

INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

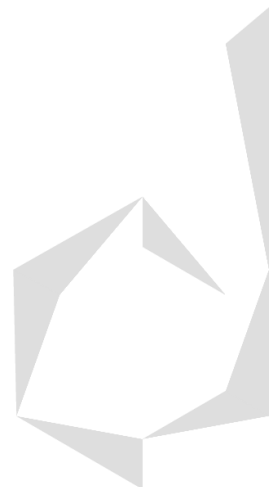
DESARROLLO FULL STACK

HTML, CSS & JAVASCRIPT

Flexbox CSS: Acomodar  
Contenedores y/o Etiquetas HTML

# Contenido

Flexbox de CSS .....	2
<b>Justify Content</b> – Alineación Horizontal de Elementos .....	2
<b>Align Items</b> – Alineación Vertical de Elementos.....	3
<b>Flex Direction</b> – Dirección de Elementos.....	6
<b>Order</b> – Cambiar el Orden de un Solo Elemento .....	8
<b>Align Self</b> – Alineación Vertical de un Solo Elemento.....	9
<b>Flex Wrap</b> – Distribución de Elementos en una o varias Filas .....	10
<b>Align Content</b> – Distribución o Separación de Vertical .....	12
Resumen de Flexbox .....	16
<b>Justify Content</b> – Alineación Horizontal de Elementos .....	17
<b>Align Items</b> – Alineación Vertical de Elementos.....	17
<b>Flex Direction</b> – Dirección de Elementos.....	17
<b>Order</b> – Cambiar el Orden de un Solo Elemento .....	18
<b>Align Self</b> – Alineación Vertical de un Solo Elemento.....	18
<b>Flex Wrap</b> – Distribución de Elementos en una o varias Filas .....	18
<b>Flex Flow</b> – Combinación de flex-direction y flex-wrap.....	19
<b>Align Content</b> – Distribución o Separación de Vertical .....	19
Referencias: .....	21



# Flexbox de CSS

Flexbox es una herramienta de CSS que sirve para acomodar elementos dentro de un DIV u otra etiqueta, para poder aplicar esta herramienta se debe utilizar la instrucción **display: flex;** así todos los elementos que se encuentren dentro del contenedor que indica la instrucción se acomodarán de cierta forma. Existe una página llamada flexbox froggy que explica su funcionamiento en forma de juego:

<https://flexboxfroggy.com/#es>

## Justify Content – Alineación Horizontal de Elementos

Alineación horizontal de los elementos pertenecientes a la fila de un contenedor con Flexbox.

**FLEXBOX FROGGY**Nivel 1 de 24 ▶

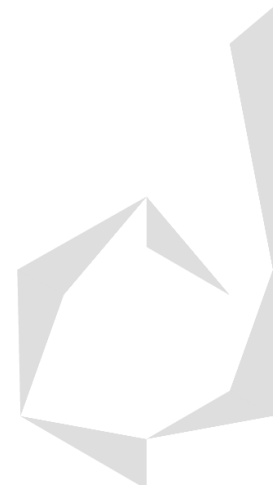
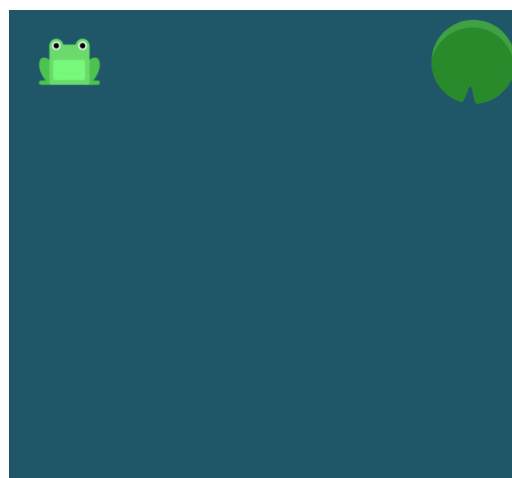
Bienvenido a Flexbox Froggy, un juego donde ayudarás a Froggy y a sus amigos escribiendo código CSS. Guía a esta rana hacia la hoja de lirio en la derecha, usando la propiedad **justify-content**, la cual alinea elementos horizontalmente y acepta los siguientes valores:

- **flex-start**: Alinea elementos al lado izquierdo del contenedor.
- **flex-end**: Alinea elementos al lado derecho del contenedor.
- **center**: Alinea elementos en el centro del contenedor.
- **space-between**: Muestra elementos con la misma distancia entre ellos.
- **space-around**: Muestra elementos con la misma separación alrededor de ellos.

Por ejemplo, **justify-content: flex-end;** moverá la rana a la derecha.

```
1 #pond {  
2   display: flex;  
3   justify-content: |  
4 }  
5  
6  
7  
8  
9  
10
```

Siguiente



## FLEXBOX FROGGY

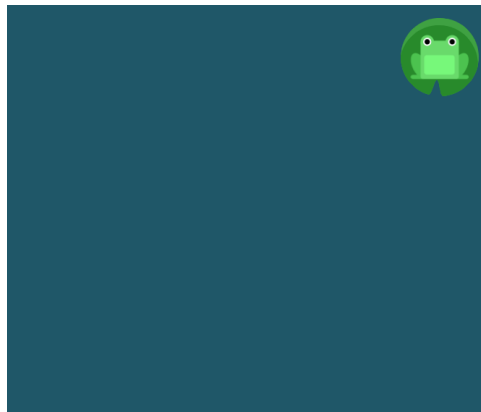
Nivel 1 de 24

Bienvenido a Flexbox Froggy, un juego donde ayudarás a Froggy y a sus amigos escribiendo código CSS. Guía a esta rana hacia la hoja de lirio en la derecha, usando la propiedad `justify-content`, la cual alinea elementos horizontalmente y acepta los siguientes valores:

- `flex-start`: Alinea elementos al lado izquierdo del contenedor.
- `flex-end`: Alinea elementos al lado derecho del contenedor.
- `center`: Alinea elementos en el centro del contenedor.
- `space-between`: Muestra elementos con la misma distancia entre ellos.
- `space-around`: Muestra elementos con la misma separación alrededor de ellos.

Por ejemplo, `justify-content: flex-end;` moverá la rana a la derecha.

```
1 #pond {
2   display: flex;
3   justify-content: flex-end;
4 }
5
```



## Align Items – Alineación Vertical de Elementos

Alineación vertical de los elementos pertenecientes a la fila de un contenedor con Flexbox.

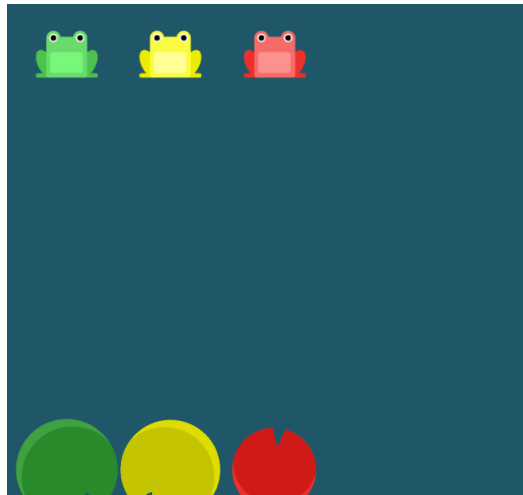
## FLEXBOX FROGGY

Nivel 5 de 24

Ahora usa `align-items` para ayudar a las ranas a llegar al fondo del estanque. Esta propiedad CSS alinea elementos verticalmente y acepta los siguientes valores:

- `flex-start`: Alinea elementos a la parte superior del contenedor.
- `flex-end`: Alinea elementos a la parte inferior del contenedor.
- `center`: Alinea elementos en el centro (verticalmente hablando) del contenedor.
- `baseline`: Muestra elementos en la línea base del contenedor
- `stretch`: Elementos se estiran para ajustarse al contenedor.

```
1 #pond {
2   display: flex;
3   align-items:
4 }
5
6
```



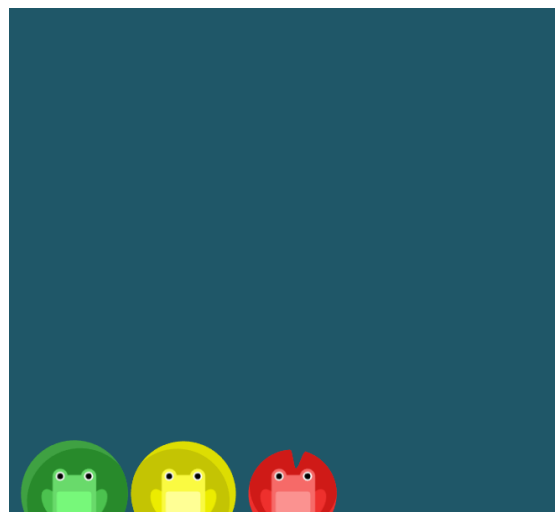
## FLEXBOX FROGGY

Nivel 5 de 24 ▾

Ahora usa **align-items** para ayudar a las ranas a llegar al fondo del estanque. Esta propiedad CSS alinea elementos verticalmente y acepta los siguientes valores:

- **flex-start**: Alinea elementos a la parte superior del contenedor.
- **flex-end**: Alinea elementos a la parte inferior del contenedor.
- **center**: Alinea elementos en el centro (verticalmente hablando) del contenedor.
- **baseline**: Muestra elementos en la línea base del contenedor
- **stretch**: Elementos se estiran para ajustarse al contenedor.

```
1 #pond {  
2   display: flex;  
3   align-items: flex-end;  
4 }  
5  
6
```



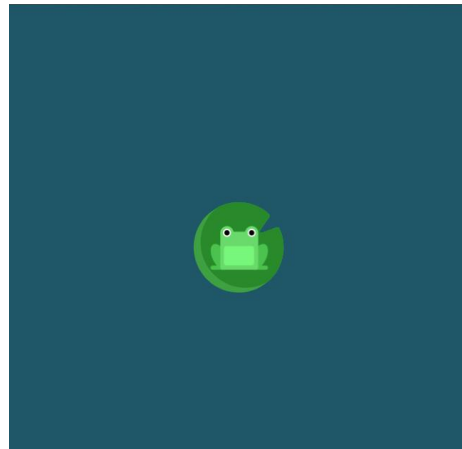
Alineación horizontal (justify content) y vertical (align items) con Flexbox.

### FLEXBOX FROGGY

Nivel 6 de 24

Mueve la rana al centro del estanque, usando una combinación de `justify-content` y `align-items`.

```
1 #pond {  
2   display: flex;  
3   justify-content: center;  
4   align-items: center;  
5 }
```

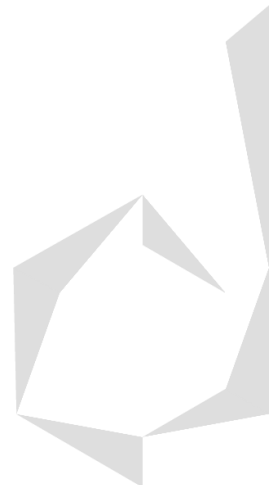
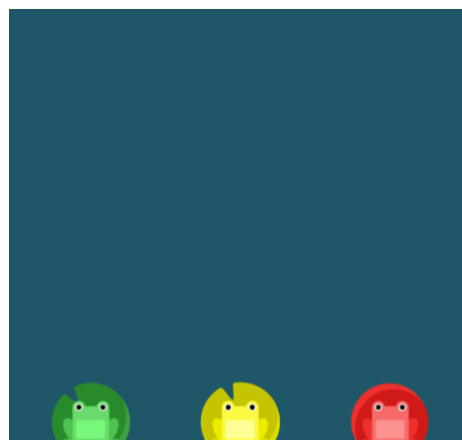


### FLEXBOX FROGGY

Nivel 7 de 24

Nuevamente, las ranas necesitan cruzar el estanque. En esta oportunidad, las hojas de lirio tienen mucho espacio alrededor de ellas. Debes usar una combinación de `justify-content` y `align-items`.

```
1 #pond {  
2   display: flex;  
3   align-items: flex-end;  
4   justify-content: space-around;  
5 }
```



## Flex Direction – Dirección de Elementos

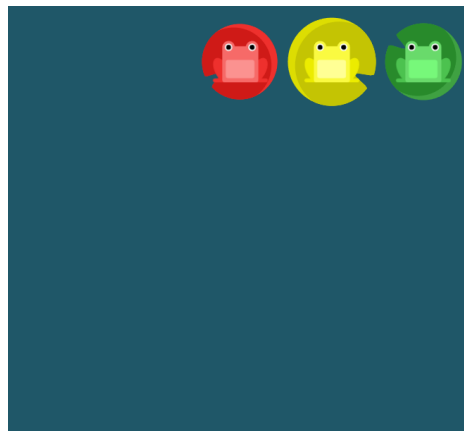
Indica la dirección en forma de fila, fila en reversa, columna o columna en reversa con Flexbox.

**FLEXBOX FROGGY** Nivel 8 de 24

Las ranas necesitan ponerse en el mismo orden que sus hojas de lirio usando **flex-direction**. Esta propiedad CSS define la dirección de los elementos en el contenedor, y acepta los siguientes valores:

- **row**: Elementos son colocados en la misma dirección del texto.
- **row-reverse**: Elementos son colocados en la dirección opuesta al texto.
- **column**: Elementos se colocan de arriba hacia abajo.
- **column-reverse**: Elementos se colocan de abajo hacia arriba.

```
1 #pond {
2   display: flex;
3   flex-direction: row-reverse;
4 }
5
```



**FLEXBOX FROGGY** Nivel 9 de 24

Ayuda a las ranas a encontrar su hoja de lirio en la columna usando **flex-direction**. Esta propiedad CSS define la dirección de los elementos en el contenedor, y acepta los siguientes valores:

- **row**: Elementos son colocados en la misma dirección del texto.
- **row-reverse**: Elementos son colocados en la dirección opuesta al texto.
- **column**: Elementos se colocan de arriba hacia abajo.
- **column-reverse**: Elementos se colocan de abajo hacia arriba.

```
1 #pond {
2   display: flex;
3   flex-direction: column;
4 }
5
```



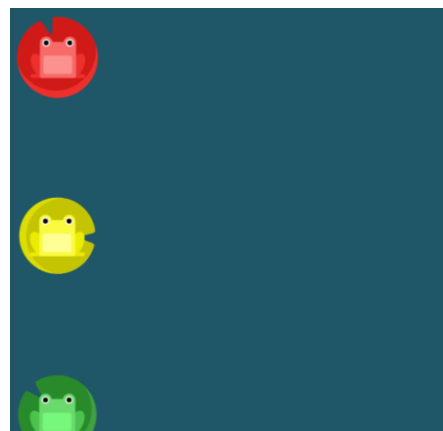
Alineación de dirección (flex-direction) y horizontal (justify-content) con Flexbox.

### FLEXBOX FROGGY

Nivel 12 de 24

Ayuda a las ranas a encontrar sus hojas de lirio usando `flex-direction` y `justify-content`.

```
1 #pond {  
2   display: flex;  
3   flex-direction: column-reverse;  
4   justify-content: space-between;  
5 }
```



### FLEXBOX FROGGY

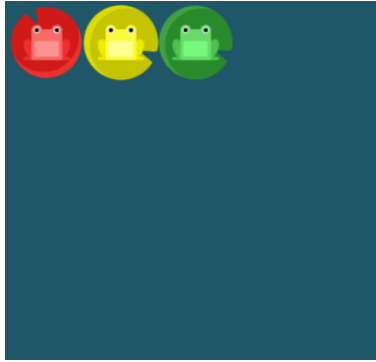
Nivel 10 de 24

Ayuda a las ranas a llegar a sus hojas de lirio. A pesar de que ellas parecen estar cerca, necesitarás usar `flex-direction` y `justify-content` para llegar hasta allí.

Nota que cuando estableces la dirección a una fila invertida o columna, el inicio y el final también se invierten.

```
1 #pond {  
2   display: flex;  
3   flex-direction: row-reverse;  
4   justify-content: flex-end;  
5 }
```





## Order – Cambiar el Orden de un Solo Elemento

Por default los elementos en HTML se colocan de izquierda a derecha y de arriba hacia abajo uno tras otro dependiendo del orden en el que se hayan creado las etiquetas dentro del código, pero la propiedad `order` de flexbox se puede utilizar para cambiar el orden en el que se muestran los elementos en una página web sin cambiar su posición en el código HTML. Al asignar un número a la propiedad `order` de un elemento, se le está indicando al navegador que ese elemento debe aparecer antes o después de los demás, ya que por default todos los elementos tienen un `order` de 0. Los elementos con un número de `order` más bajo se mostrarán primero, y los elementos con un número de `order` más alto se mostrarán después.

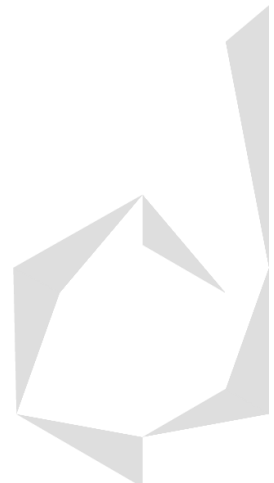
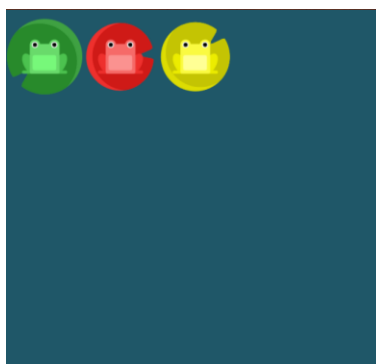
Por ejemplo, si se tiene tres elementos flexibles y se les asignan los números de `order` 2, 1 y 3 respectivamente, el elemento con `order` 1 aparecerá primero, luego el elemento con `order` 2 y, por último, el elemento con `order` 3.

**FLEXBOX FROGGY** Nivel 14 de 24

A veces, invertir el orden de una fila o columna de un contenedor no es suficiente. En esos casos, nosotros podemos aplicar la propiedad `order` a elementos individuales. Por defecto, los elementos tienen un valor 0, pero nosotros podemos usar esta propiedad para establecerlo a un número entero positivo o negativo.

Usa la propiedad `order` para reordenar las ranas de acuerdo a sus hojas de lirio.

```
1 #pond {
2   display: flex;
3 }
4
5 .yellow {
6   order:1
7 }
```



En el siguiente ejemplo se crea una clase y cada uno de sus elementos están en posición 0 inicialmente, si se le asigna un `order 1` (no importando el número que se le ponga el chiste es que sea positivo) se moverá un lugar hacia la derecha y el elemento que estaba a la derecha ocupará su lugar, lo mismo pasa si se asigna un `order -1` pero se moverá hacia la izquierda (no importando el número que se le ponga el chiste es que sea negativo), todos los elementos (divs) a los que le aplique mi clase se irán hacia la derecha (positivos) o hacia la izquierda (negativos). Pero esto solo aplica si se busca mover simplemente 1 elemento de varios.

```
FLEXBOX FROGGY
Nivel 16 de 24

Otra propiedad que puedes aplicar a elementos individuales es align-self. Esta propiedad acepta los mismos valores de align-items y sus valores son usados para un elemento específico

1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   align-self: flex-end;
8 }
```



## Align Self – Alineación Vertical de un Solo Elemento

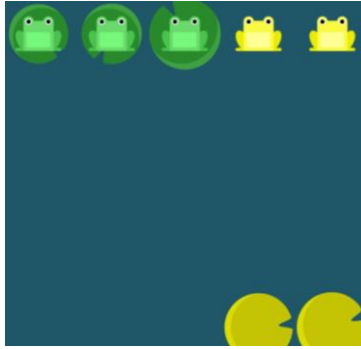
En vez de alinear verticalmente todos los elementos pertenecientes a la fila de un contenedor como se hace con la instrucción `align-items`, se puede utilizar la instrucción `align-self` de Flexbox cuando se busca acomodar de forma diferente un solo elemento de forma vertical, anulando la instrucción previamente declarada con la instrucción `align-items`. Es posible combinar su uso con la etiqueta `order`.

```
FLEXBOX FROGGY
Nivel 17 de 24

Combina order con align-self para ayudar a las ranas a llegar a sus destinos.

1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   order:1;
8   align-items: flex-end;
9 }
10
```

Siguiente



**FLEXBOX FROGGY** Nivel 17 de 24

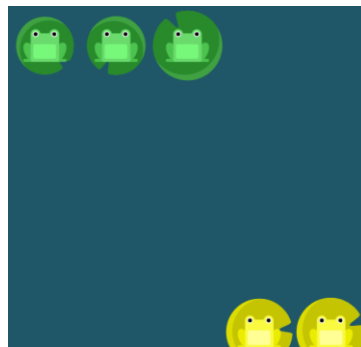
Combina `order` con `align-self` para ayudar a las ranas a llegar a sus destinos.

```

1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   order:1;
8   align-self: flex-end;
9 }
10

```

Siguiente



## Flex Wrap – Distribución de Elementos en una o varias Filas

Indica si los elementos se deben apretar en una sola fila, probablemente deformando su contenido o si pueden abarcar más de una.

**FLEXBOX FROGGY** Nivel 18 de 24

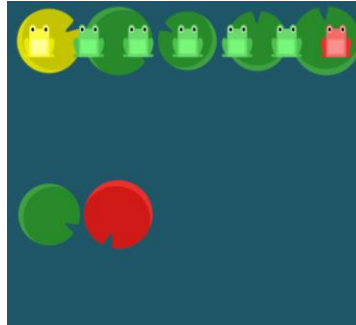
¡Oh, no! Todas las ranas están apretadas en una sola fila de hojas de lirio. Manos a la obra, vamos a distribuir las correctamente usando la propiedad `flex-wrap`, la cual acepta los siguientes valores:

- `nowrap`: Cada elemento se ajusta en una sola línea.
- `wrap`: los elementos se envuelven alrededor de líneas adicionales.
- `wrap-reverse`: Los elementos se envuelven alrededor de líneas adicionales en reversa.

```

1 #pond {
2   display: flex;
3   flex-wrap: nowrap;
4 }

```



## FLEXBOX FROGGY

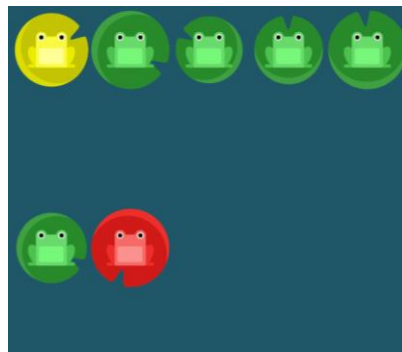
Nivel 18 de 24

¡Oh, no! Todas las ranas están apretadas en una sola fila de hojas de lirio. Manos a la obra, vamos a distribuir las ranas correctamente usando la propiedad **flex-wrap**, la cual acepta los siguientes valores:

- **nowrap**: Cada elemento se ajusta en una sola línea.
- **wrap**: los elementos se envuelven alrededor de líneas adicionales.
- **wrap-reverse**: Los elementos se envuelven alrededor de líneas adicionales en reversa.

```

1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4 }
  
```



Distribución de elementos (flex-wrap) y alineación en dirección de columna (flex-direction) con Flexbox.

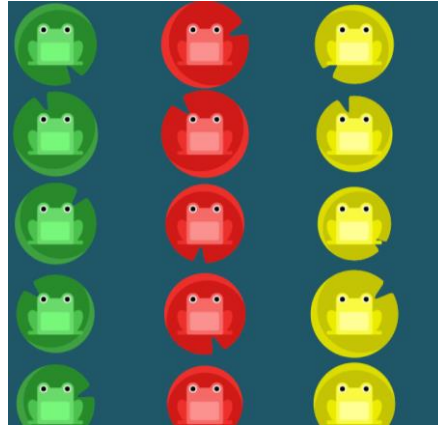
## FLEXBOX FROGGY

Nivel 19 de 24

Ayuda a este ejército de ranas a ordenarse en tres columnas usando una combinación de **flex-direction** y **flex-wrap**.

```

1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   flex-direction: column;
5 }
6
  
```



## FLEXBOX FROGGY

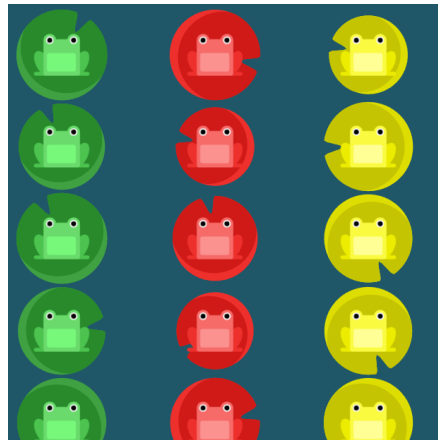
Nivel 20 de 24

Las dos propiedades `flex-direction` y `flex-wrap` son usadas a menudo en conjunto con la propiedad abreviada `flex-flow`, la cual fue creada para combinarlas. Esta propiedad abreviada, acepta un valor de cada una separada por un espacio.

Por ejemplo, puedes usar `flex-flow` para establecer filas y envolverlas.

Trata de usar `flex-flow` para volver a realizar el nivel anterior.

```
1 #pond {  
2   display: flex;  
3   flex-flow: wrap column;  
4 }  
5
```



## Align Content – Distribución o Separación de Vertical

La instrucción `align-content` sirve para distribuir verticalmente varias filas de elementos en vez de distribuir verticalmente los elementos contenidos dentro de una sola fila, como se realiza con la instrucción `align-items`, a continuación, se enlistará las funciones de cada una para denotar sus diferencias.

- **align-items:** La propiedad align-items determina cómo se alinean verticalmente los elementos que se encuentran en la fila de un contenedor. Ya sea que se alineen en la parte superior del contenedor, en su parte inferior, en el centro o incluso estirarlos para que ocupen todo el espacio vertical.
- **align-content:** La propiedad align-content controla cómo se distribuyen y alinean varias filas verticalmente en un contenedor. Ya sea que todas se alineen en la parte superior, en la parte inferior, en el centro, con espacios entre ellas o incluso estirándolas para llenar todo el espacio vertical del contenedor.

## FLEXBOX FROGGY

Nivel 21 de 24

Las ranas están repartidas por todo el estanque, pero las hojas de lirio están agrupadas en la parte superior. Puedes usar **align-content** para establecer como múltiples líneas están separadas una de otra. Esta propiedad acepta los siguientes valores:

- **flex-start:** Las líneas se posicionan en la parte superior del contenedor.
- **flex-end:** Las líneas se posicionan en la parte inferior del contenedor.
- **center:** Las líneas se posicionan en el centro (verticalmente hablando) del contenedor.
- **space-between:** Las líneas se muestran con la misma distancia entre ellas.
- **space-around:** Las líneas se muestran con la misma separación alrededor de ellas.
- **stretch:** Las líneas se estiran para ajustarse al contenedor.

Esto puede ser confuso, pero **align-content** determina el espacio entre las líneas, mientras que **align-items** determina como los elementos en su conjunto están alineados dentro del contenedor. Cuando hay solo una línea, **align-content** no tiene efecto.

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   |
5 }
6
```



## FLEXBOX FROGGY

Nivel 21 de 24

Las ranas están repartidas por todo el estanque, pero las hojas de lirio están agrupadas en la parte superior. Puedes usar `align-content` para establecer como múltiples líneas están separadas una de otra. Esta propiedad acepta los siguientes valores:

- `flex-start`: Las líneas se posicionan en la parte superior del contenedor.
- `flex-end`: Las líneas se posicionan en la parte inferior del contenedor.
- `center`: Las líneas se posicionan en el centro (verticalmente hablando) del contenedor.
- `space-between`: Las líneas se muestran con la misma distancia entre ellas.
- `space-around`: Las líneas se muestran con la misma separación alrededor de ellas.
- `stretch`: Las líneas se estiran para ajustarse al contenedor.

Esto puede ser confuso, pero `align-content` determina el espacio entre las líneas, mientras que `align-items` determina como los elementos en su conjunto están alineados dentro del contenedor. Cuando hay solo una línea, `align-content` no tiene efecto.

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   align-content: flex-start;  
5 }
```

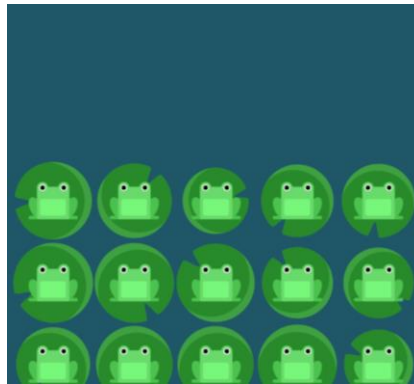


## FLEXBOX FROGGY

Nivel 22 de 24

Ahora, la corriente ha agrupado las hojas de lirio en la parte inferior. Usa `align-content` para guiar las ranas hasta allí.

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   align-content: flex-end;  
5 }  
6
```

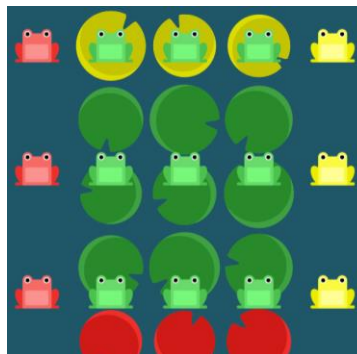


## FLEXBOX FROGGY

Nivel 23 de 24

Las ranas han tenido una fiesta, pero es momento de volver a casa. Usa una combinación de `flex-direction` y `align-content` para llevarlas a sus hojas de lirio.

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4
5
6 }
```



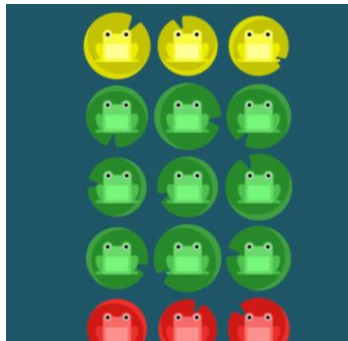
## FLEXBOX FROGGY

Nivel 23 de 24

Las ranas han tenido una fiesta, pero es momento de volver a casa. Usa una combinación de `flex-direction` y `align-content` para llevarlas a sus hojas de lirio.

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   flex-direction: column-reverse;
5   align-content: center;
6 }
```





## Resumen de Flexbox

### FLEXBOX FROGGY

◀ Nivel 24 de 24 ▶

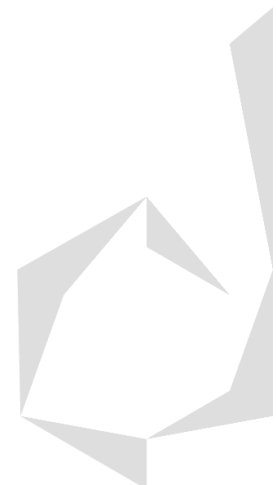
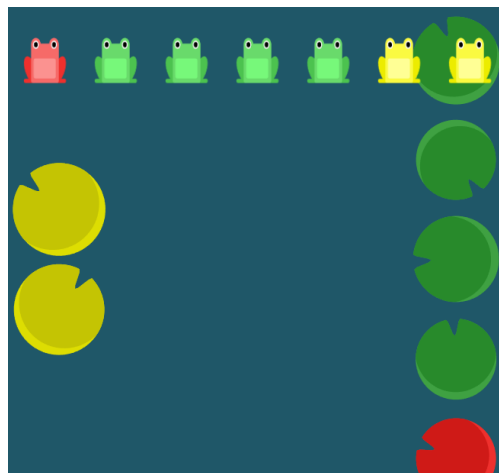
Trae a las ranas a su casa una vez más usando las propiedades CSS que aprendiste:

- `justify-content`
- `align-items`
- `flex-direction`
- `order`
- `align-self`
- `flex-wrap`
- `flex-flow`
- `align-content`

```

1 #pond {
2   display: flex;
3   |
4   |
5   |
6   |
7 }

```



## Justify Content – Alineación Horizontal de Elementos

- `justify-content`
- #7
- Alinea los elementos flex a lo largo del eje principal.
- `flex-start` `flex-end` `center`
- `space-between` `space-around`
- 

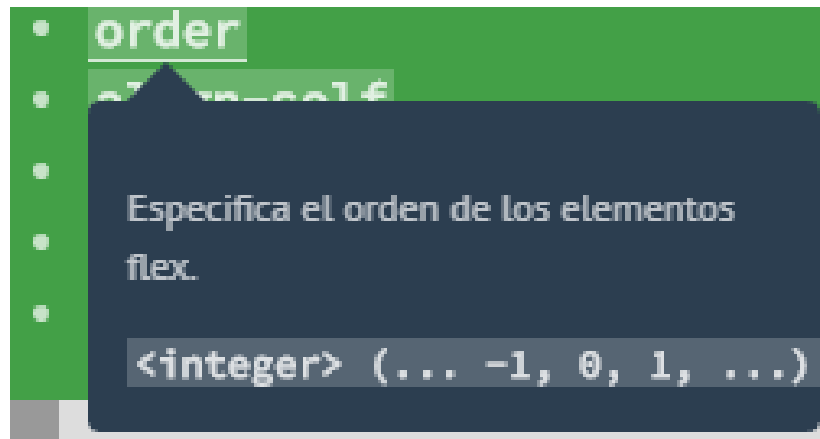
## Align Items – Alineación Vertical de Elementos

- `align-items`
- #7
- Alinea elementos Flex a lo largo del eje transversal.
- `flex-start` `flex-end` `center`
- `baseline` `stretch`
- 

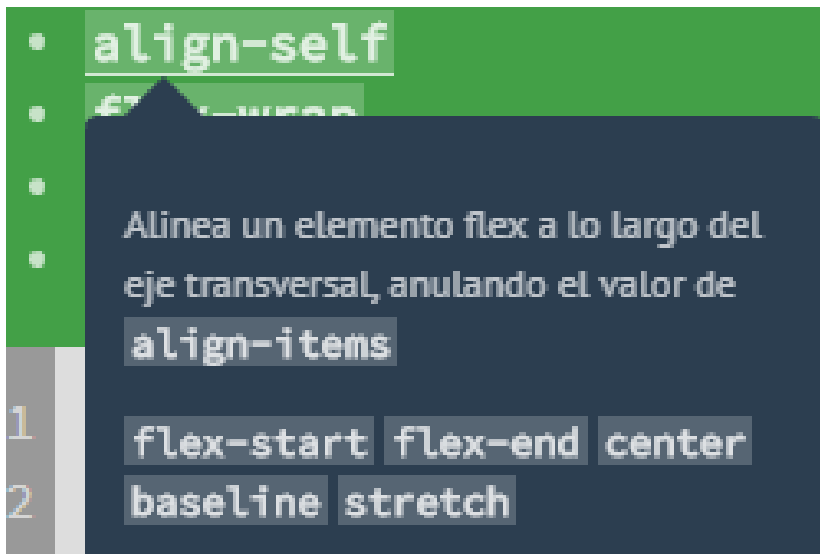
## Flex Direction – Dirección de Elementos

- `flex-direction`
- #7
- Define la dirección del eje principal.
- `row` `row-reverse` `column`
- `column-reverse`
-

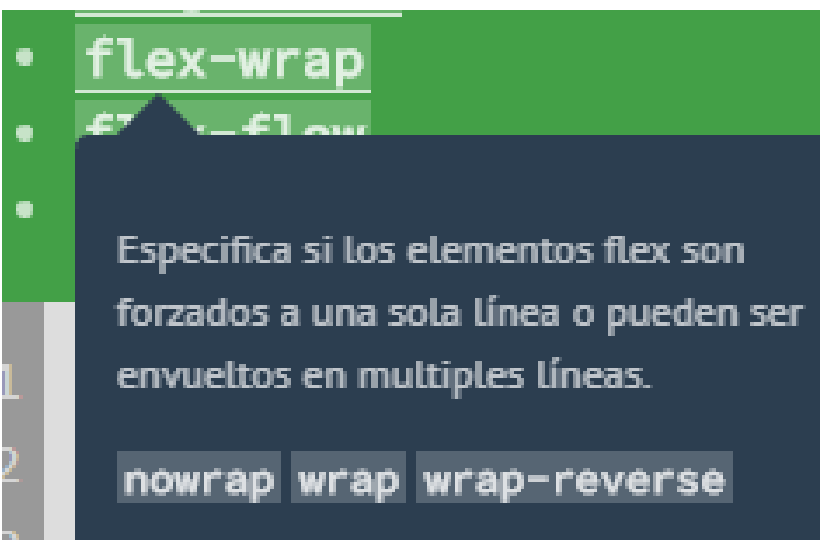
## Order – Cambiar el Orden de un Solo Elemento



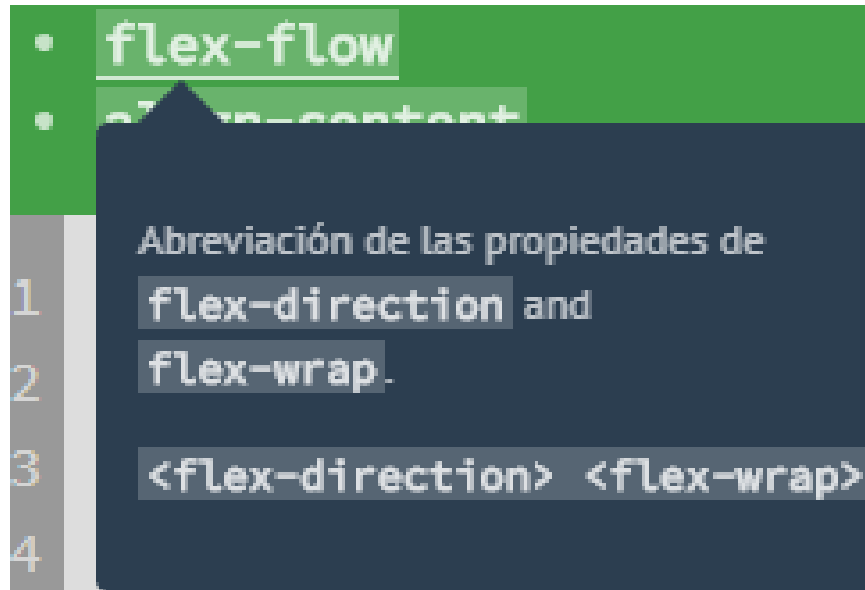
## Align Self – Alineación Vertical de un Solo Elemento



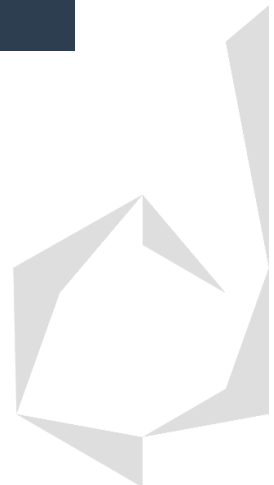
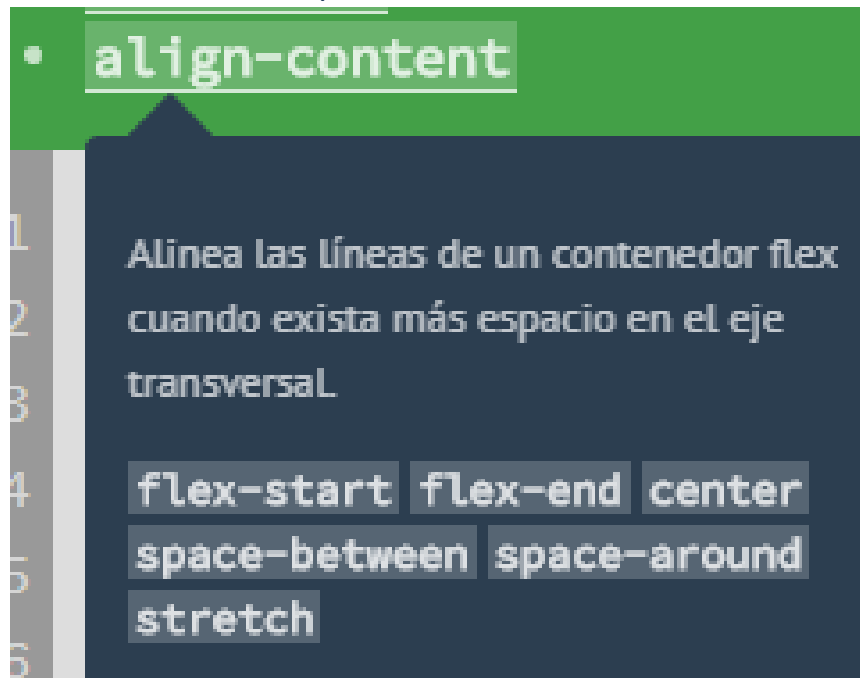
## Flex Wrap – Distribución de Elementos en una o varias Filas



## Flex Flow – Combinación de flex-direction y flex-wrap



## Align Content – Distribución o Separación de Vertical



# FLEXBOX FROGGY

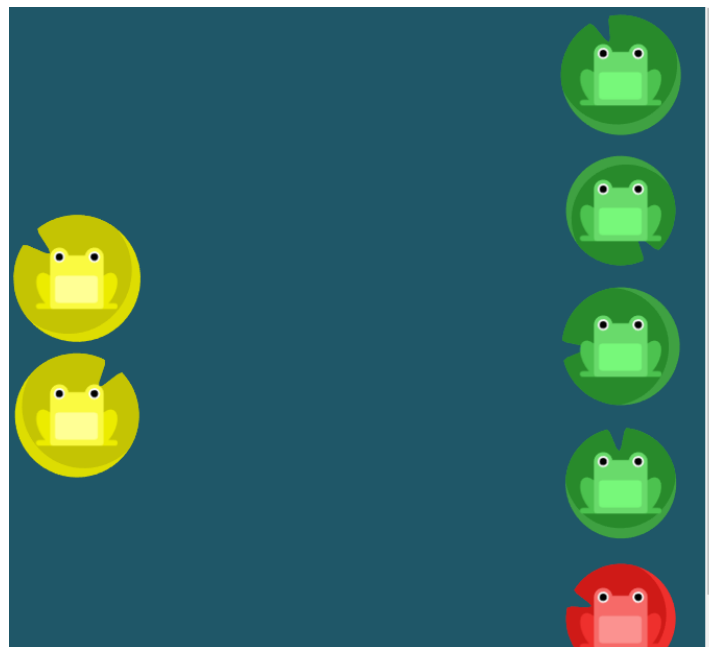
Nivel 24 de 24

Trae a las ranas a su casa una vez más usando las propiedades CSS que aprendiste:

- `justify-content`
- `align-items`
- `flex-direction`
- `order`
- `align-self`
- `flex-wrap`
- `flex-flow`
- `align-content`

```
1 #pond {  
2   display: flex;  
3   flex-flow: wrap-reverse column-reverse;  
4   justify-content: center;  
5   align-items: flex-end;  
6   align-content: space-between;  
7 }  
8  
9  
10
```

Siguiente



# Referencias:

Codepip, GitHub y Twitter, "FLEXBOX FROGGY", [Online], Available: <https://flexboxfroggy.com/#es>

