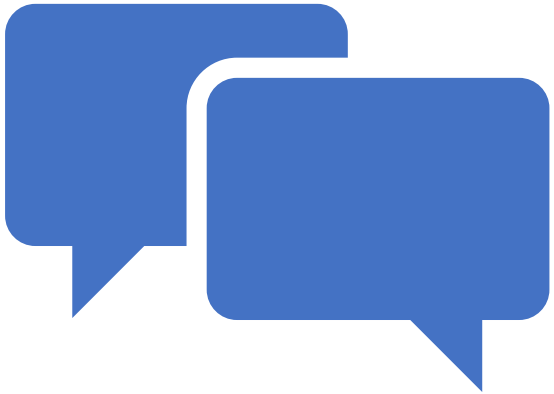
Flexbox



Media Queries

Puntos de corte (Breakpoints)

```
.example {
  padding: 20px;
  color: white;
}
/* Extra small devices (600px and down) */ @media only screen and (max-width: 600px) {
  .example {background: red;}
}
/* Small devices (600px and up) */
@media only screen and (min-width: 600px) {
  .example {background: green;}
}
/* Medium devices (768px and up) */
```

```
/* Large devices (992px and up) */
@media only screen and (min-width: 992px) {
    .example {background: orange;}
}
/* Extra large devices (1200px and up) */
@media only screen and (min-width: 1200px) {
    .example {background: pink;}
}
```

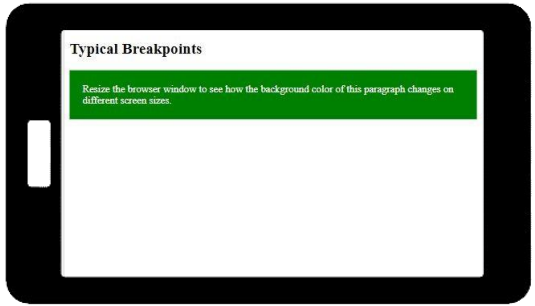
Puntos de corte (según ancho):

- Hasta 600 px: Fondo rojo
- Desde 600 px: Fondo verde
- Desde 768 px: Fondo azul
- Desde 992 px: Fondo naranja
- Desde 1200 px: Fondo rosa

Puntos de corte (Breakpoints)

400px X 600px

Extra small devices (phones, 600px and down)



650px X 400px

Small devices (portrait tablets and large phones, 600px and up)

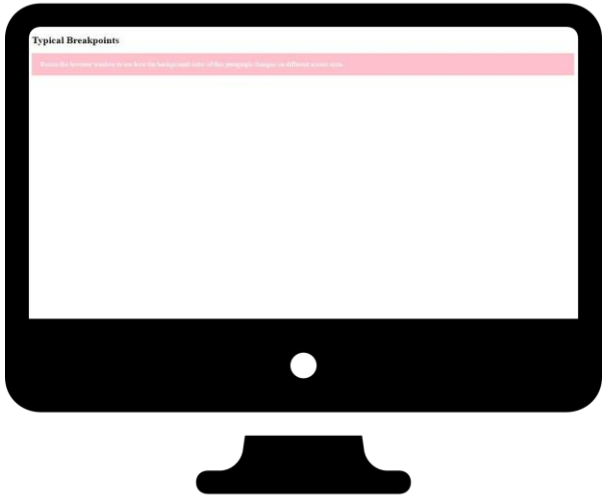
850px X 600px

Medium devices (landscape tablets, 768px and up)

Puntos de corte (Breakpoints)

1000px X 800px

Large devices (laptops/desktops, 992px and up)



1300px X 800px

Extra large devices (large laptops and desktops, 1200px and up)

Flexbox

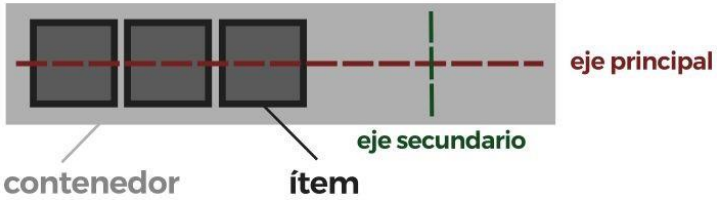
¿Qué es Flexbox?

Tradicionalmente, en CSS se ha utilizado el posicionamiento (*static, relative, absolute...*), los elementos en línea o en bloque (*y derivados*) o los **float**, lo que a grandes rasgos no deja de ser un sistema de creación de diseños bastante tradicional que no encaja con los retos que tenemos hoy en día: sistemas de escritorio, dispositivos móviles, múltiples resoluciones, etc…

Flexbox es un sistema de **elementos flexibles** que llega con la idea de olvidar estos mecanismos y acostumbrarnos a una mecánica más potente, limpia y personalizable, en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños. Está especialmente diseñado para crear, mediante CSS, **estructuras de una sola dimensión.** +info

Flexbox | Conceptos

Elementos básicos de Flexbox:



- **Contenedor:** Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles.
  - **Eje principal:** Los contenedores flexibles tienen una orientación principal específica. Por defecto es el eje horizontal.
  - **Eje secundario:** La orientación secundaria es perpendicular a la principal.
- **Ítem:** Son los elementos hijos flexibles del contenedor.

Flexbox | Conceptos

Imaginemos el siguiente escenario:

```
<div class="container"> <!-- Flex container -->
    <div class="item item-1">1</div> <!-- Flex items -->
    <div class="item item-2">2</div>
    <div class="item item-3">3</div>
</div>
```

Para activar el modo **flexbox**, hemos utilizado sobre el elemento contenedor la propiedad **display** y especificamos el valor **flex** o **inline-flex** (dependiendo de cómo queramos que se comporte el contenedor)

Flexbox | Conceptos

Propiedad **display**:

Tipo de elemento	Descripción
`inline-flex`	Establece un contenedor en línea, similar a `inline-block` (ocupa solo el contenido).
`flex`	Establece un contenedor en bloque, similar a `block` (ocupa todo el ancho del padre).

Flexbox | Dirección de los ejes

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems a lo largo del eje principal del contenedor. Son las siguientes:

Propiedad	Valor	Significado
`flex-direction`	**row** | row-reverse | column | column-reverse	Cambia la orientación del eje principal.
`flex-wrap`	**nowrap** | wrap | wrap-reverse	Evita o permite el desbordamiento (multilinea).

Flexbox | Dirección de los ejes

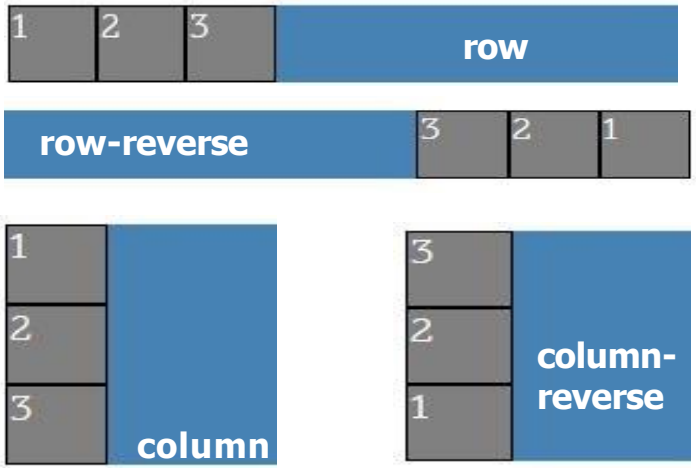
- Mediante la propiedad **flex-direction** podemos modificar la dirección del **eje principal** del contenedor para que se oriente en horizontal (*por defecto*) o en vertical. Además, también podemos incluir el sufijo **-reverse** para indicar que coloque los ítems en orden inverso.

Valor	Descripción
`row`	Establece la dirección del eje principal en horizontal.
`row-reverse`	Establece la dirección del eje principal en horizontal (invertido).
`column`	Establece la dirección del eje principal en vertical.
`column-reverse`	Establece la dirección del eje principal en vertical (invertido).

Flexbox | row y row-reverse

row y **row-reverse** determinan el orden de los elementos. Aplicando estas propiedades modificamos el flujo del eje principal:

```
.container {
    background: steelblue;
    display: flex;
    flex-direction: column;
}

.item {
    background: grey;
}
```



Flexbox | flex-wrap

- Existe otra propiedad llamada **flex-wrap** con la que podemos especificar el comportamiento del contenedor respecto a evitar que se desborde (*nowrap*, *valor por defecto*) o permitir que lo haga, en cuyo caso, estaríamos hablando de un **contenedor flexbox multilínea**.

Valor	Descripción
nowrap	Establece los ítems en una sola línea (no permite que se desborde el contenedor).
wrap	Establece los ítems en modo multilínea (permite que se desborde el contenedor).
wrap-reverse	Establece los ítems en modo multilínea, pero en dirección inversa.

Flexbox | flex-wrap

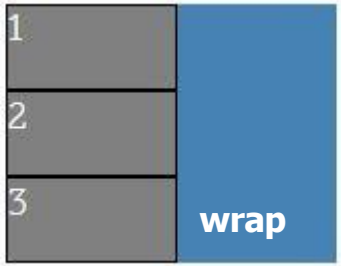
Teniendo en cuenta estos valores de la propiedad **flex-wrap**, podemos conseguir cosas como la siguiente:

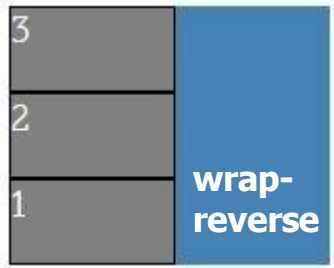
```
.container {
    background: steelblue;
    display: flex;
    width: 200px;
    flex-wrap: wrap;   /* Comportamiento por defecto: nowrap */
}

.item {
    background: grey;
    width: 50%;
}
```

Flexbox | flex-wrap

Con **nowrap** los 3 ítems se muestran en una misma línea. El tamaño de los items se ajustan al contenedor, manteniendo sus proporciones.

Si especificamos **wrap** el contenedor se puede desbordar, pasando a ser un contenedor multilínea que muestra uno o más elementos en la línea siguiente.





Flexbox | Atajo: dirección de los ejes

- Existe una propiedad de atajo (*short-hand*) llamada **flex-flow**, con la que podemos resumir los valores de las propiedades **flex-direction** y **flex-wrap**, especificandose en una sola propiedad y ahorrándonos utilizar las propiedades concretas:

```
.container {
    /* flex-flow: <flex-direction> <flex-wrap>; */
    flex-flow: row wrap;
}
```

Flexbox | Propiedades de alineación

Disponemos de 4 propiedades relativas a la alineación, la primera relativa al eje principal y las restantes al secundario:

Propiedad	Valor	Eje
justify-content	**flex-start** | flex-end | center | space-between | space-around | space-evenly	1
align-content	flex-start | flex-end | center | space-between | space-around | space-evenly | **stretch**	2
align-items	flex-start | flex-end | center | **stretch** | baseline	2
align-self	**auto** | flex-start | flex-end | center | stretch | baseline	2

- **justify-content:** Alinea los ítems del eje principal.
- **align-items:** Alinea los ítems del eje secundario.

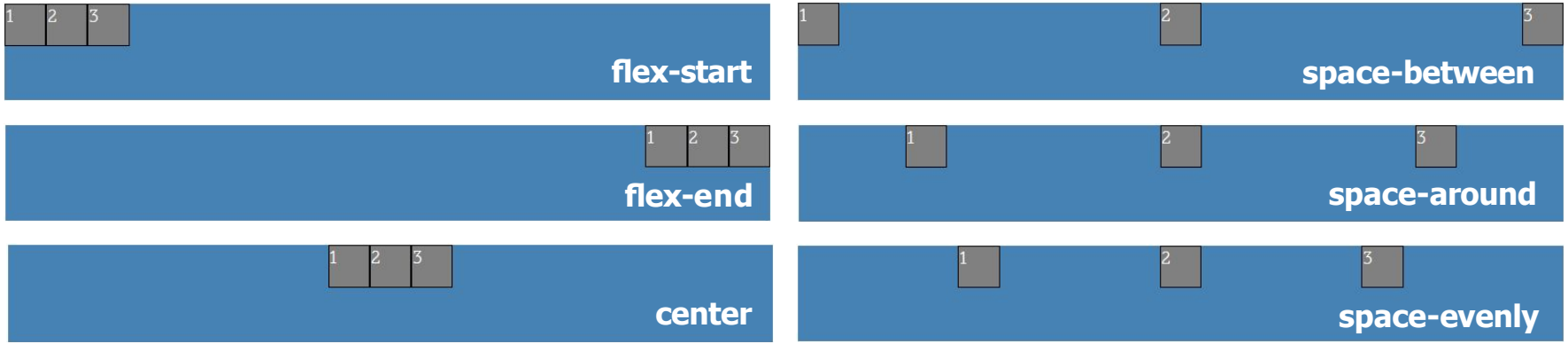
Flexbox | Eje principal

La propiedad **justify-content** sirve para colocar los ítems de un contenedor mediante una disposición concreta a lo largo del **eje principal**:

Valor	Descripción
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems dejando el máximo espacio para separarlos.
space-around	Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).
space-evenly	Distribuye los ítems dejando el mismo espacio (solapado) a izquierda y derecha.

Flexbox | Eje principal

Con estos valores de la propiedad **justify-content** modificamos la disposición de los ítems del contenedor, distribuyéndose como se ve en el siguiente ejemplo (nótense los números para observar el orden de cada ítem):



Flexbox | Eje principal

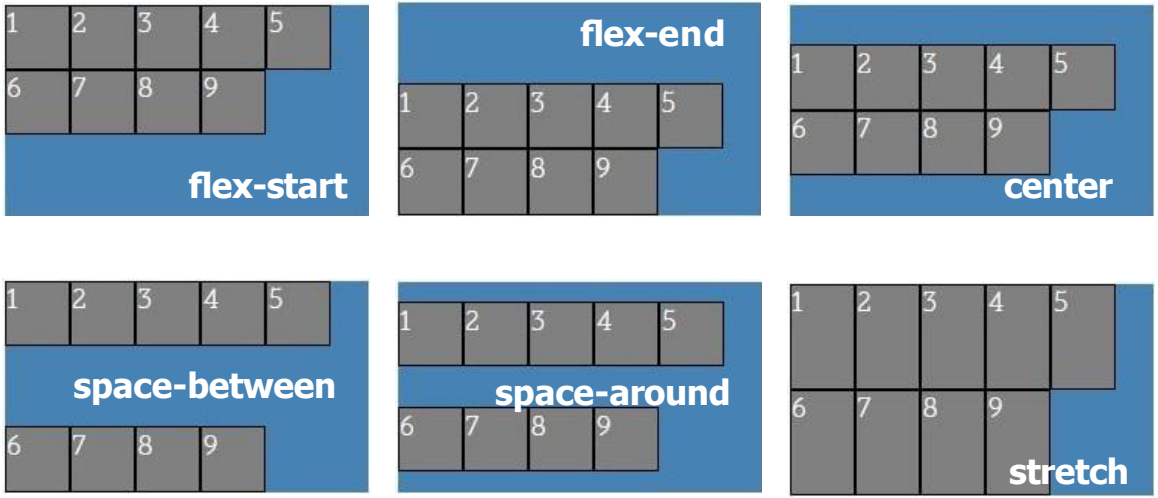
align-content permite manejar contenedores **flex multilínea**. Estos contenedores dividen el eje principal en múltiples líneas, dado que los ítems no caben en el ancho disponible. Sus valores son los siguientes:

Valor	Descripción
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems desde el inicio hasta el final.
space-around	Distribuye los ítems dejando el mismo espacio a los lados de cada uno.
stretch	Estira los ítems para ocupar de forma equitativa todo el espacio.

Flexbox | Eje principal

En un contenedor multilínea de 200 píxeles de alto con ítems de 50px de alto, podemos utilizar la propiedad **align-content** para alinear los ítems de forma vertical de modo que se queden en la zona inferior del contenedor:

```
.container {
    background: #CCC;
    display: flex;
    width: 200px;
    height: 200px;
    flex-wrap: wrap;
    align-content: flex-end;
}
.item {
    background: #777;
    width: 50%;
    height: 50px;
}
```



Flexbox | Eje secundario

align-items alinea los ítems en el *eje secundario* del contenedor. A diferencia de **align-content**, **align-items** opera sobre el eje secundario. Los valores que puede tomar son los siguientes:

Valor	Descripción
flex-start	Alinea los ítems al principio del eje secundario.
flex-end	Alinea los ítems al final del eje secundario.
center	Alinea los ítems al centro del eje secundario.
stretch	Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
baseline	Alinea los ítems en el contenedor según la base del contenido de los ítems del contenedor.

Puedes ver un ejemplo interactivo aquí.

Flexbox | Eje secundario

align-self actúa como **align-items**, pero sobre un ítem hijo específico, sobreescribiendo su comportamiento. La propiedad puede tomar los siguientes valores:

Valor	Descripción
flex-start	Alinea los ítems al principio del contenedor.
flex-end	Alinea los ítems al final del contenedor.
center	Alinea los ítems al centro del contenedor.
stretch	Alinea los ítems estirándolos al tamaño del contenedor.
baseline	Alinea los ítems en el contenedor según la base de los ítems.
auto	Hereda el valor de **align-items** del padre (si no se ha definido, es **stretch**).

Flexbox | Eje secundario

Si se especifica el valor **auto** a la propiedad **align-self**, el navegador le asigna el valor de la propiedad **align-items** del contenedor padre, y en caso de no existir, el valor por defecto **stretch**. Ver segundo ejemplo interactivo aquí.

Existe un atajo para establecer valores de **align-content** y de **justify-content** a la vez, denominada **place-content**. Las dos clases siguientes proporcionan las mismas características:

```
.container {
    display: flex;
    place-content: flex-start flex-end;
}
```

```
.container {
    align-content: flex-start;
    justify-content: flex-end;
}
```

Flexbox | Propiedades de hijos

Las siguientes propiedades se aplican sobre los ítems hijos:

Propiedad	Valor	Descripción
flex-grow	0 | NUMBER	Número que indica el factor de crecimiento del ítem respecto al resto.
flex-shrink	1 | NUMBER	Número que indica el factor de decrecimiento del ítem respecto al resto.
flex-basis	SIZE | content	Tamaño base de los ítems antes de aplicar variación.
order	0 | NUMBER	Número (peso) que indica el orden de aparición de los ítems.

Si con **flex-grow** indicamos un valor de 1a todos los ítems, todos serán del mismo tamaño. Si colocamos un valor de 1a todos, pero a uno le indicamos 2, ese ítem será más grande que los anteriores. Por defecto tienen un valor de 0.

Flexbox | Propiedades de hijos

flex-shrink es opuesta a **flex-grow**, aplica un factor de decrecimiento. De esta forma, los ítems que tengan un valor numérico más grande, serán más pequeños, mientras que los que tengan un valor numérico más pequeño serán más grandes, justo al contrario de como funciona la propiedad **flex-grow**.

Por último, tenemos la propiedad **flex-basis**, que define el tamaño por defecto (de base) que tendrán los ítems antes de aplicarle la distribución de espacio. Generalmente, se aplica un tamaño (unidades, porcentajes, etc...), pero también se puede aplicar la palabra clave **content** que ajusta automáticamente el tamaño al contenido del ítem, que es su valor por defecto.

Flexbox | Atajo: Propiedades de hijos

Existe una propiedad llamada **flex** que sirve de atajo para estas tres propiedades de los ítems hijos. Funciona de la siguiente forma:

```
.item {
    /* flex: <flex-grow> <flex-shrink> <flex-basis> */
    flex: 1 3 35%;
}
```

Flexbox | Huecos (gaps)

Las propiedades **row-gap** y **column-gap** establecen el tamaño del «hueco» entre ítems desde el elemento contenedor, sin necesidad de utilizar **padding** o **margin** en los elementos hijos.

Propiedad	Valor	Descripción
row-gap	**normal** | SIZE	Espacio entre filas (sólo si flex-direction: column)
column-gap	**normal** | SIZE	Espacio entre columnas (sólo si flex-direction: row)

Sólo una de las dos propiedades tendrá efecto, dependiendo de si **flex-direction** está establecida en **column** o en **row**. Es posible usar ambas cuando **flex-wrap: wrap**, disponiendo de multicolumnas flexbox. Ver ejemplo interactivo aquí.

Flexbox | Orden de los ítems

Por último, y quizás una de las propiedades más interesantes, es **order**, que modifica y establece el orden de los ítems según una secuencia numérica.

Por defecto, todos los ítems flex tienen un **order: 0** implícito, aunque no se especifique. Si indicamos un **order** con un valor numérico, se reubican los ítems según su número, colocando primero los ítems con número más pequeño (incluso valores negativos) y después los ítems con números más altos. De esta forma reordenamos fácilmente los ítems, incluso utilizando *media queries* o *responsive design*.

Ver ejemplo interactivo aquí.