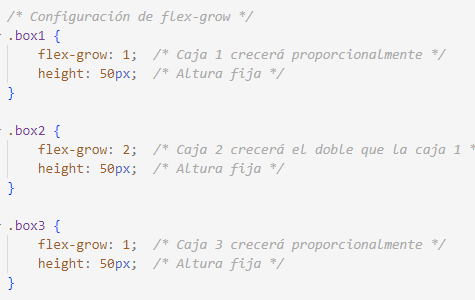


**Propiedades que se aplican a los hijos de un contenedor con display: flex**

* + order: define el orden en que se visualizan los elementos

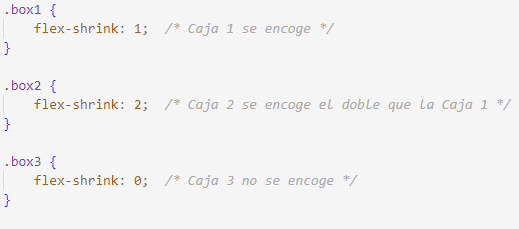
 

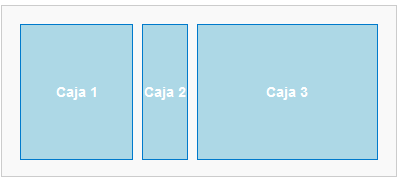
* + flex-grow: define cómo los elementos dentro de un contenedor **deben crecer** para ocupar el espacio disponible en el contenedor





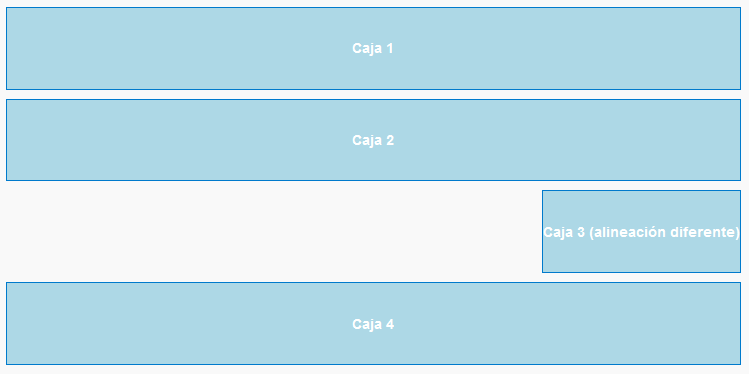
* + flex-shrink: cuanto puede encogerse un elemento cuando no hay suficiente espacio





* + flex-basis: define el tamaño base del elemento antes de aplicar lex-grow y flex-shrink
  + aling-self: permite alinear un elemento individual de manera diferente al resto

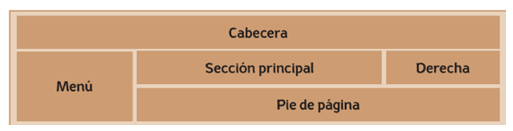




\* Abreviatura flex

flex; (flex-grow) (flex-shrink) (flex-basis);

**GRID**

****

**Propiedades del contenedor con display: grid;**

* **grid-template-columns:** define el numero y el tamaño de las columnas
* **grid-template-rows:** define el numero y el tamaño de las filas
  + por pixeles:



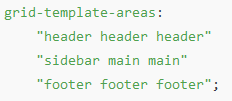
3 columnas de 100, 200 y 100 pixeles

* + por fracciones:



3 columnas, la del medio es el doble que las demás

* **grid-template-areas:** asignar areas con nombres



**Propiedades de los hijos del contenedor con display: grid**

* **grid-column:** define desde que columna a que columna va ese ítem



Desde la columna 2 a la 4

* **grid-row:** define de que fila a que fila va ese ítem

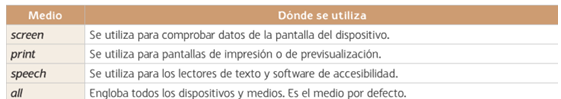


Desde la fila 1 a la 2

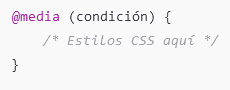
* **grid-area (abreviatura)**

grid-area: (grid-row-start) (grid-column-start) (grid-row-end) (grid-column-end);

**MEDIA QUERY**

****

**Sintaxis**

****

**Condiciones**

* ancho de la ventana
  + max-width
  + min-width
  + max-height
  + min-height
* multiples condiciones

