

PROGRAMACION PARA INTERNET

FLEXBOX

FLEXBOX

PROFESOR

Torres Villanueva Nancy Michelle

Rodríguez Loza Sergio Alain
INCO (214288953)

SECCIÓN: D06
CICLO: 2017B

05/09/2017

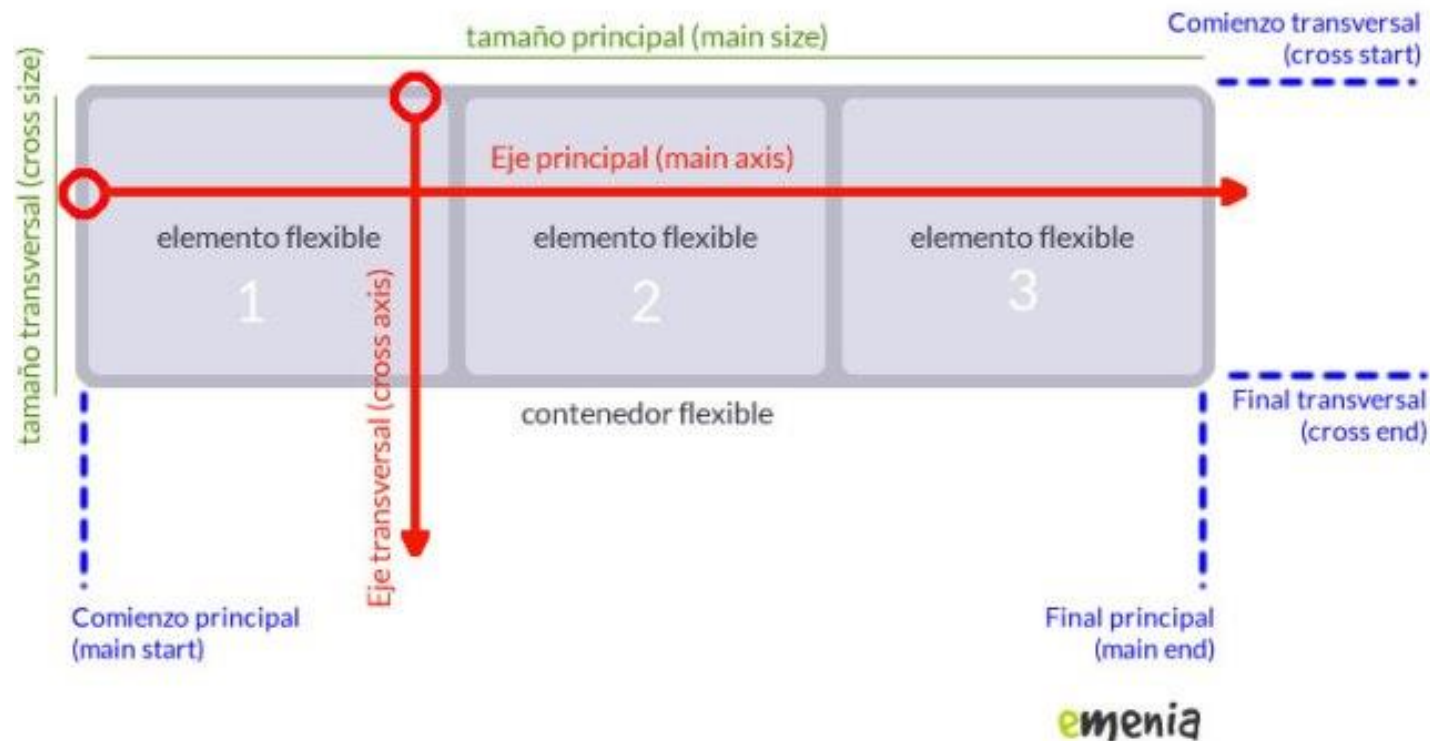


En qué consiste Flexbox.

- ▶ Facilita la creación de páginas flexibles, fluidas y dinámicas que funcionen en la amplia variedad de dispositivos existentes utilizando CSS.
- ▶ Se crea un modelo de caja o contenedor optimizado.
- ▶ Ajusta los tamaños y la disposición de los elementos que se encuentran dentro de un contenedor o caja.
- ▶ Adaptación siempre al espacio disponible.

display: flex

- La declaración `display: flex;` define un “contenedor flexible” y convierte de forma automática a sus “hijos” directos en “elementos flexibles”.



Propiedades del contenedor - flex (Padre)

flex-direction

- Esta propiedad define las direcciones del eje principal, es decir, hacia donde se moverán los Elementos - Flex, tanto horizontalmente como verticalmente.

flex-direction

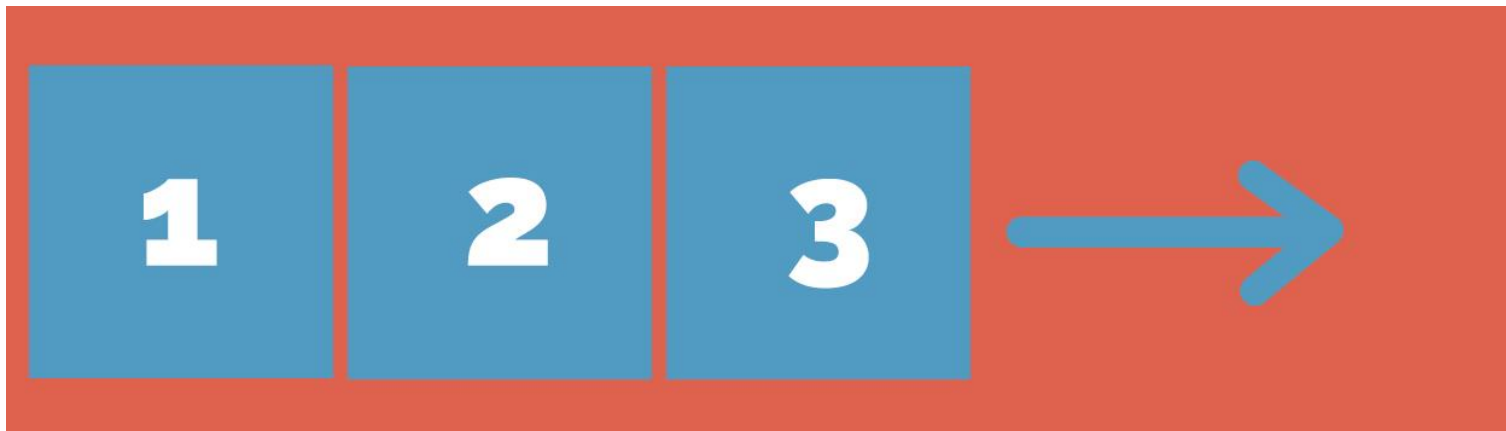
“

Valor por defecto: row

Horizontalmente:

```
.padre {  
    flex-direction: row;  
}
```

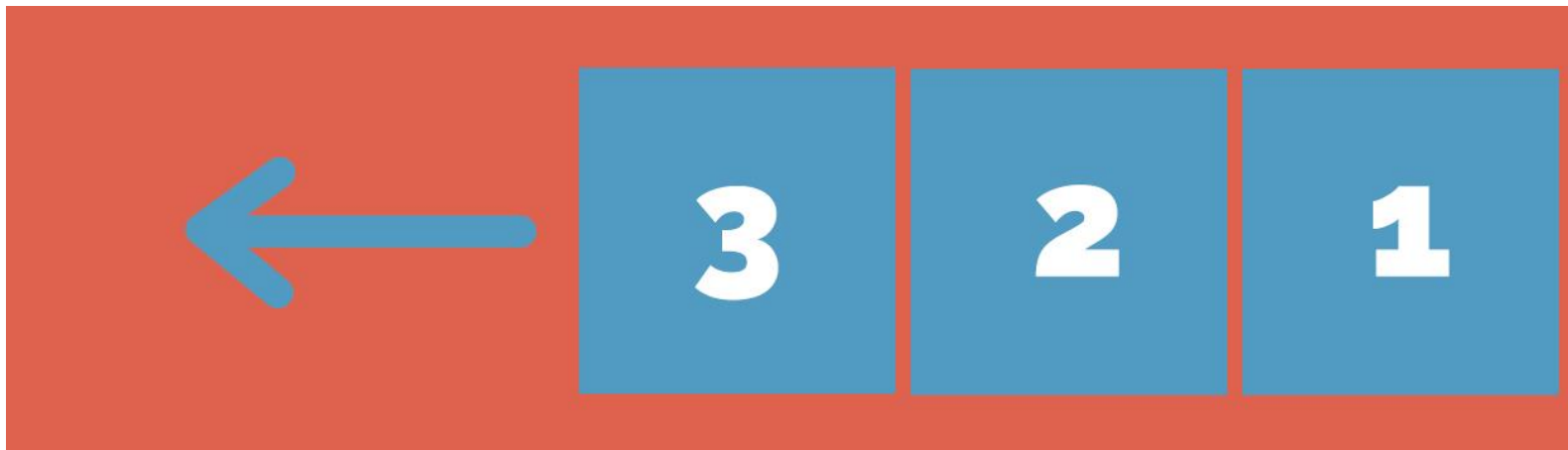
- Los elementos - flex se apilan en una fila de izquierda a derecha.



Horizontalmente:

```
.padre {  
    flex-direction: row-reverse;  
}
```

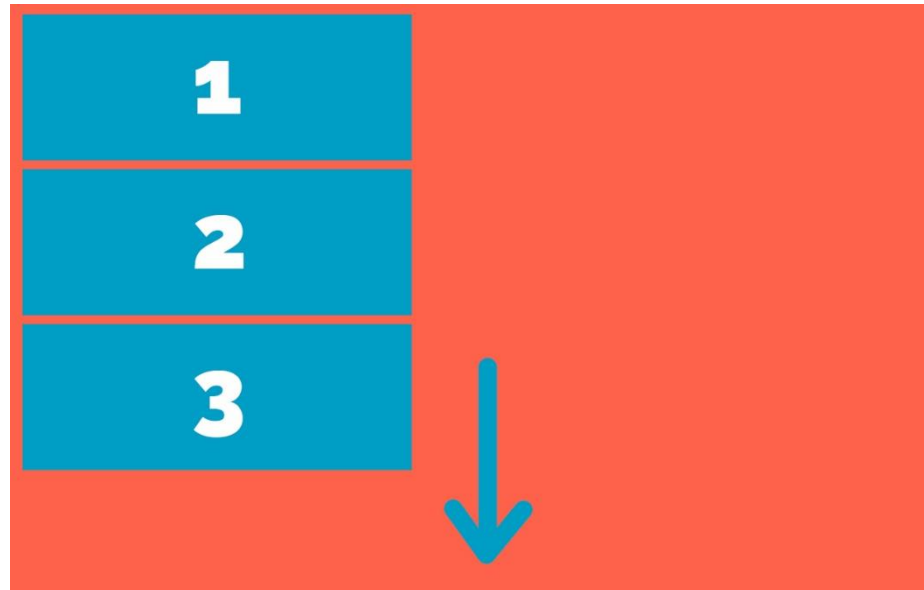
- Los elementos - flex se apilan en una fila de derecha a izquierda.



Verticalmente:

```
.padre {  
    flex-direction: column;  
}
```

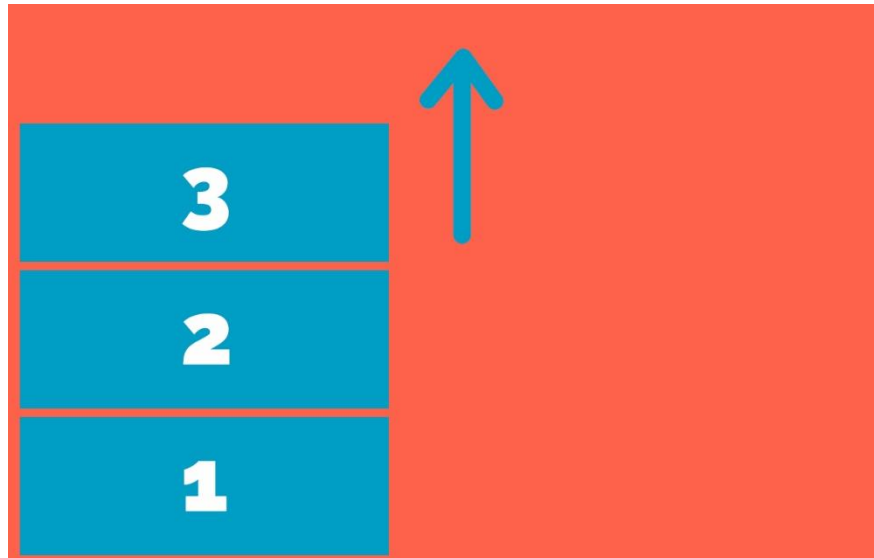
- Los elementos - flex se apilan en una columna de arriba abajo.



Verticalmente:

```
.padre {  
    flex-direction: column-reverse;  
}
```

- Los elementos - flex se apilan en una columna de abajo arriba.



flex-wrap

- Al manejar Flexbox el concepto inicial es fijar los elementos - flex en una sola línea, pero con la propiedad flex-wrap controlamos si el contenedor flex mueve sus elementos - flex en líneas individuales o múltiples.

flex-wrap

“

Valor por defecto: nowrap

flex-wrap

```
.padre {  
    flex-wrap: nowrap;  
}
```

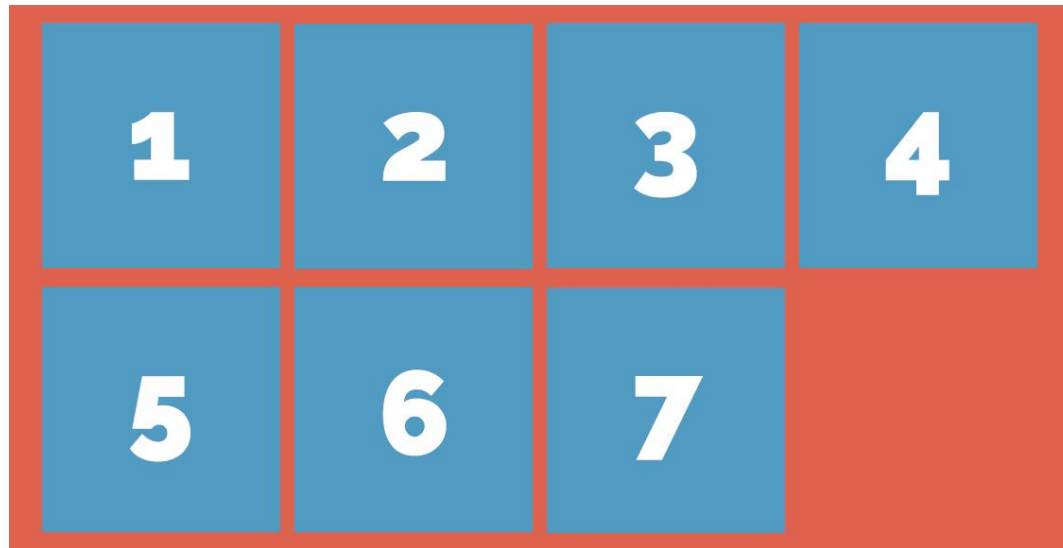
- Los elementos - flex se muestran en una fila y se pueden encoger dependiendo de su contenedor - flex.



flex-wrap

```
.padre {  
    flex-wrap: wrap;  
}
```

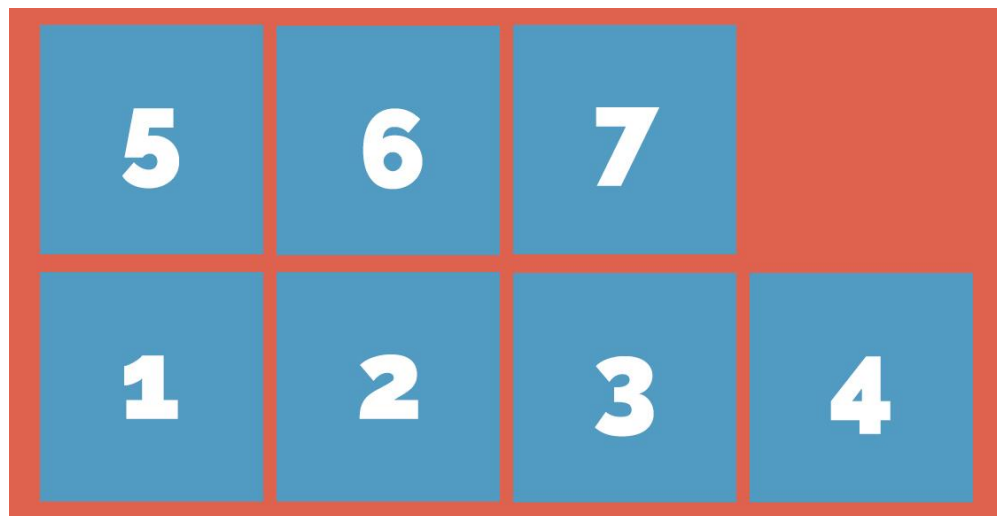
- Los elementos - flex se muestran en varias filas (si es necesario), de izquierda a derecha y de arriba abajo.



flex-wrap

```
.padre {  
    flex-wrap: wrap-reverse;  
}
```

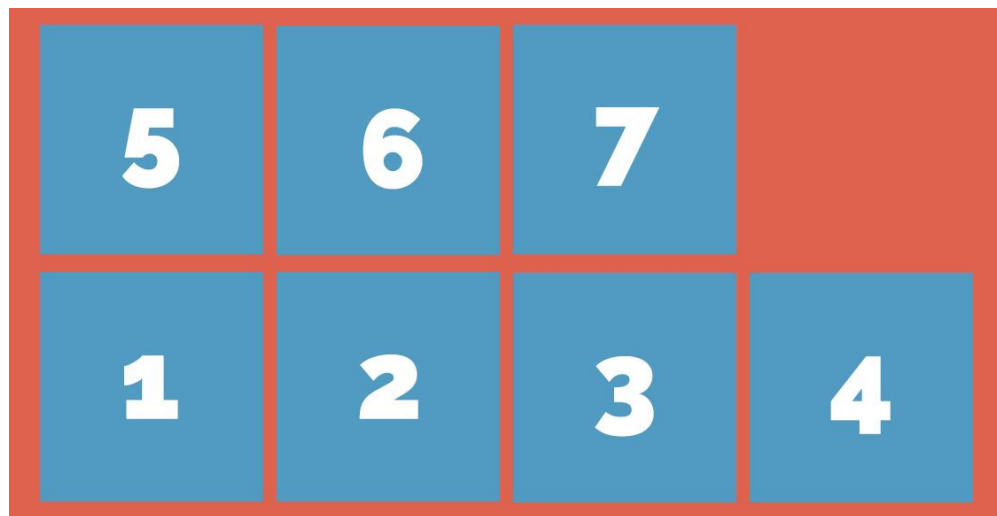
- Los elementos - flex se muestran en varias filas (si es necesario), de izquierda a derecha y de abajo arriba.



flex-wrap

```
.padre {  
    flex-wrap: wrap-reverse;  
}
```

- Los elementos - flex se muestran en varias filas (si es necesario), de izquierda a derecha y de abajo arriba.



flex-flow

- Esta propiedad es una forma rápida para establecer las propiedades flex-direction y flex-wrap.

flex-flow

“

Valor por defecto: row nowrap

```
.padre {  
    flex-flow: column-reverse wrap;  
}
```

justify-content

- La propiedad justify-content alinea los elementos - flex a lo largo del eje principal de la línea actual (contenedor - flex).

justify-content

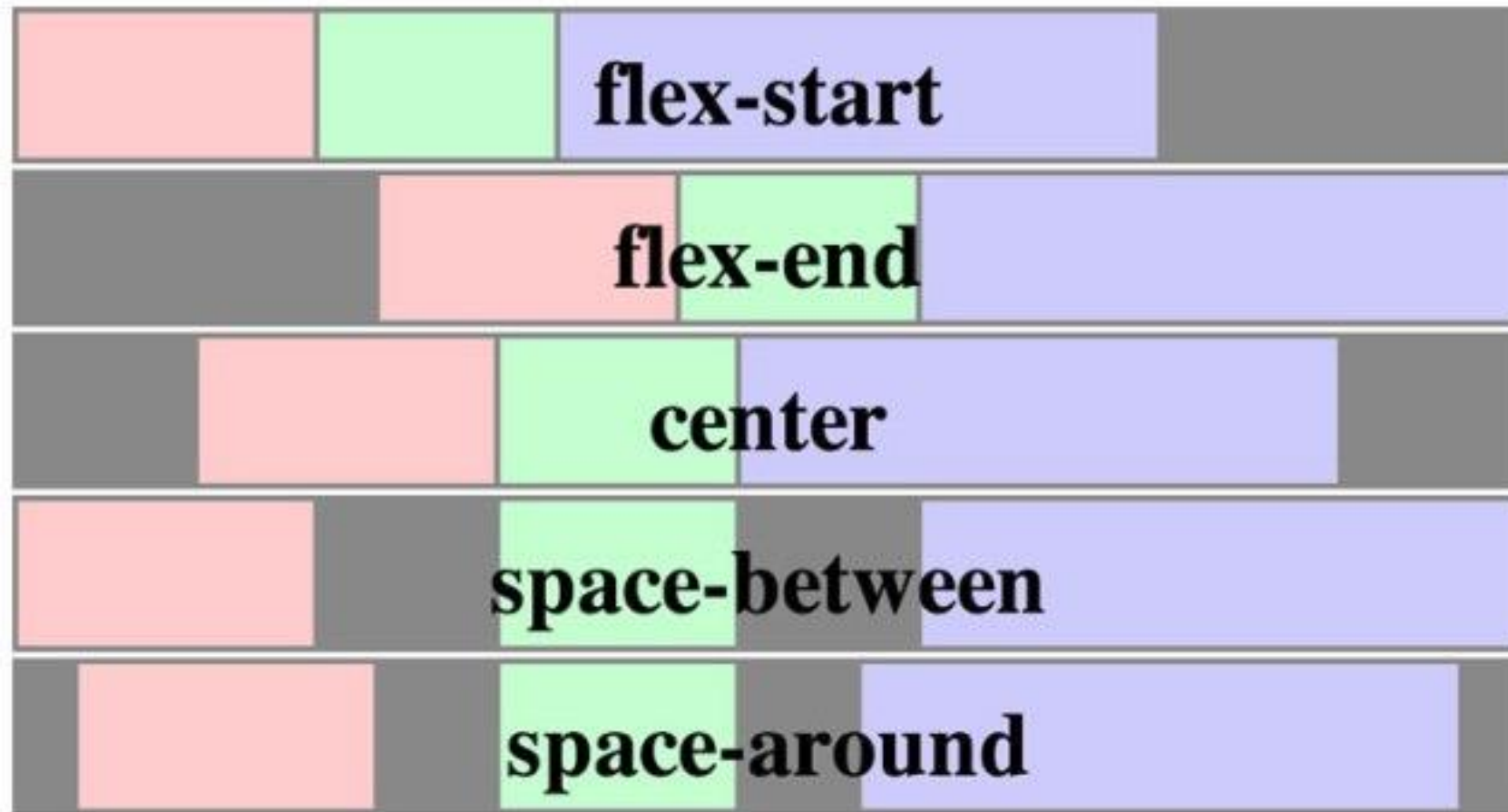
“

Valor por defecto: flex-start

Alineación de los elementos flexibles.

- ▶ Podemos alinear los elementos flexibles en el Eje principal con `justify-content` y en el Eje transversal con `align-items` y `align-self`.
- ▶ Si hay espacio extra dentro de un contenedor flexible la propiedad `justify-content` puede definir cómo se usa ese espacio que sobra. Las opciones son:
 - ▶ `flex-start`: se distribuyen todos pegados al inicio.
 - ▶ `flex-end`: se distribuyen todos pegados al final.
 - ▶ `center`: se distribuyen todos alineados al centro.
 - ▶ `space-between`: se distribuyen ocupando todo el espacio disponible, con separaciones iguales entre ellos, pero sin dejar espacio al inicio y al final.
 - ▶ `space-around`: se distribuyen ocupando todo el espacio disponible, con separaciones iguales entre ellos, dejando espacio al inicio y al final.

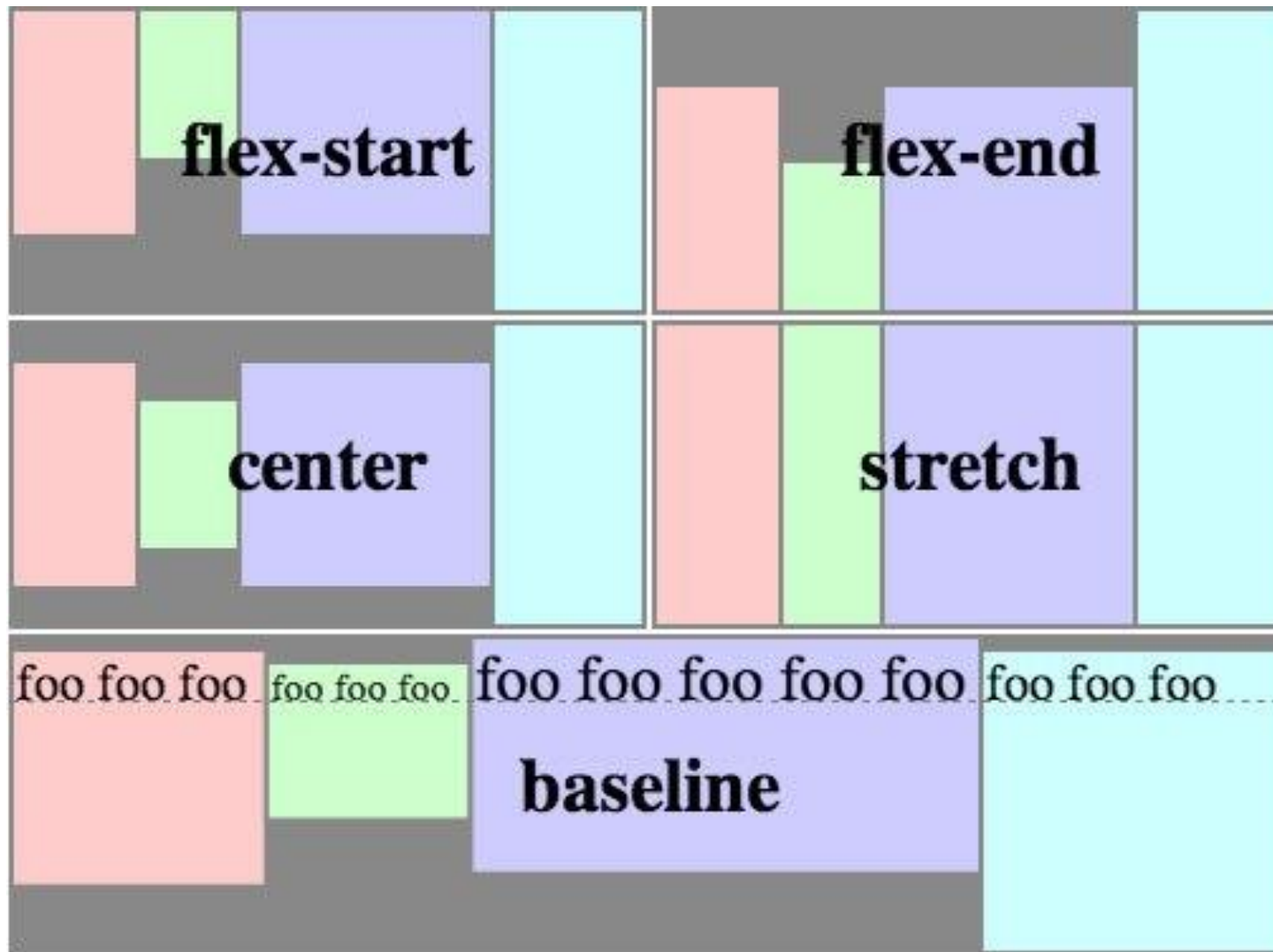
Alineación de los elementos flexibles.



Alineación de los elementos flexibles.

- ▶ Podemos alinear los elementos flexibles en el eje transversal con `align-items` y con `align-self`.
- ▶ `Align-items` establece el valor por defecto para todos.
- ▶ `Align-self` se aplica a elementos individuales sobre-escribiendo `align-items` para ese elemento.
 - ▶ `auto`: Sólo se puede aplicar en `align-self` y equivale al valor de `align-items` del elemento padre, o a `stretch` si el elemento no tiene padre.
 - ▶ `flex-start`
 - ▶ `flex-end`
 - ▶ `center`
 - ▶ `baseline`
 - ▶ `stretch`

Alineación de los elementos flexibles.



align-content

- La propiedad align-content alinea las líneas de un contenedor - flex cuando los elementos no utilizan todo el espacio disponible en el eje secundario.

align-content

“

Valor por defecto: `stretch`

align-content

```
.padre {  
    align-content: stretch;  
}
```

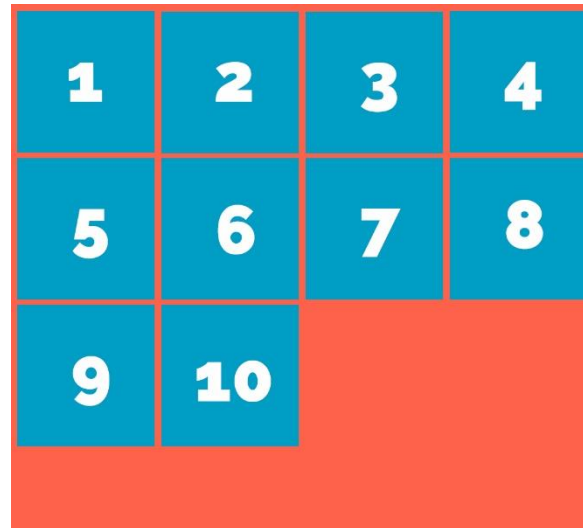
- Los elementos - flex se muestran con espacio distribuido después de cada fila de elementos - flex.

1	2	3	4
5	6	7	8
9	10		

align-content

```
.padre {  
    align-content: flex-start;  
}
```

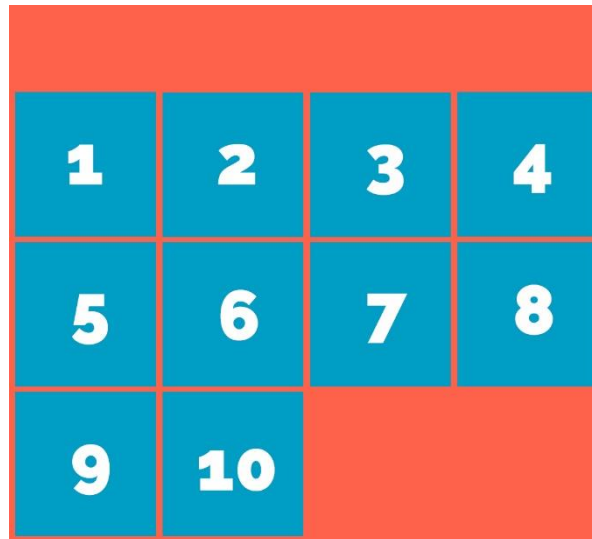
- Los elementos - flex se apilan hacia el inicio secundario del contenedor - flex.



align-content

```
.padre {  
    align-content: flex-end;  
}
```

- Los elementos - flex se apilan hacia el fin secundario del contenedor - flex.



align-content

```
.padre {  
    align-content: center;  
}
```

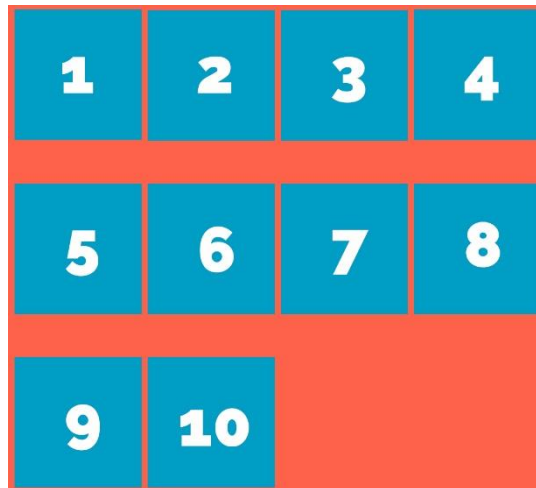
- Las filas se apilan en el centro del eje secundario.

1	2	3	4
5	6	7	8
9	10		

align-content

```
.padre {  
    align-content: space-between;  
}
```

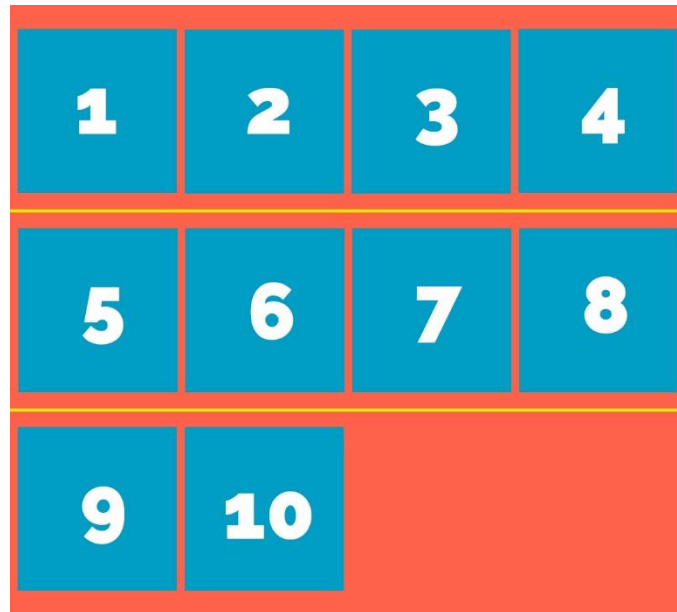
- Las filas de los elementos - flex tienen la misma separación entre ellos, pero la primera y última fila están alineadas con los bordes del contenedor - flex.



align-content

```
.padre {  
    align-content: space-around;  
}
```

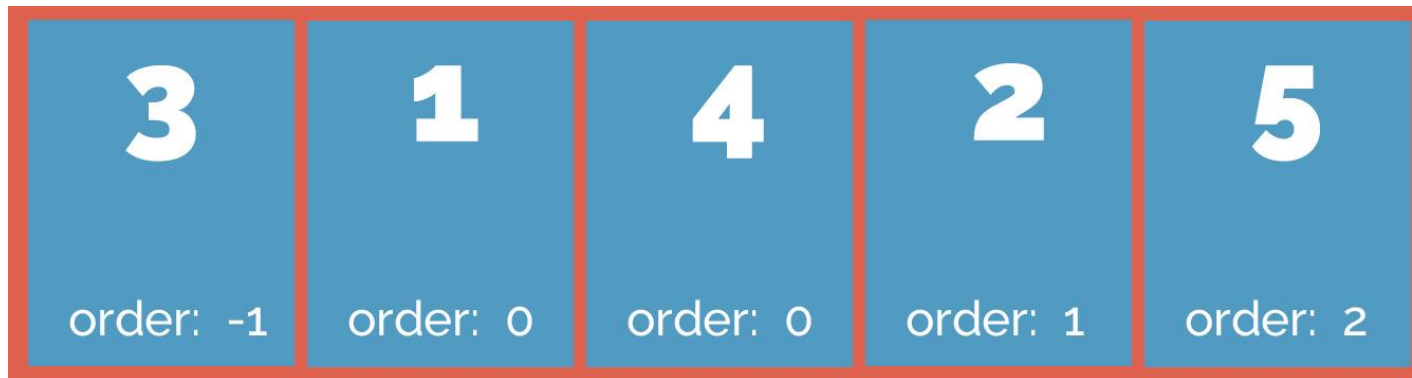
- Los elementos - flex tienen la misma separación en cada fila de elementos - flex.



Propiedades de los elementos - flex (Hijos)

order

- ▶ Con esta propiedad controlamos el orden de los elementos - flex que están dentro del contenedor - flex.
- ▶ Los elementos - flex pueden ser reordenados con esta simple propiedad sin la necesidad de reestructurar el código HTML.



flex-grow

- Esta propiedad especifica el factor que determina hasta que punto un elemento - flex crecerá en relación con el resto de los elementos - flex.

```
.hijo{  
    flex-grow: <number>;  
}
```

flex-grow

- Si los elementos - flex tienen el mismo valor para flex-grow, todos van a tener el mismo tamaño, según el contenedor - flex.



flex-grow

- Al definir que en el segundo elemento - flex la propiedad flex-grow va a tener un valor de dos, este va a tomar un ancho que equivale a la suma de dos elementos.



flex-shrink

- Esta propiedad especifica cómo el elemento - flex se reducirá con respecto al resto de los elementos flexibles dentro del mismo contenedor.

flex-shrink

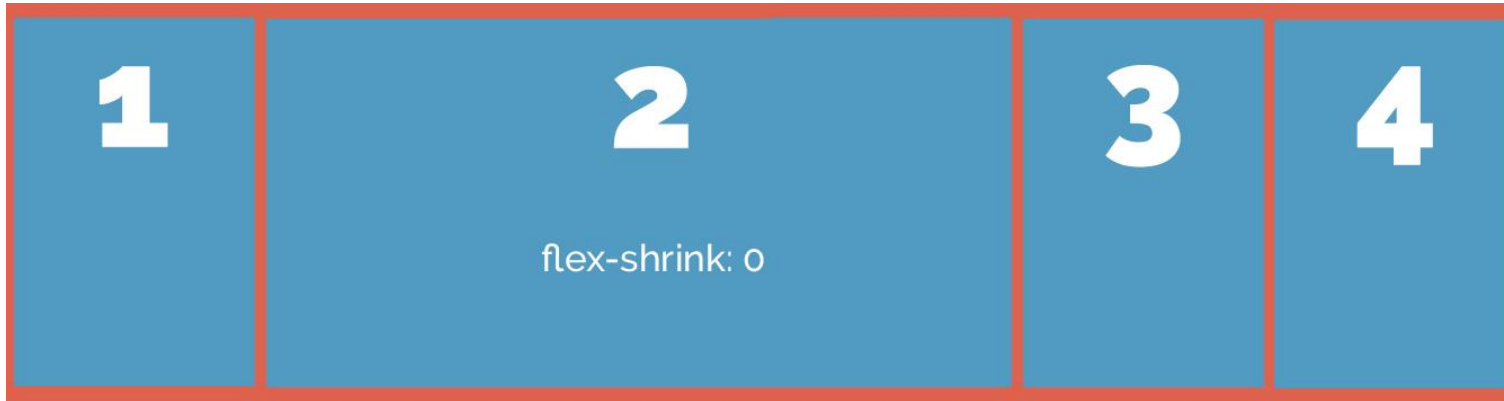
“

Valor por defecto: 1

```
.hijo{  
    flex-shrink: <number>;  
}
```

flex-shrink

- Por defecto, todos los **elementos - flex** se pueden reducir, pero si le ponemos el valor de 0 no se van a encoger, sino, que van a mantener el tamaño original.



flex-basis

- Esta propiedad tiene los mismos valores que las propiedades width y height y especifica el tamaño principal del elemento - flex, distribuyéndose de acuerdo con los factores flex.

flex-basis

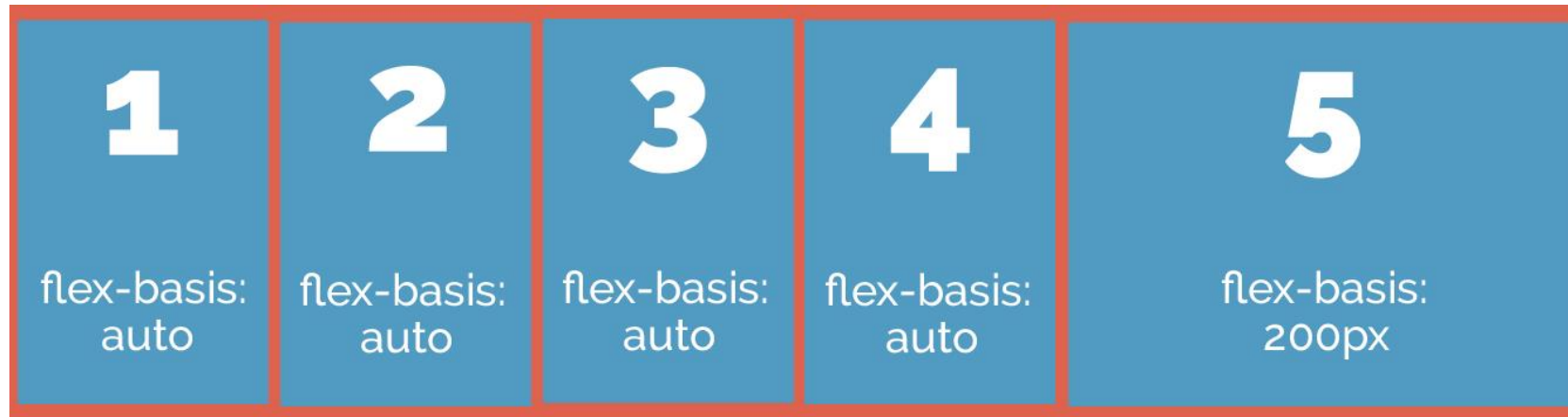
“

Valor por defecto: auto

```
.hijo{  
    flex-basis: auto | <width>;  
}
```

flex-basis

- Aquí especificamos que el quinto elemento - flex dicta el tamaño inicial del elemento.



align-self

- Esta propiedad permite la alineación por defecto (o la específica por align-items) para ser anulados por elementos - flex individuales.

align-self

“

Valor por defecto: auto

```
.hijo{  
  align-self: auto | flex-start | flex-end |  
  center | baseline | stretch;  
}
```

align-self

- El tercer y quinto elemento - flex han anulado la alineación a través de la propiedad align-self.



Bibliografía

- ▶ *FlexBox: La caja flexible con CSS 3 - Tutorial.* emenia. Consultado el 4 de septiembre del 2017, en <https://www.emenia.es/flexbox-la-caja-flexible-css3/>
- ▶ *El gran poder de CSS3: FlexBox.* (2015). Fili Santillán. Consultado el 4 de septiembre del 2017, en <https://filisantillan.com/el-gran-poder-de-css3-flexbox/>