UT4_2.- HOJAS DE ESTILO. AVANZADO - FLEX

RESULTADOS DE APRENDIZAJE ASOCIADOS
1. Reconoce las características de lenguajes de marcas analizando e interpretando fragmentos de código. 2. Utiliza lenguajes de marcas para la transmisión de información a través de la Web analizando la estructura de los documentos e identificando sus elementos
CRITERIOS DE EVALUACIÓN
- De RA1 – desde CEA hasta CEK - De RA2 – desde CEA hasta CEH

UT4_2.- HOJAS DE ESTILO. AVANZADO - FLEX

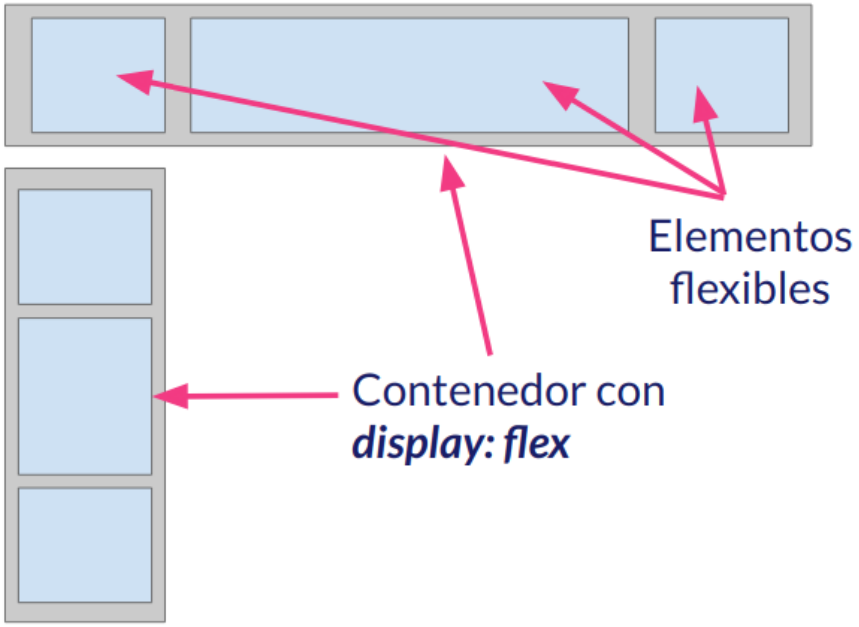
Índice de contenido

1.- Elementos de la maquetación con flex

La principal idea que hay detrás de la maquetación **FLEX** es que vamos a tener un elemento que va a poder *controlar* las propiedades de los elementos que contiene.

Por lo que vamos a tener dos tipos de elementos:

- ✓ El **contenedor flex** que tendrá asignada la propiedad CSS *display:flex* y que va a controlar (ya veremos cómo) ciertas propiedades de los elementos que contiene. Los vemos en gris en la imagen inferior.
- ✓ Los **elementos flexibles** que son los elementos que están dentro del contenedor y cuyas propiedades modificaremos. Los vemos en azul en la imagen inferior.



Con FLEX vamos a poder modificar propiedades CSS como:

- ✓ La altura de los elementos flexibles.
- ✓ La anchura.
- ✓ El orden en el que se nos van a presentar.
- ✓ La alineación vertical.
- ✓ La alineación horizontal.
- ✓ La distribución de los elementos flexibles a lo largo del contenedor.

Es decir, vamos a poder controlar propiedades que usamos para maquetar de manera mucho más ágil a lo que lo hacemos con las técnicas tradicionales de maquetado.

2.- Propiedades FLEX

El primer paso para maquetar usando elementos FLEX es añadir el valor **flex** a la propiedad **display** del contenedor.

```
<div class="contenedor">
    <div><p>Primera frase</p></div>
    <div><p>Segunda frase</p></div>
    <div><p>Tercera frase</p></div>
</div>
```

```
.contenedor {
    display:flex;
}

p {
    border: 1px solid red;
}
```

Con la propiedad ***display:flex*** podremos ver que de manera automática los elementos ajustan su anchura a su contenido y se distribuyen en horizontal.

Primera frase Segunda frase Tercera frase

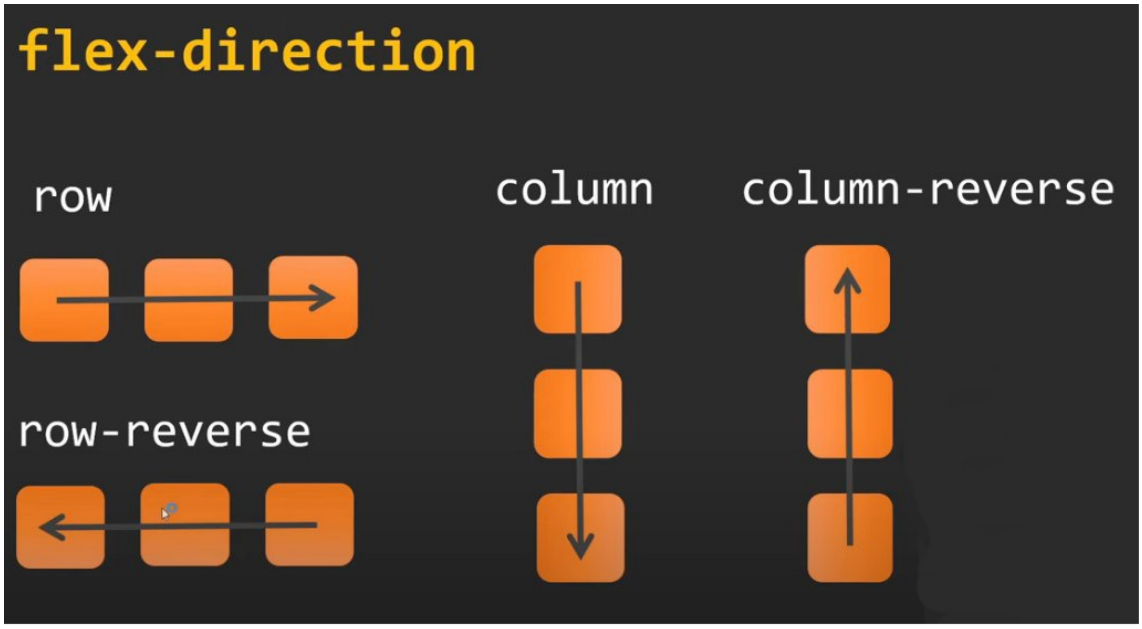
Además de con esta regla, desde el contenedor flex podemos modificar de manera directa varias propiedades de los elementos hijos. Por ejemplo:

- ✓ La dirección en la que se van a mostrar.
- ✓ Cómo se van a ajustar dentro del contenedor.
- ✓ La alineación horizontal de los elementos flexibles.
- ✓ La alineación vertical de los elementos flexibles cuando solo ocupan una línea.
- ✓ La alineación vertical de los elementos flexibles cuando ocupan más de una línea.

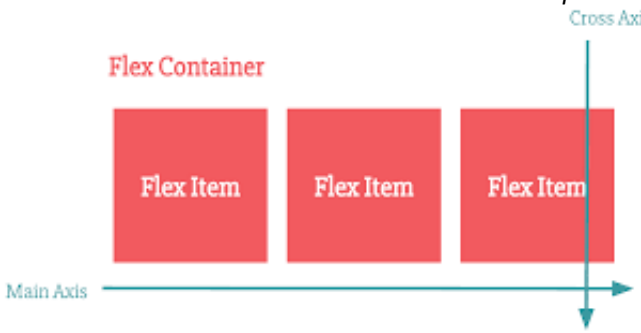
2.1.- Dirección de los elementos flexibles

La ajustaremos mediante la propiedad **flex-direction** que podrá tomar los siguientes valores:

- ✓ **row:** es la opción por defecto y ajustará los elementos flexibles de izquierda a derecha.
- ✓ **row-reverse:** igual que la anterior, pero de derecha a izquierda.
- ✓ **column:** ajustará los elementos flexibles en columna, de arriba a abajo.
- ✓ **column-reverse:** igual que la de arriba, pero de abajo a arriba.



Es importante tener en cuenta que en flex tenemos dos ejes main axis y cross axis, que por defecto son el eje horizontal y vertical.



Al darle valor column (o column-reverse) a la propiedad **flex-direction** se modifican estos ejes y el horizontal pasa a ser el vertical y el vertical pasa a ser el horizontal. Este comportamiento nos va a afectar cuando trabajamos con la alineación vertical y horizontal que veremos más adelante.

```
.contenedor {
    display:flex;
    flex-direction: column;
}
```

Primera frase

Segunda frase

Tercera frase

2.2.- El ajuste de los elementos flexibles

Hemos visto como antes, por defecto y sin indicar ninguna anchura, los elementos flexibles adecuaban su tamaño a su contenido (si no les dábamos anchura) y se ponían todos a la izquierda permaneciendo siempre así, aunque el ancho de la pantalla sea muy pequeño. Esto puede provocar desajustes en pantallas muy pequeñas. Este contratiempo podemos controlarlo usando la propiedad **flex-wrap** y eligiendo uno de los siguientes valores:

- ✓ **no-wrap:** es el valor por defecto y fuerza para que siempre los elementos estén en la misma línea, aunque esto suponga que se

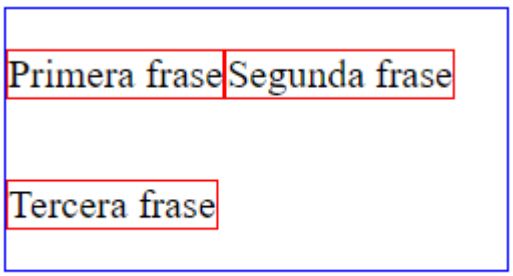
salgan del contenedor (independientemente del ancho que se les haya dado).

- ✓ **wrap:** provoca un salto de línea si la anchura de los elementos (fijada por nosotros o por el contenedor) es superior a la del contenedor.
- ✓ **wrap-reverse:** lo mismo que arriba, pero de abajo a arriba.

Podemos ver el efecto de los últimos valores en la siguiente imagen:

Las dos propiedades, **flex-direction** y **flew-wrap** podemos juntarlas en la propiedad **flex-flow** con dos partes como, por ejemplo:

```
.contenedor {
    border: 1px solid blue;
    display:flex;
    width: 200px;
    flex-flow: row wrap;
}
```



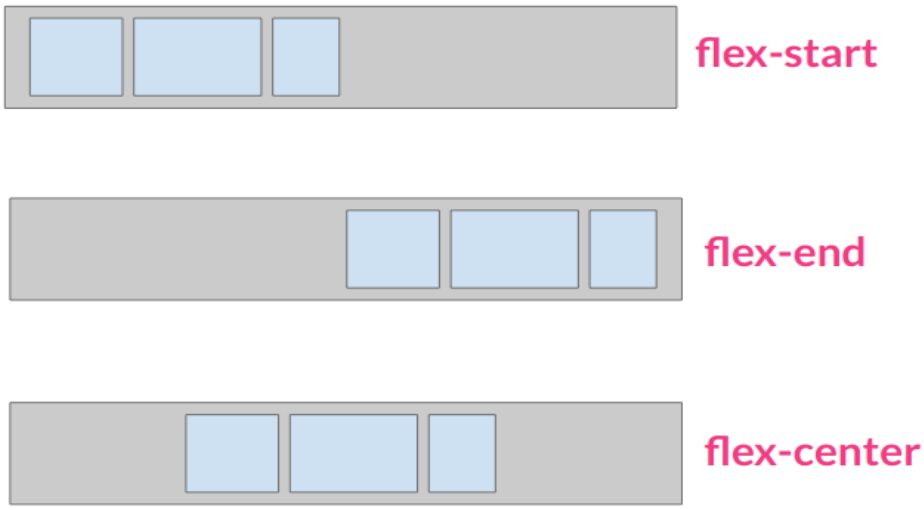
2.3.- Alineación horizontal de los elementos flexibles

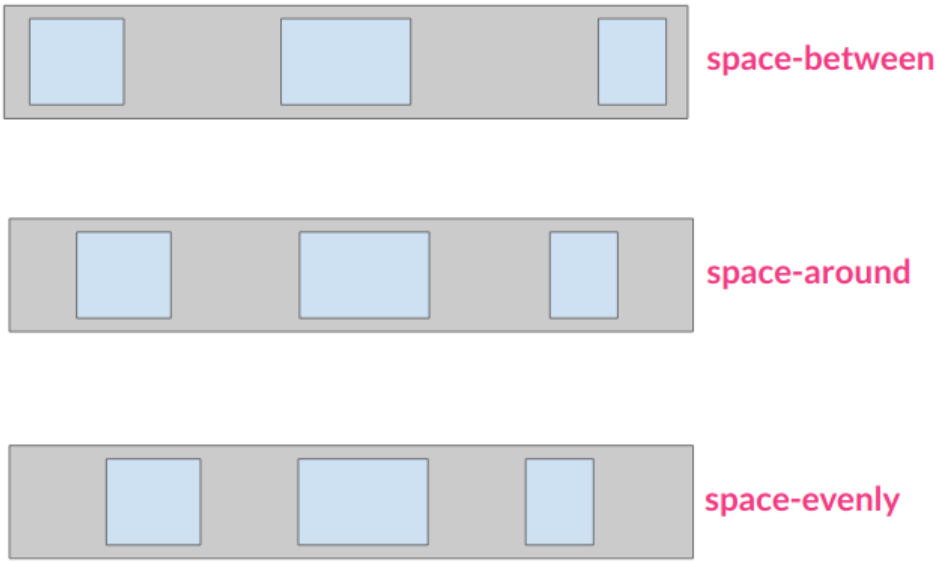
Podemos alinear horizontalmente los elementos flexibles, tengan o no tengan establecida una anchura, añadiendo la propiedad **justify-content** al elemento contenedor.

Esta propiedad puede tener 6 valores distintos:

- ✓ **flex-start:** Los elementos flexibles se sitúan al principio.
- ✓ **flex-end:** Los elementos flexibles se sitúan al final.
- ✓ **center:** Los elementos se centran horizontalmente
- ✓ **space-beetween:** Distribuye el espacio restante entre los elementos, pero el primero y el último están en los bordes.
- ✓ **space-around:** Distribuye el espacio restante entre los elementos, pero no tiene en cuenta la distancia a los bordes.
- ✓ **space-evenly:** Distribuye el espacio restante entre los elementos y tiene en cuenta la distancia a los bordes.

Visualmente:

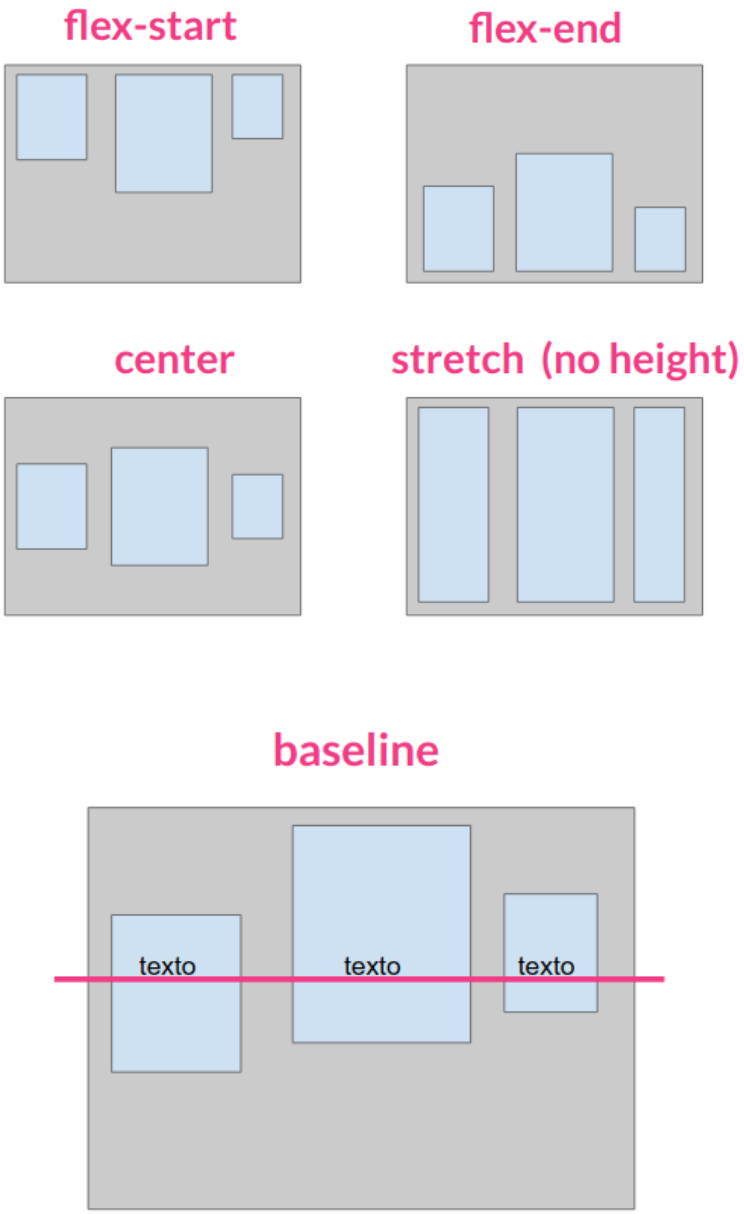




2.4.- Alineación vertical de los elementos flexibles

Podemos alinear verticalmente los elementos flexibles añadiendo la propiedad **align-items** que puede tomar los siguientes valores:

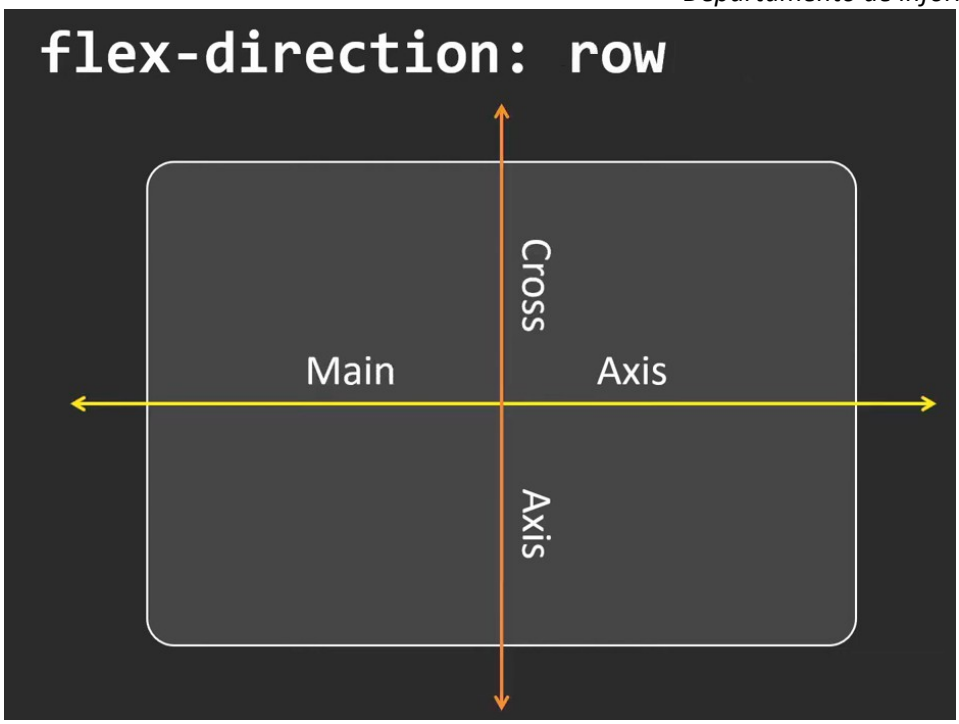
- ✓ **flex-start:** Los elementos se ponen junto al borde superior.
- ✓ **flex-end:** Los elementos se ponen junto al borde inferior.
- ✓ **center:** Los elementos flexibles se centran verticalmente.
- ✓ **stretch:** Los elementos crecen en altura para ocupar toda la altura del contenedor flexible. No deben tener altura fija establecida.
- ✓ **baseline:** Los elementos se alinean en relación con la primera línea de texto que posean los elementos flexibles.

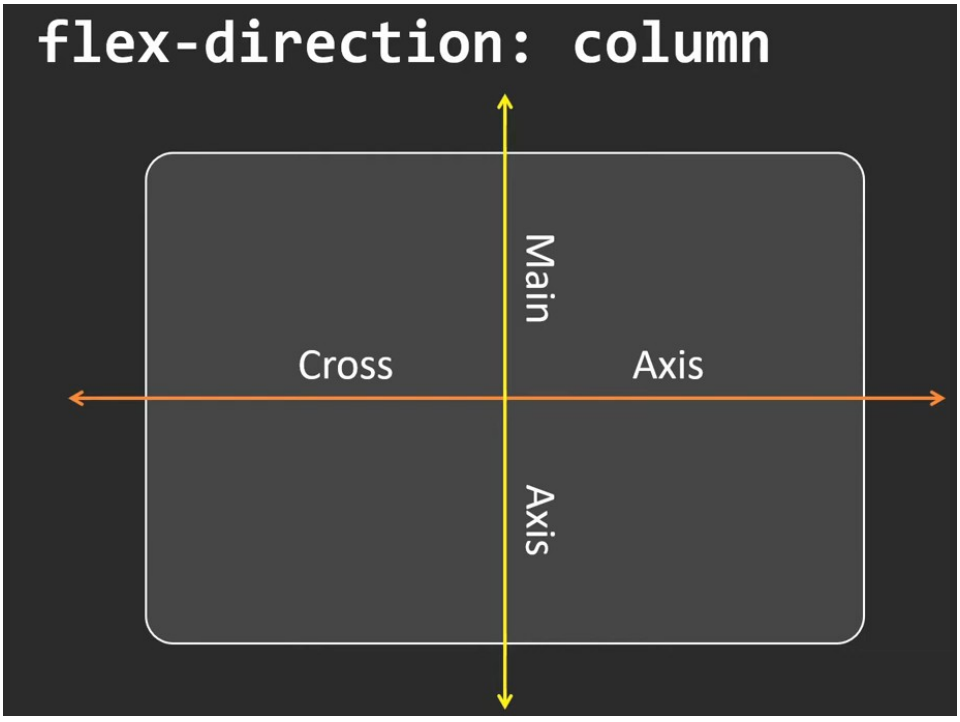


2.5.- Cross Axis y Main Axis

Como ya vimos en el primer apartado, estos ejes intercambian sus valores cuando damos valor column a la propiedad flex-direction.

Cuando damos valor column a la propiedad flex-direction, **lo que sucede es que la propiedad justify-content pasa a utilizarse para el eje vertical y la propiedad aling-items pasa a utilizarse sobre el eje horizontal.** Como decíamos antes esto se debe al cambio de ejes que se produce.



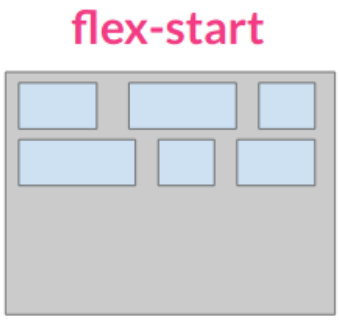


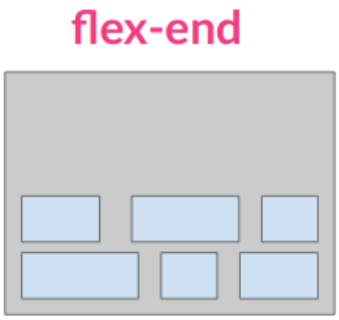
2.6.- Alineación vertical – Wrap (Cuando hay varias líneas)

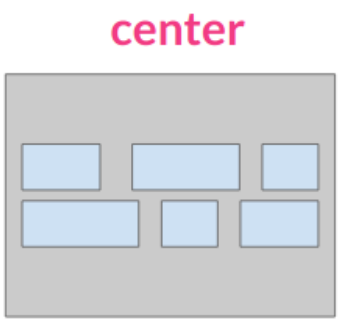
Si al usar la propiedad *flex-wrap:wrap* resulta que hay varias líneas de elementos flexibles, también podemos alinearlas usando la propiedad **align-content** que toma los valores:

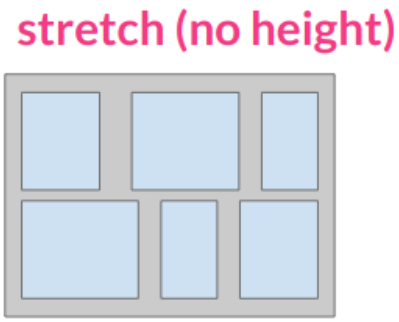
- ✓ **flex-start**

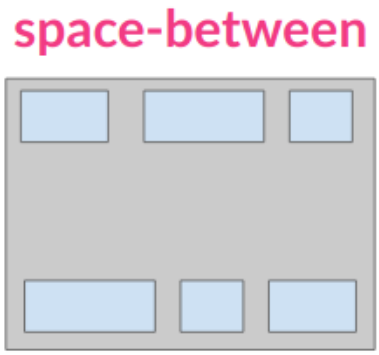
- ✓ **flex-end**
- ✓ **center**
- ✓ **stretch**
- ✓ **space-between**
- ✓ **space-around**

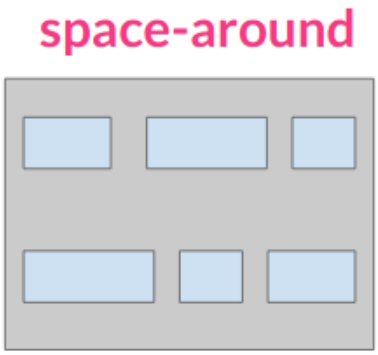












2.7.- Order

Nos permite modificar el flujo de nuestro documento .html. Podemos modificar el orden en el que se muestran elementos de pantalla, independientemente de su colocación en nuestro documento .html

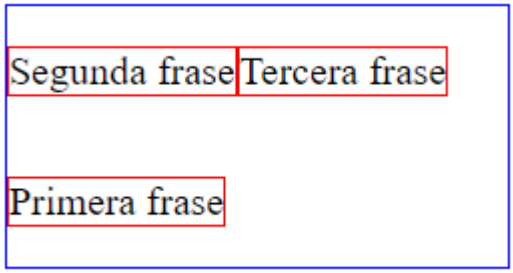
```
.posicion{
    order: 1;
 }
```

Cuando damos valor a la propiedad ***order*** y modificamos su posición, lo hacemos en función de su valor, los elementos que tienen un valor para la propiedad ***order*** mayor, se colocan a la derecha de los elementos que tiene un valor para orden menor

En este caso podemos ver que la frase1 que en nuestro **.html** la hemos escrito en la primera posición, aparece en la última posición, esto es debido a que el valor de ***order*** para este elemento es mayor que para el resto de los elementos.

```
<div class="contenedor">
        <div class="frase1"><p>Primera frase</p></div>
        <div><p>Segunda frase</p></div>
        <div><p>Tercera frase</p></div>
</div>
```

```
.frase1{
    order:1;
}
```



Por defecto todos los elementos tienen order 0.

Si quieres repasar:
https://www.youtube.com/watch?v=BIXgEpITld4

Si quieres ampliar:
https://www.mclibre.org/consultar/htmlcss/css/css-flexbox.html#display

https://www.joshwcomeau.com/css/interactive-guide-to-flexbox/