

Flexbox

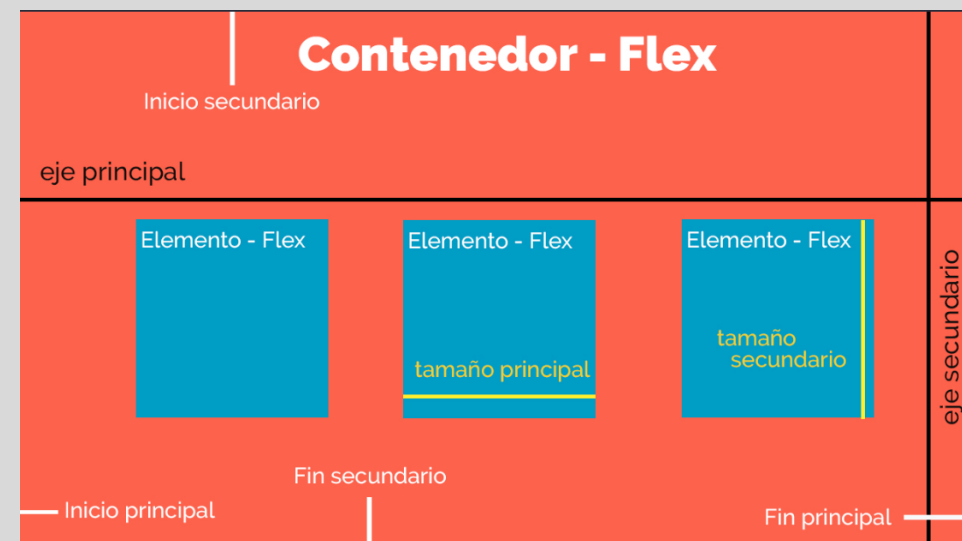
- Es una tecnología que proviene de css3, con la que se tiene acceso a un modelo de cajas flexibles que permite ordenar cualquier tipo de contenedores.
- Es uno de los valores de html5 con el que vamos a poder darle a la propiedad css “Display” y que va a permitir junto con otras propiedades maquetar paginas web de una manera mas fácil a la manera tradicional



Instructora Zaida Patricia Ojeda

Estructura de flexbox

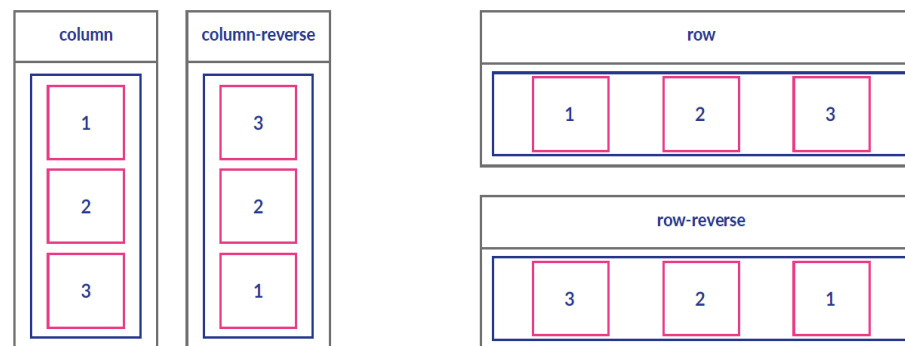
- **Contenedor- flex:** es el elemento padre que contiene los elementos hijos para definirlo usa flex en la propiedad display
- **Elemento – flex:** son los elementos hijos que tendrán un comportamiento automático dependiendo de lo que defina el padre
- **Ejes:** cada diseño FLEXBOX esta compuesto por 2 ejes



Propiedades

- **Flex direction:** ayuda a definir si el contenedor va a desplegar los hijos como columna o como fila

```
.contenedor {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



Propiedades

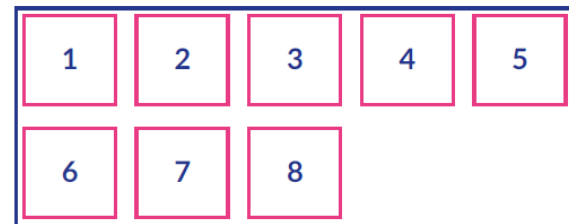
- **Flex wrap**: esta propiedad permite hacer flexible el contenedor

```
.contenedor {  
  flex-wrap: wrap; /* El contenedor puede caer si lo necesita */  
  flex-wrap: nowrap; /* El contenedor NO puede caer */  
  flex-wrap: wrap-reverse; /* El contenedor puede caer pero en el  
orden contrario */  
}
```

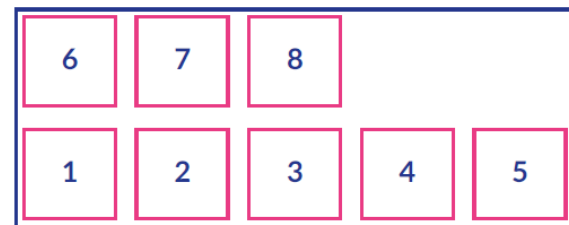
nowrap;



wrap;



wrap-reverse;



Propiedades

- **Flex-direction:** esta propiedad permite englobar en una sola las dos propiedades interior's

```
flex-flow: <flex-direction> <flex-wrap>;
```

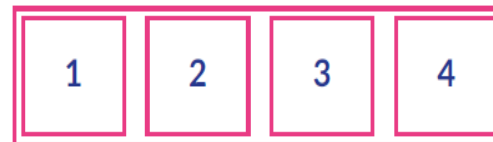
Ejemplo, para crear una fila que pueda llegar a convertirse en columna en dispositivos móviles usando la propiedad flex-Flow se necesitaría algo así:

```
flex-flow: row wrap;
```

flex-wrap: wrap;
flex-direction: row;



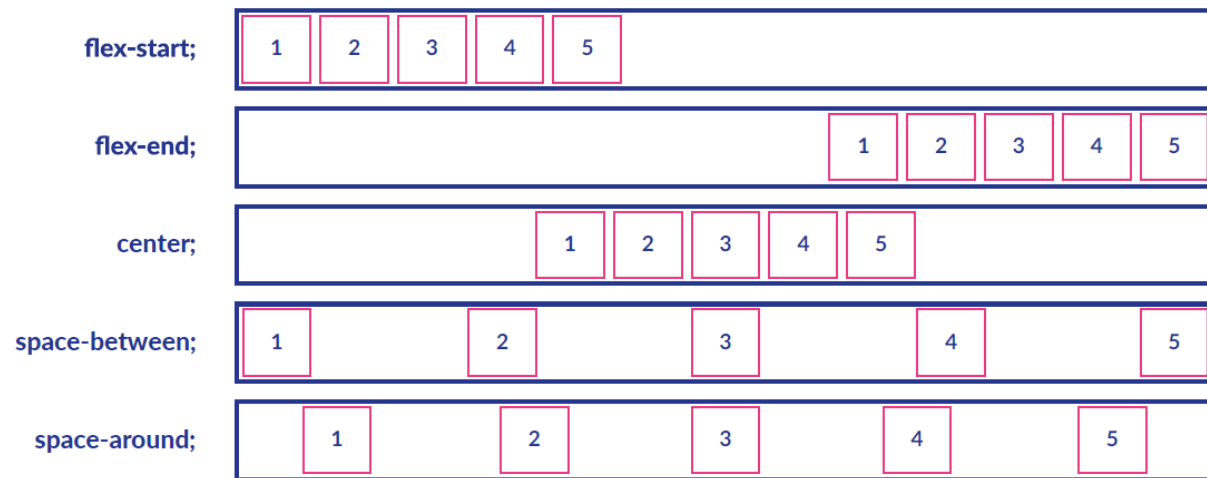
flex-flow: row wrap;



Propiedades para alinear

- **Justify-content:** ayuda a organizar los hijos del contenedor.
- Las propiedades mas destacadas son space-around y space-between que calculan el espacio disponible despues de colocar los hijos y lo repartira a los lados y entre ellos respectivamente.

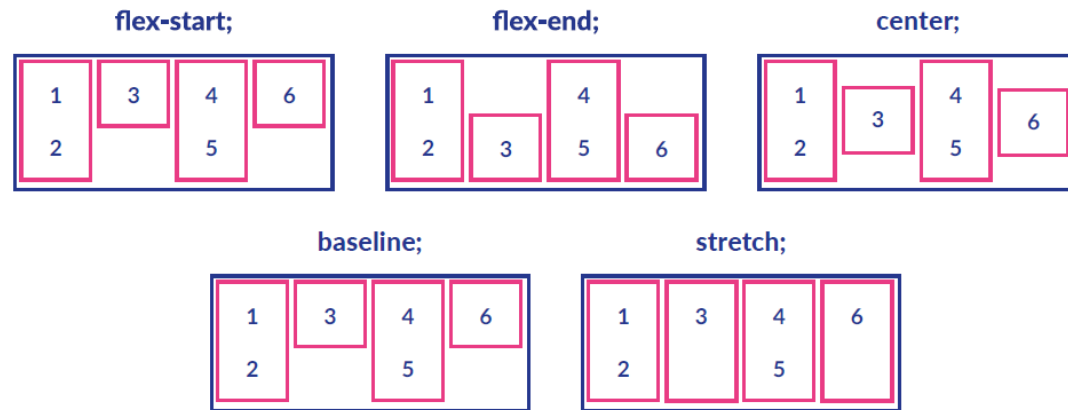
```
.contenedor {  
  justify-content: flex-start | flex-end | center | space-between  
  space-around | space-evenly;  
}
```



Propiedades para alinear

- **Align-items**: ayuda a trabajar con el espacio restante y con sus hijos, con la diferencia que considera a los hijos como un conjunto, al que procede a alinear

```
.contenedor {  
  align-items: stretch | flex-start | flex-end | center | baseline;  
}
```



Propiedades para ordenar

- **Order:** sirve para definir en cada hijo que orden va a tener el contenedor



```
.flex-item:nth-of-type(2) { order: 4; }  
.flex-item:nth-of-type(3) { order: 1; }  
.flex-item:nth-of-type(4) { order: 5; }  
.flex-item:nth-of-type(5) { order: 2; }
```


Ejemplo:

- Creamos una estructura html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, user-scalable=no,
  initial-scale=1, maximum-scale=1, minimum-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/flex.css">
  <title></title>
</head>
<body>
  <div class="contenedor">
    <div class="caja uno">1</div>
    <div class="caja dos">2</div>
    <div class="caja tres">3</div>
  </div>
</body>
</html>
```

Ejemplo:

- En el Css aplicamos los siguientes estilos

```
*{  
  margin:0;  
  padding: 0;  
  -webkit-box-sizing:border-box;  
  -moz-box-sizing:border-box;  
  box-sizing: border-box;  
}  
  
body{  
  background: #f2f2f2;  
  font-family: Arial;  
}  
  
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
}
```

```
.caja{  
  width: 100px;  
  /*height: 100px;*/  
  background: #f79256;  
  text-align:center;  
  margin: 20px;  
  padding: 20px;  
  font-size: 40px;  
  color:#fff;  
}
```



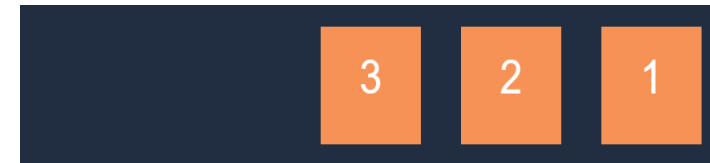
Ejemplo:

- Propiedades para el elemento *padre*

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row-reverse;  
}
```



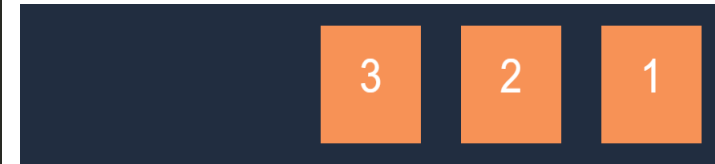
Ejemplo:

- Propiedades para el elemento padre

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row-reverse;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: column;  
}
```

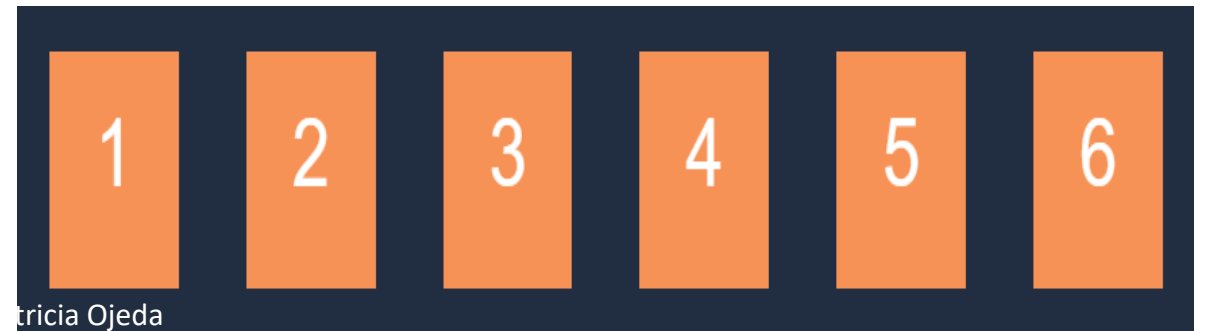


Ejemplo:

- Propiedades para el elemento padre.
- Para el ejemplo si agrego mas elementos en el html

```
<body>  
  <div class=contenedor>  
    <div class="caja uno">1</div>  
    <div class="caja dos">2</div>  
    <div class="caja tres">3</div>  
    <div class="caja tres">4</div>  
    <div class="caja tres">5</div>  
    <div class="caja tres">6</div>  
  </div>
```

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
}
```



Ejemplo:

Si agregamos la propiedad
`flex-wrap: wrap;`

Que es el valor que viene por
defecto, respeta los valores de la
caja

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
}
```



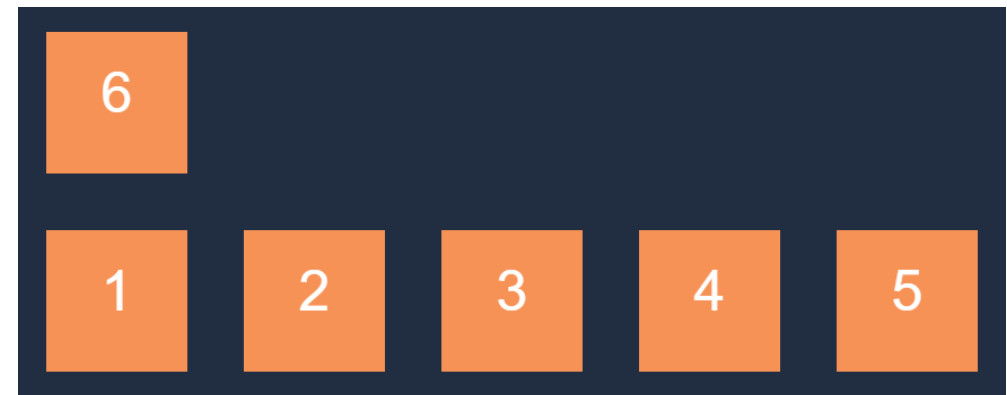
Ejemplo:

Si agregamos la propiedad
`flex-wrap: wrap-reverse;`

* Otra propiedad que permite
hacer lo mismo, en una sola
instrucción es:

`flex-flow: row wrap;`

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap-reverse;  
}
```



Ejemplo:

Para alinear los elementos,
usamos justify content:

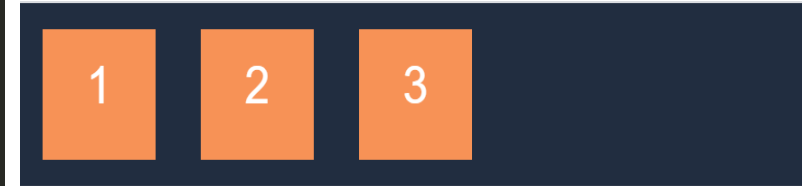
Con los valores

flex-start

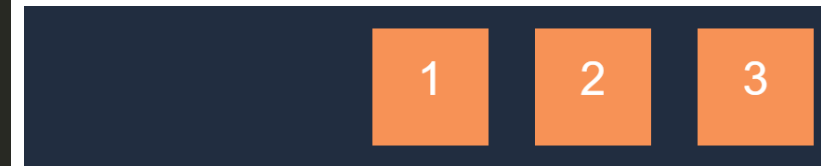
Flex-end

center

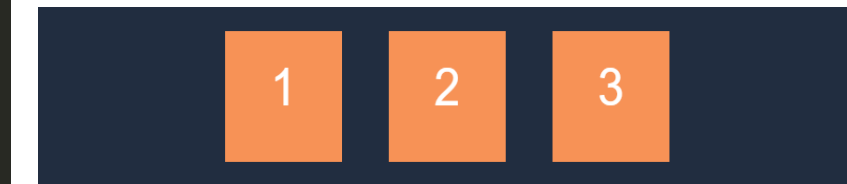
```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: flex-start;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: flex-end;  
}
```



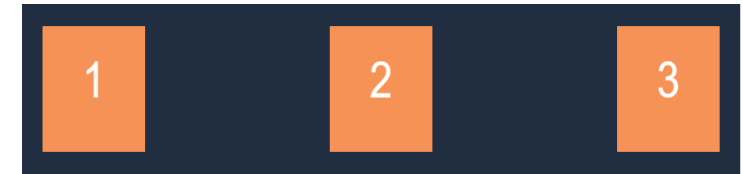
```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: center;  
}
```



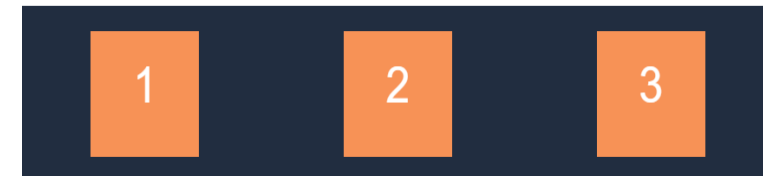
Ejemplo:

Para alinear los elementos:
Justify-content con los valores
Space-between
Space-around

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: space-between;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  /*height: 500px;*/  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: space-around;  
}
```



Ejemplo:

Vamos a dar un alto (height) al contenedor para evidenciar de mejor forma como funciona la propiedad

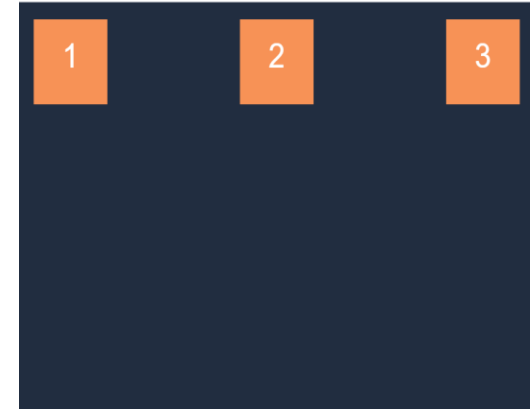
Align-ítems, con los valores

Flex-start

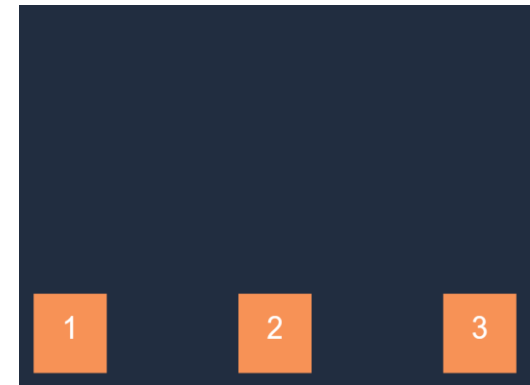
Flex-end

center

```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  height: 500px;  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: space-between;  
  align-items: flex-start;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  height: 500px;  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: space-between;  
  align-items: flex-end;  
}
```



```
.contenedor{  
  width: 80%;  
  max-width: 700px;  
  height: 500px;  
  background: #212d40;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  /*flex-flow: row wrap; */  
  justify-content: space-between;  
  align-items: center;  
}
```



Ejemplo:

Para que se pueda ver este siguiente ejemplo comentamos el alto de la caja

`align-items: stretch;`

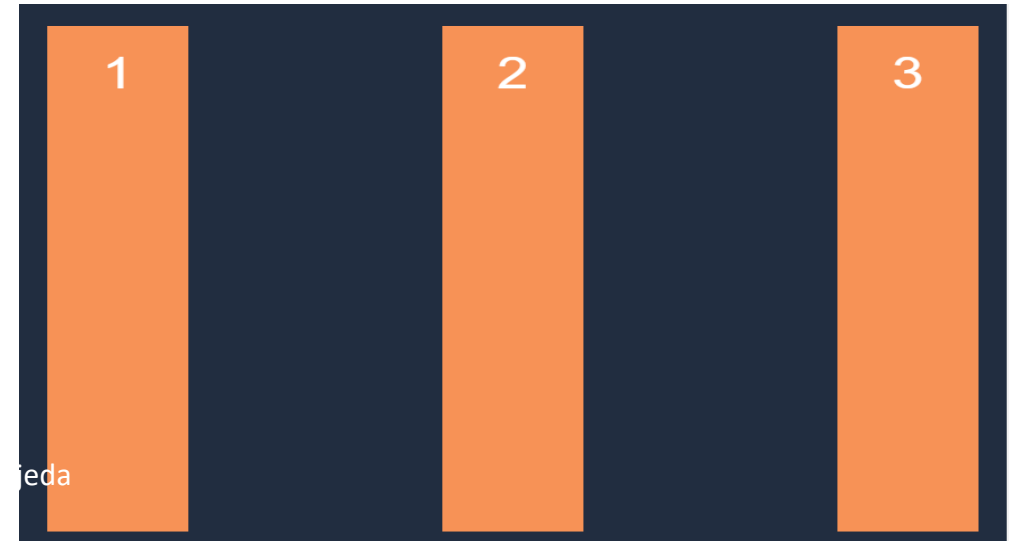
Se alarga hasta que abarque el 100% del contenedor, respeta el padding

`*align-items: baseline;`

Baseline: alinea con respecto a la fuente

```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  align-items: stretch;
}

.caja{
  width: 100px;
  /*height: 100px;*/
  background: #f79256;
  text-align: center;
  margin: 20px;
  padding: 20px;
  font-size: 40px;
  color: #fff;
}
```



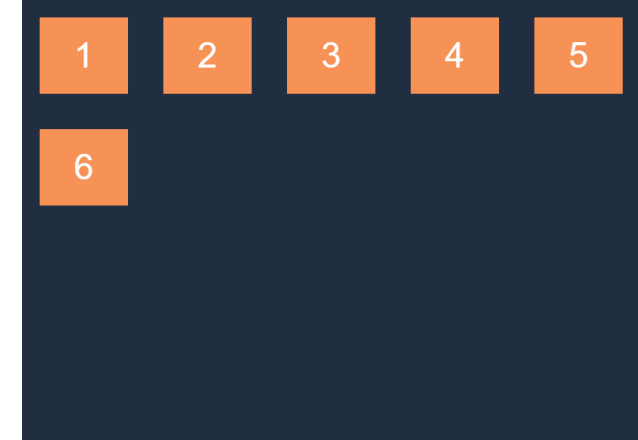
Ejemplo:

Otra propiedad es

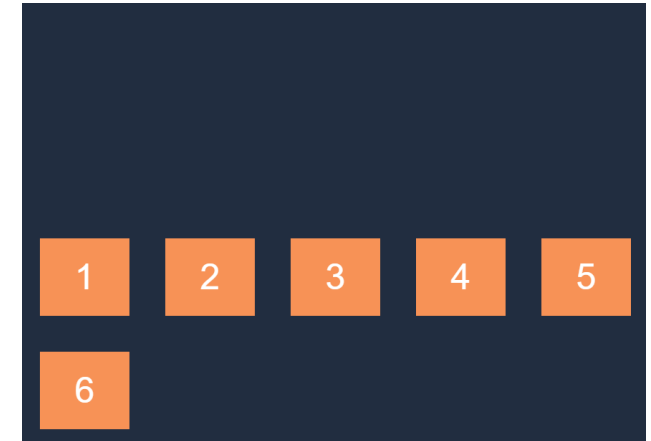
Align-content es similar a align-items, pero solo se puede usar una de las dos.

Permite centrar verticalmente, es mas útil cuando tenemos mas de una línea mas elementos de esta manera:

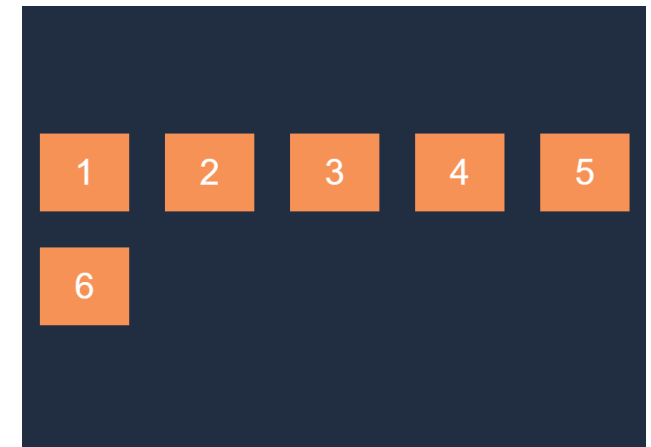
```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: flex-start;
}
```



```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: flex-end;
}
```



```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: center;
}
```



Ejemplo:

Otra propiedad es

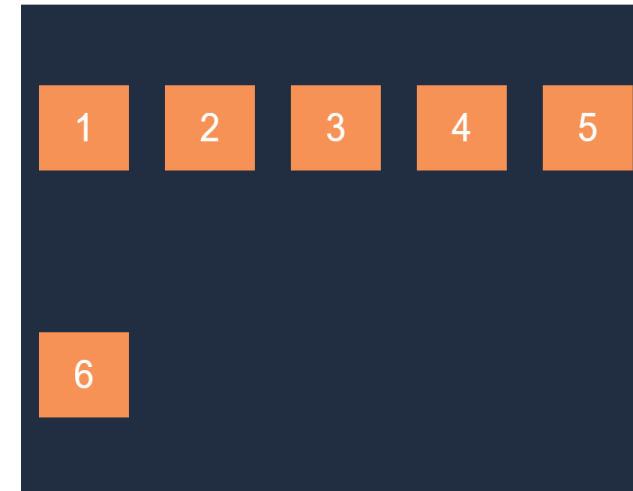
Align-content es similar a align-items, pero solo se puede usar una de las dos.

Permite centrar verticalmente, funciona cuando hay mas de una línea mas elementos de esta manera:

```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: space-between;
}
```



```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: space-around;
}
```



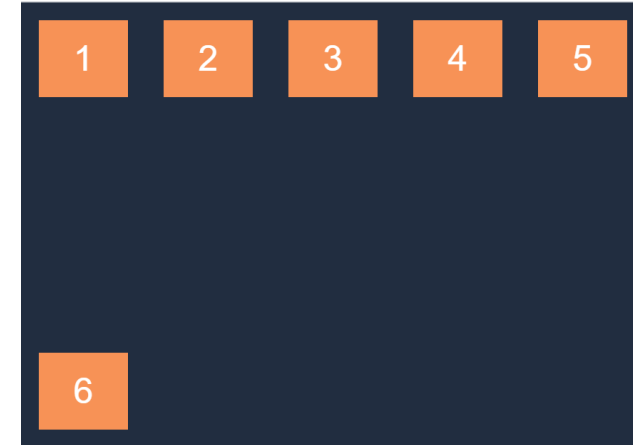
Ejemplo:

Otra propiedad es

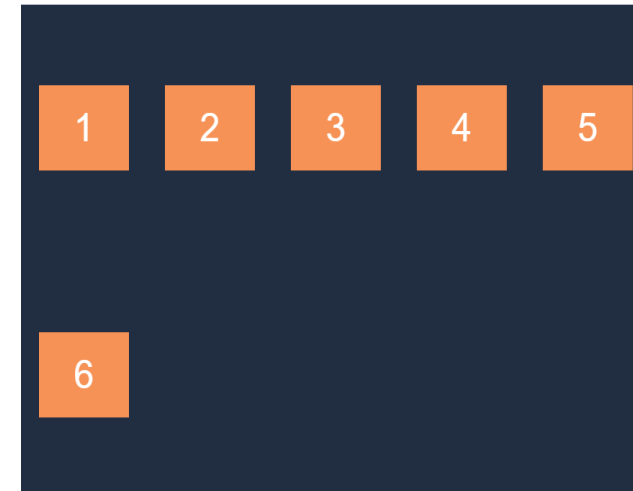
Align-content es similar a align-items, pero solo se puede usar una de las dos.

Permite centrar verticalmente, funciona cuando hay mas de una línea mas elementos de esta manera:

```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: space-between;
}
```



```
.contenedor{
  width: 80%;
  max-width: 700px;
  height: 500px;
  background: #212d40;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  /*flex-flow: row wrap; */
  justify-content: space-between;
  /*align-items: baseline;*/
  align-content: space-around;
}
```



Ejemplo:

Propiedades para el elemento hijo,

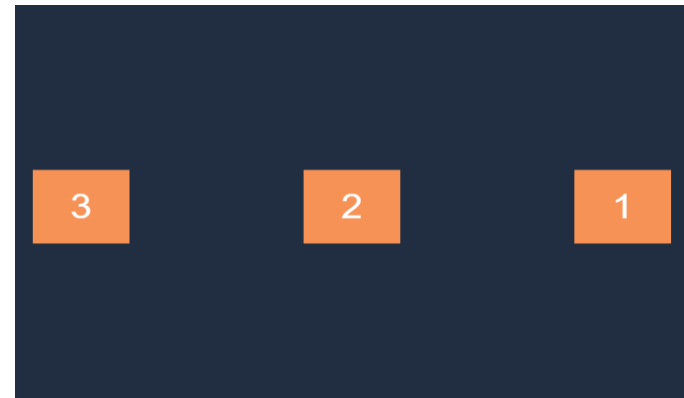
Es decir a la caja.

Se agregaron otras clases a las cajas

La primera propiedad es ORDER , para cambiar el orden que tiene cada una

```
<body>
  <div class=contenedor>
    <div class="caja uno">1</div>
    <div class="caja dos">2</div>
    <div class="caja tres">3</div>
  </div>
```

```
.uno{
  order: 3
}
.dos{
  order: 2;
}
.tres{
  order: 1
}
```



Ejemplo:

Propiedad flex-grow: acepta un numero que permite hacer crecer el elemento en relación con los demás, por defecto tiene un valor de 1 .

Si por ejemplo cambio el ancho de la caja y agrego otra caja, todas las cajas tienen un ancho mínimo de 150px, pero como la caja 4 no cabe se adapta y se pone debajo y como tiene un flex-grow de 1 abarca el 100%.

En el eventual caso que quiera modificar

Hay que tener en cuenta que como

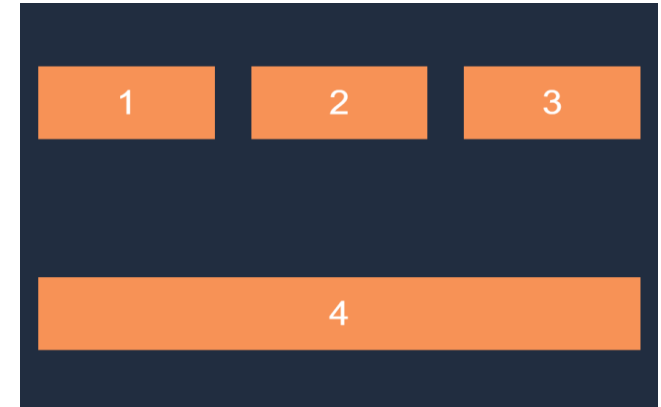
Ocupa mas de una línea se debe usar

Align-content

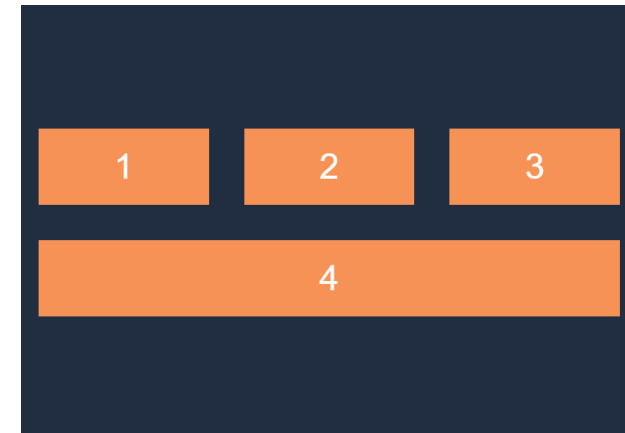
```
.caja{  
  width: 100px;  
  /*height: 100px;*/  
  background: #f79256;  
  text-align:center;  
  margin: 20px;  
  padding: 20px;  
  font-size: 40px;  
  color:#fff;  
  flex-grow: 1;  
}
```



```
.caja{  
  width: 150px;  
  /*height: 100px;*/  
  background: #f79256;  
  text-align:center;  
  margin: 20px;  
  padding: 20px;  
  font-size: 40px;  
  color:#fff;  
  flex-grow: 1;  
}
```



Align-content:center



Ejemplo:

Propiedad flex-shrink: es la habilidad que tiene un elemento para hacerse mas pequeño en proporción a los demás, de acuerdo con el tamaño que tenga definido

*otra propiedad es flex-basis

Que va a decir el tamaño inicial que tiene un elemento , similar a un width

```
.caja{
  width: 100px;
  /*height: 100px;*/
  background: #f79256;
  text-align:center;
  margin: 20px;
  padding: 20px;
  font-size: 40px;
  color:#fff;
  flex-grow: 1;
}

.dos{
  flex-shrink: 0;
  width: 50px;
}
```



```
.dos{
  flex-shrink: 0;
  flex-basis: 200px;
}
```

