



Instituto Nacional de México

Instituto Tecnológico de Tijuana

Departamento de sistemas y computación

Ingeniería informática

Práctica 1:
Tutorial de Flexbox Froggy

Materia:
Desarrollo de aplicaciones

Periodo: Verano de Ago-Dic 2020-2

Unidad: 2

Facilitador:
Dra. Daniela A. Sánchez Vizcarra
Alumno:
Luis Eduardo Fierro López

Grupo:

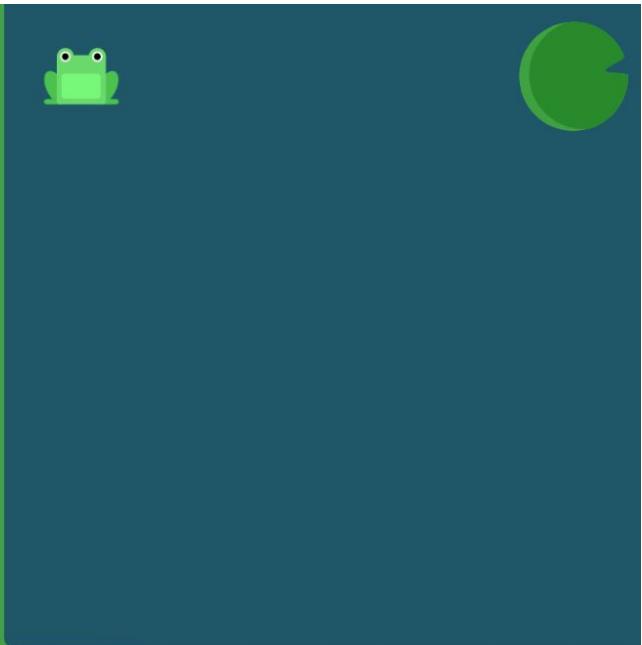
1

Fecha:

Tijuana Baja California a 17 de Octubre del 2020

Actividad 1. Realizar el tutorial de Flexbox Froggy

Realizar el tutorial de Flexbox Froggy completando un total de 24 niveles escribiendo código CSS. El tutorial se realiza de manera didáctica cambiando la posición y otros atributos de una ranita en la parte de la derecha de la página.



The screenshot shows the first level of the Flexbox Froggy game. On the left, there's a green frog icon. On the right, there's a large circular pond. At the top left, it says "FLEXBOX FROGGY". In the center, it says "Nivel 1 de 24". Below that, there's some text about the game and a list of justify-content values. A code editor window shows a snippet of CSS:

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

At the bottom right of the code editor is a red "Siguiente" button. At the very bottom of the screen, there's a footer with links to Codeipip, GitHub, Twitter, and Settings.

Códigos utilizados

justify-content: Alinea los elementos horizontalmente.

- **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.

align-items: Alinea los elementos verticalmente.

- **flex-start:** Alinea elementos en 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.

flex-direction: Esta propiedad CSS define la dirección de los elementos en el contenedor, y acepta los siguientes valores.

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

Usa la propiedad **order** para reordenar.

align-self es para valores específicos, acepta los mismos valores que **align-items**.

flex-wrap, propiedad para distribuir.

- **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.

align-content: para establecer como múltiples líneas están separadas una de otra

- **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.

Nivel 1

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:

Para este nivel utilizamos el siguiente código:

justify-content: flex-end;

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

[Siguiente](#)

Después de colocar esa línea de código la rana cambiará de su posición inicial tomando como nueva posición el final de la línea colocando en un nenúfar.

Antes:



Después:



Nivel 2

Nuevamente, utiliza **justify-content** para ayudar a esas ranas a llegar a sus hojas de lirio. Recuerda que esta propiedad CSS alinea elementos horizontalmente y acepta los siguientes valores:

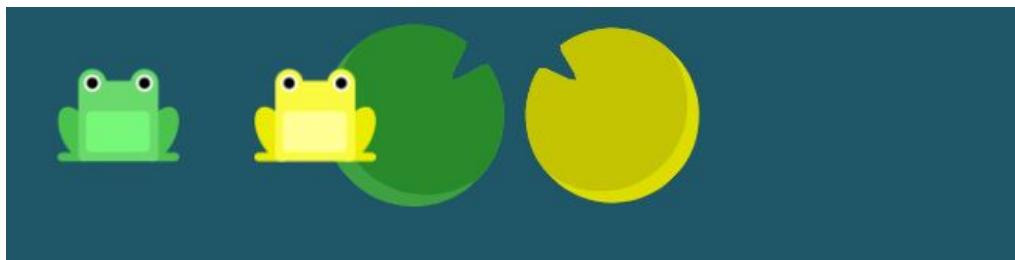
Pare este nivel utilizamos el siguiente código:

justify-content: center;

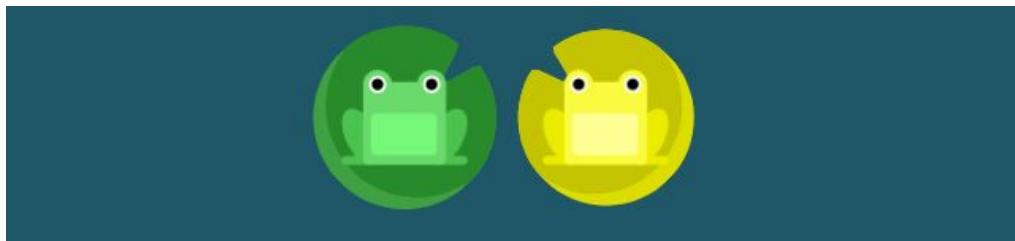
```
1 #pond {  
2   display: flex;  
3   justify-content: center;  
4 }  
5  
6  
7  
8  
9  
10
```

[Siguiente](#)

Este código hará que las ranitas se centren en la página quedando de la siguiente manera.
Antes:



Después:



Nivel 3

Ayuda a las tres ranas a encontrar sus hojas de lirio usando **justify-content**. En esta oportunidad, las hojas de lirio tienen mucho espacio a su alrededor.

Si no recuerdas los valores de una propiedad, puedes pasar el cursor sobre el nombre de una e inmediatamente se mostrarán. Prueba pasando el cursor sobre **justify-content**.

Código utilizado:

justify-content: space-around;

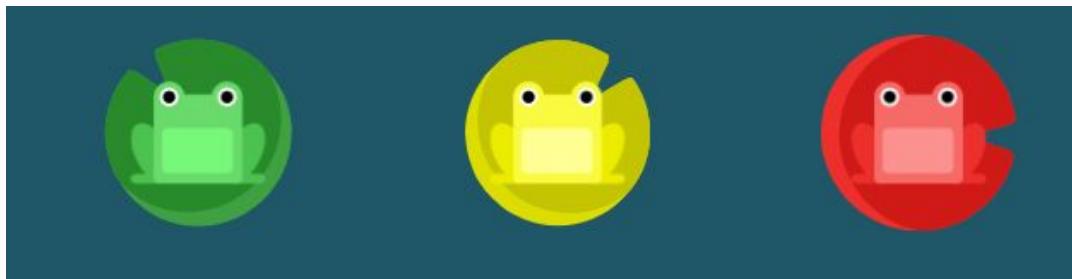
```
1 #pond {  
2   display: flex;  
3   justify-content: space-around;  
4 }  
5  
6  
7  
8  
9  
10
```

[Siguiente](#)

El código anterior ayudará a las tres ranitas a sentarse en sus nenúfares de su mismo color.
Antes:



Después:



Nivel 4

Ahora las hojas de lirio de los bordes se han desplazado a un costado, aumentando así el espacio entre ellas. Usa **justify-content**. En esta oportunidad, las hojas de lirio tienen el mismo espacio entre ellas.

Código utilizado:

justify-content: space-between;

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

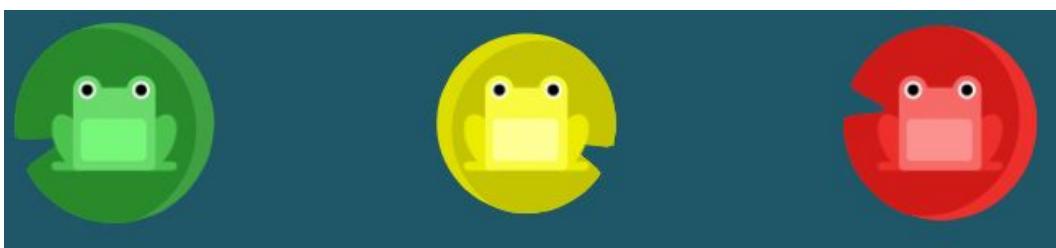
[Siguiente](#)

Similar al anterior pero una en la izquierda, otra en el centro y la última en la derecha.

Antes:



Después:



Nivel 5

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:

Código utilizado:

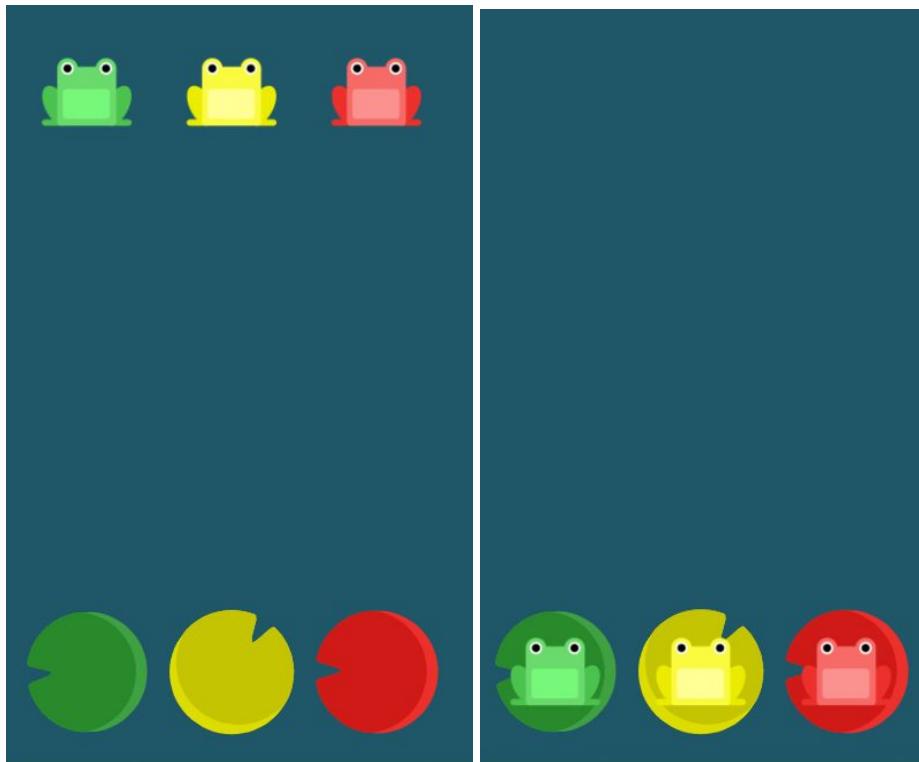
```
align-items: flex-end;
```

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

[Siguiente](#)

Ahora van para abajo.

Antes y después:



Nivel 6

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

Código utilizado:

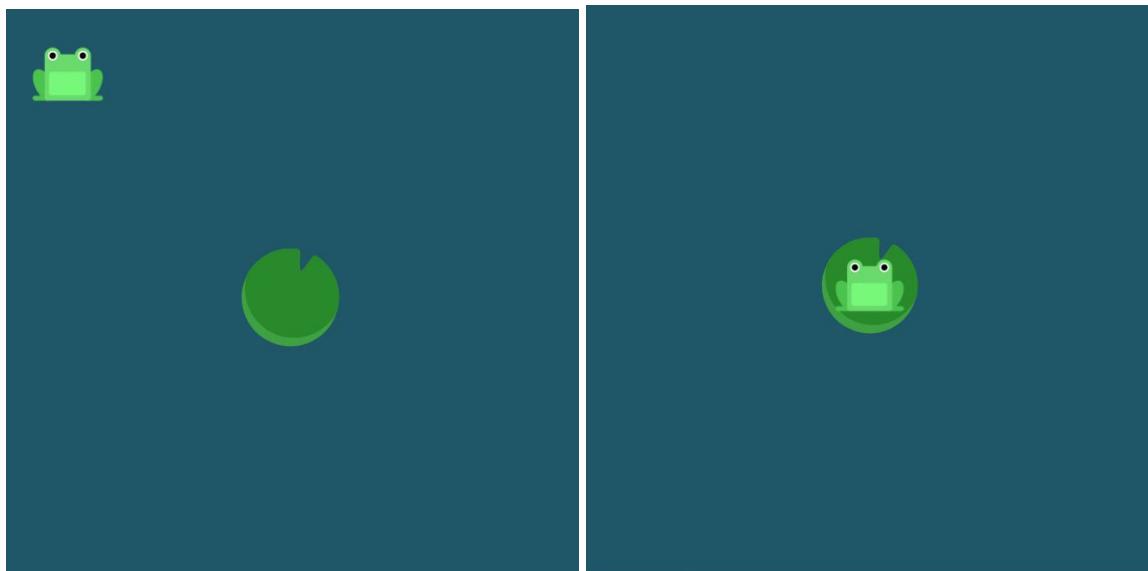
```
justify-content: center;  
align-items: center;
```

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

[Siguiente](#)

Ahora se acomoda en el centro.

Antes y después:



Nivel 7

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**.

Código utilizado:

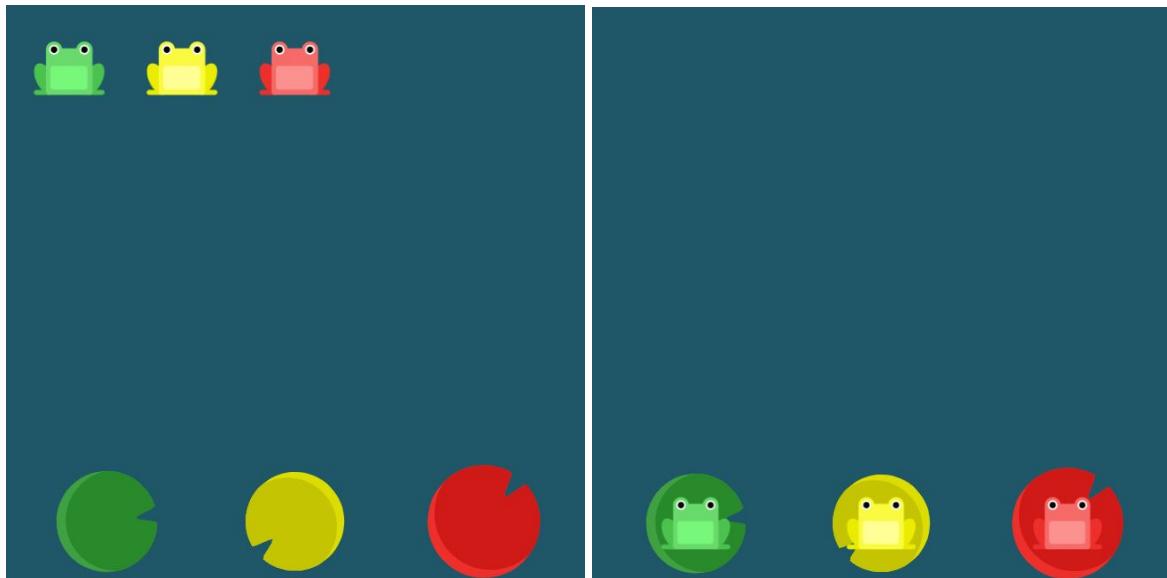
```
justify-content: space-around;  
align-items: flex-end;
```

```
1 #pond {  
2   display: flex;  
3   justify-content: space-around;  
4   align-items: flex-end;  
5 }  
6  
7  
8  
9  
10
```

[Siguiente](#)

Se bajan y se separan.

Antes y después:



Nivel 8

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:

Código utilizado:

```
flex-direction: row-reverse;
```

```
1 #pond {  
2   display: flex;  
3   flex-direction: row-reverse;  
4 }  
5  
6  
7  
8  
9  
10
```

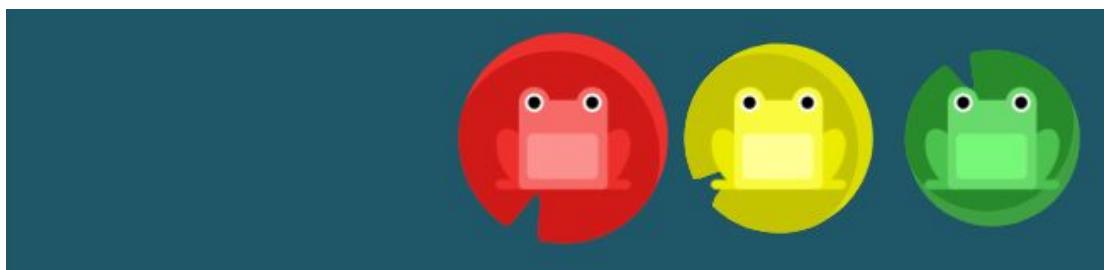
[Siguiente](#)

Como podemos observar las ranitas invierten horizontalmente.

Antes:



Después:



Nivel 9

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:

Código utilizado:

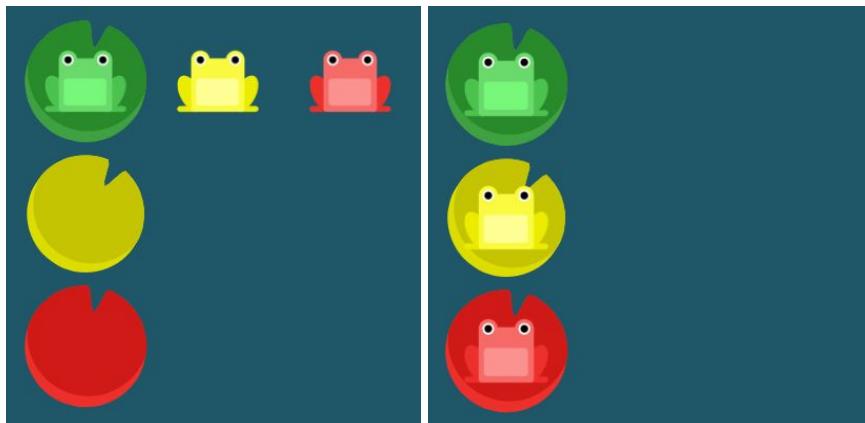
```
flex-direction: column;
```

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

[Siguiente](#)

Parece como si las rotaramos hacia abajo.

Antes y después



Nivel 10

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.

Código utilizado:

```
flex-direction: row-reverse;  
justify-content: flex-end;
```

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

[Siguiente](#)

Ahora los invertimos doble para que encajen.

Antes y después



Nivel 11

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

Nota que cuando es una columna, **justify-content** cambia a vertical y **align-items** a horizontal.

Código utilizado:

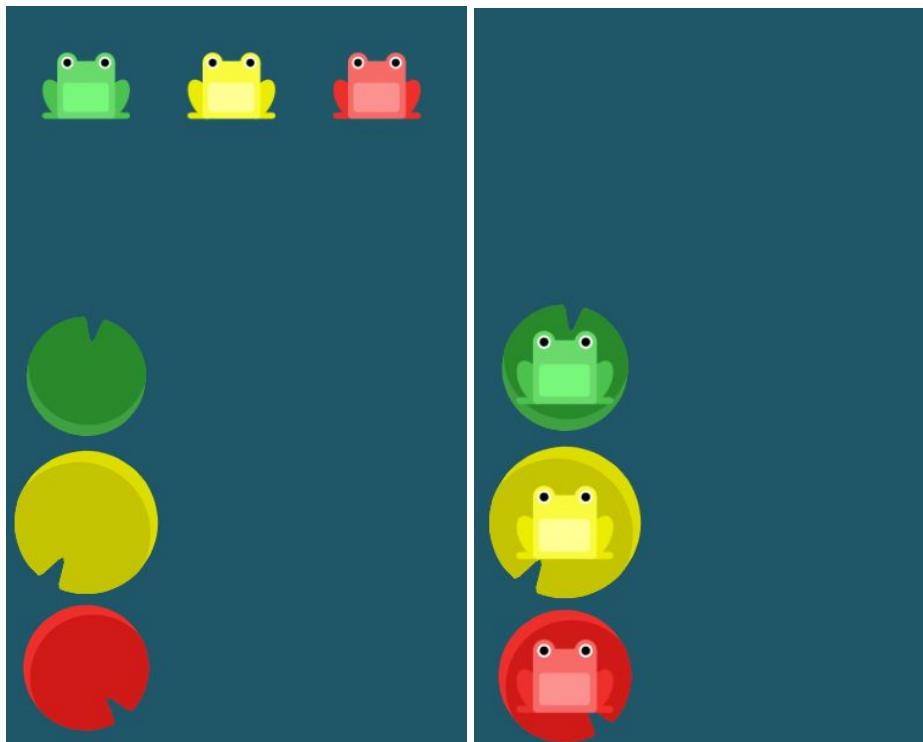
```
flex-direction: column;  
justify-content: flex-end;
```

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

[Siguiente](#)

En este nivel los acomodamos en forma de columna y las bajamos.

Antes y después



Nivel 12

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

Código utilizado:

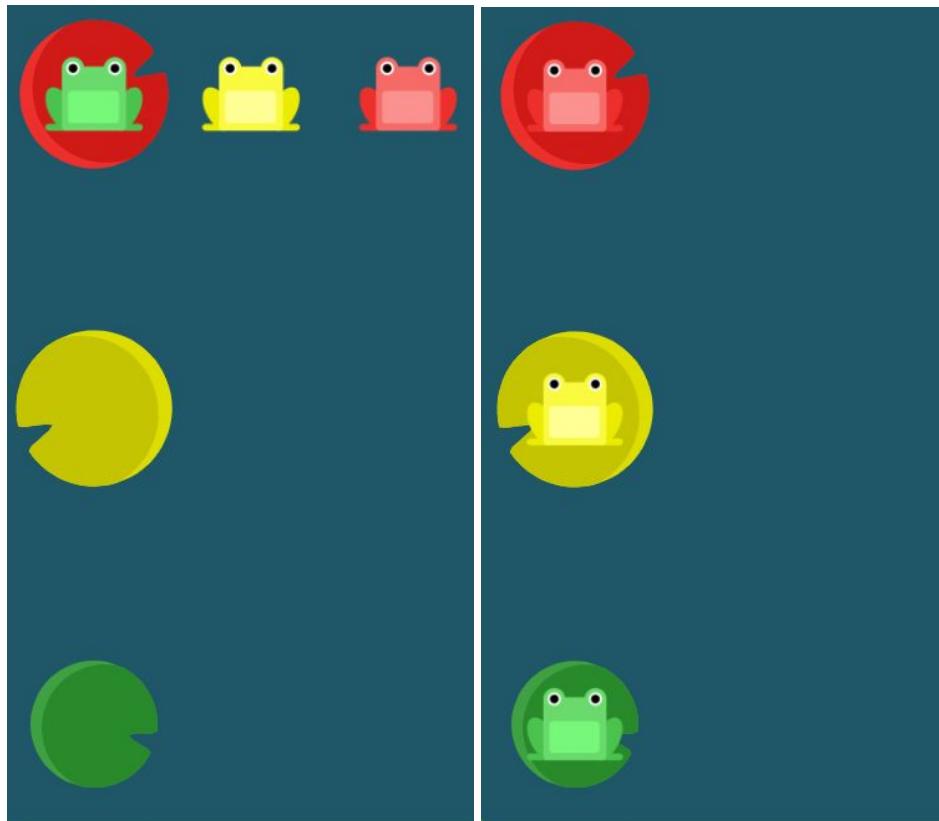
```
flex-direction: column-reverse;  
justify-content: space-between;
```

```
1 #pond {  
2   display: flex;  
3   flex-direction: column-reverse;  
4   justify-content: space-between;  
5 }  
6  
7  
8  
9  
10
```

[Siguiente](#)

Aquí se acomodan en forma de columna invertida y se deja espacio entre ellas.

Antes y después:



Nivel 13

Ayuda a las ranas a encontrar sus hojas de lirio usando **flex-direction**, **justify-content**, y **align-items**.

Código utilizado:

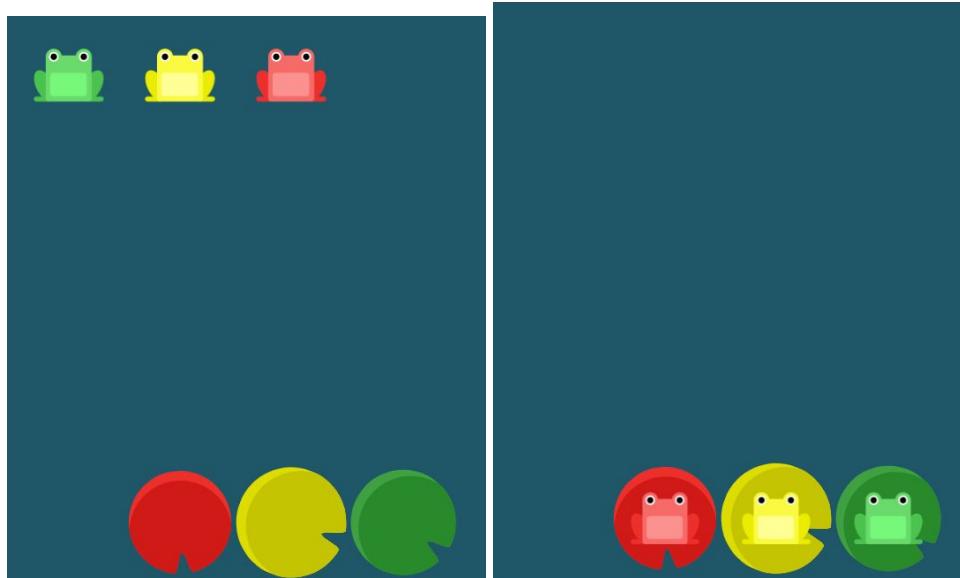
```
flex-direction: row-reverse;  
justify-content: center;  
align-items: flex-end;
```

```
1 #pond {  
2   display: flex;  
3   flex-direction: row-reverse;  
4   justify-content: center;  
5   align-items: flex-end;  
6 }  
7  
8  
9  
10
```

[Siguiente](#)

El nivel de la mala suerte ya se pone difícil pidiendo 3 opciones de formato distintas para invertir, centrar y bajar a las ranitas.

Antes y después:



Nivel 14

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.

Código utilizado:

order: 1

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

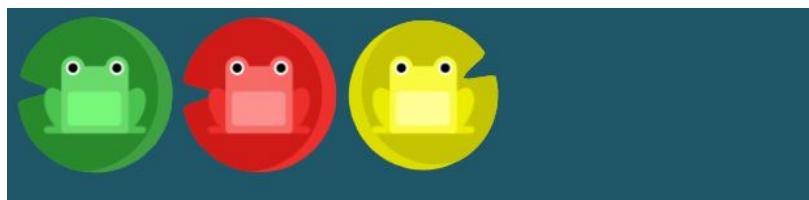
[Siguiente](#)

En esta ocasión las ordenamos con 1 porque quien sabe pero así funciona.

Antes:



Después:



Nivel 15

Usa la propiedad **order** para enviar la rana de color rojo a su hoja de lirio.

Código utilizado:

order: -1

```
1 #pond {  
2   display: flex;  
3 }  
4  
5 .red {  
6   order: -1  
7 }  
8  
9  
10
```

[Siguiente](#)

La verdad provee números al azar y salió con el -1

Antes:



Después:



Nivel 16

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

Código utilizado:

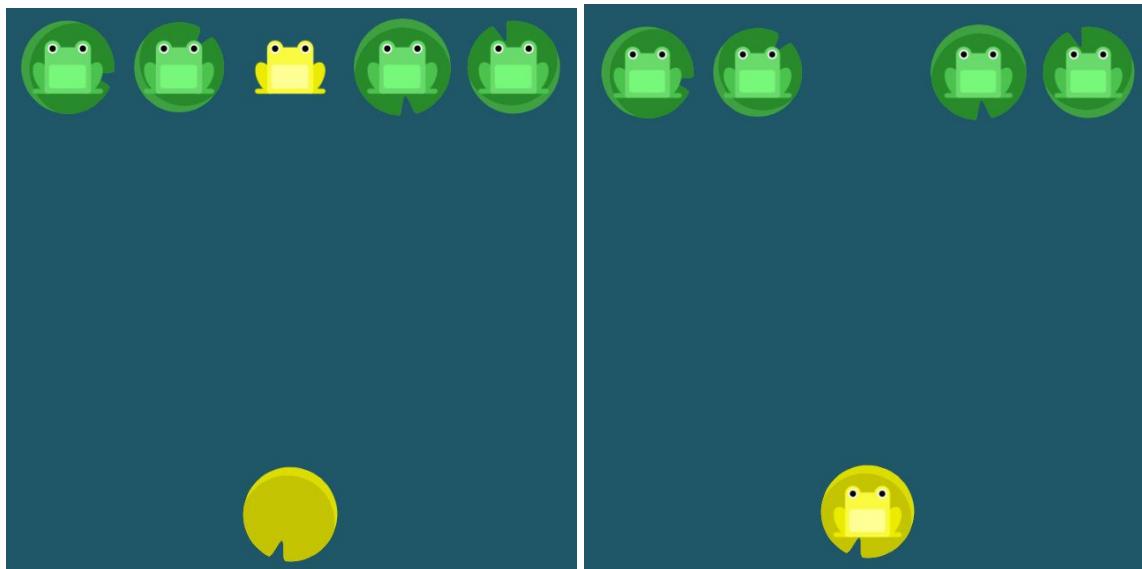
```
align-self: flex-end;
```

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

[Siguiente](#)

Aquí específicamente movemos al amarillo de manera individual.

Antes y después:



Nivel 17

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

Código utilizado:

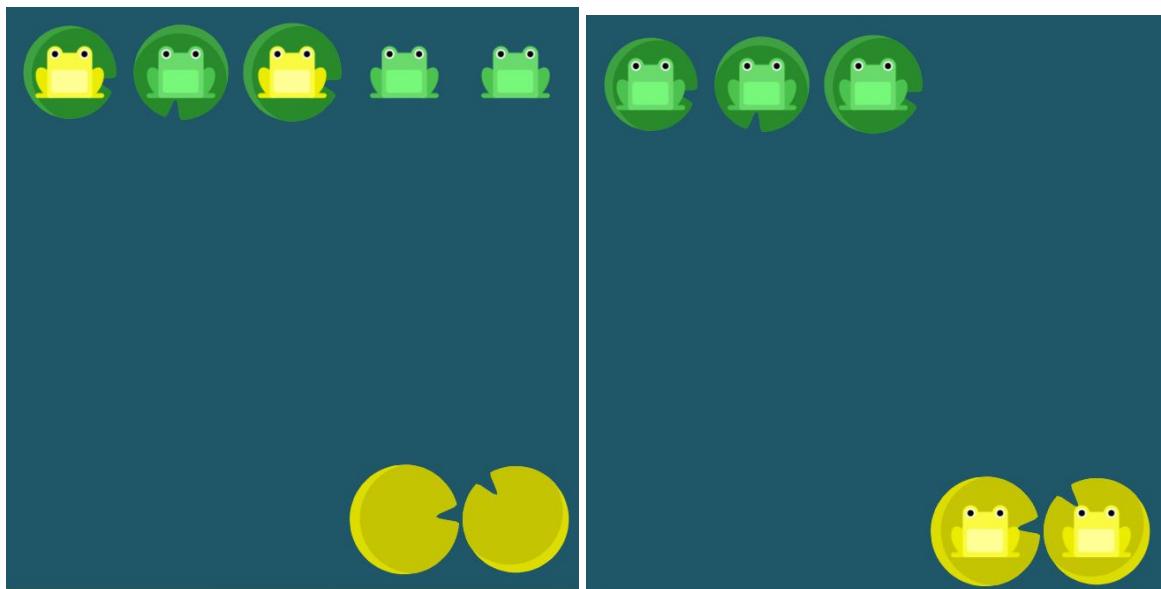
```
order: 1;  
align-self: flex-end;
```

```
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](#)

Al igual que el anterior, solo ordenamos las amarillas y las colocamos hasta abajo.

Antes y después:



Nivel 18

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

Código utilizado:

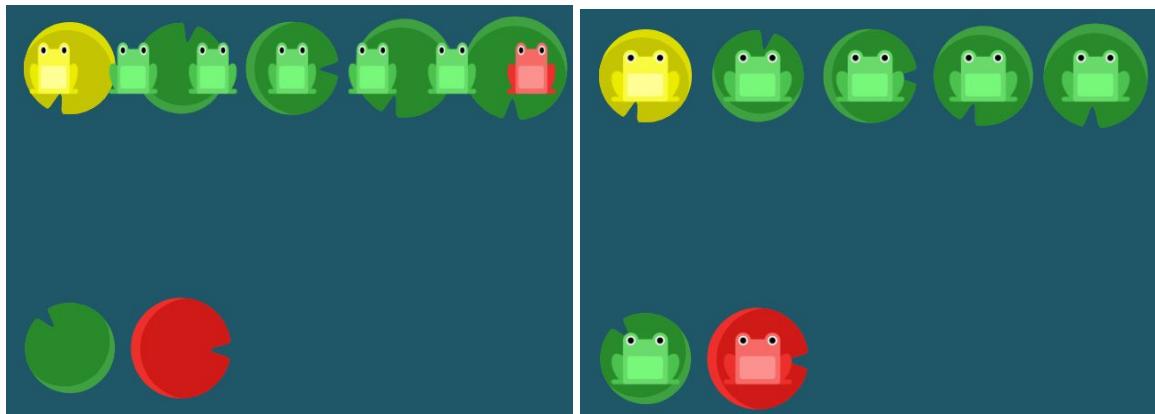
```
flex-wrap: wrap;
```

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

[Siguiente](#)

Ahora las tenemos que distribuir y ordenar.

Antes y después:



Nivel 19

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

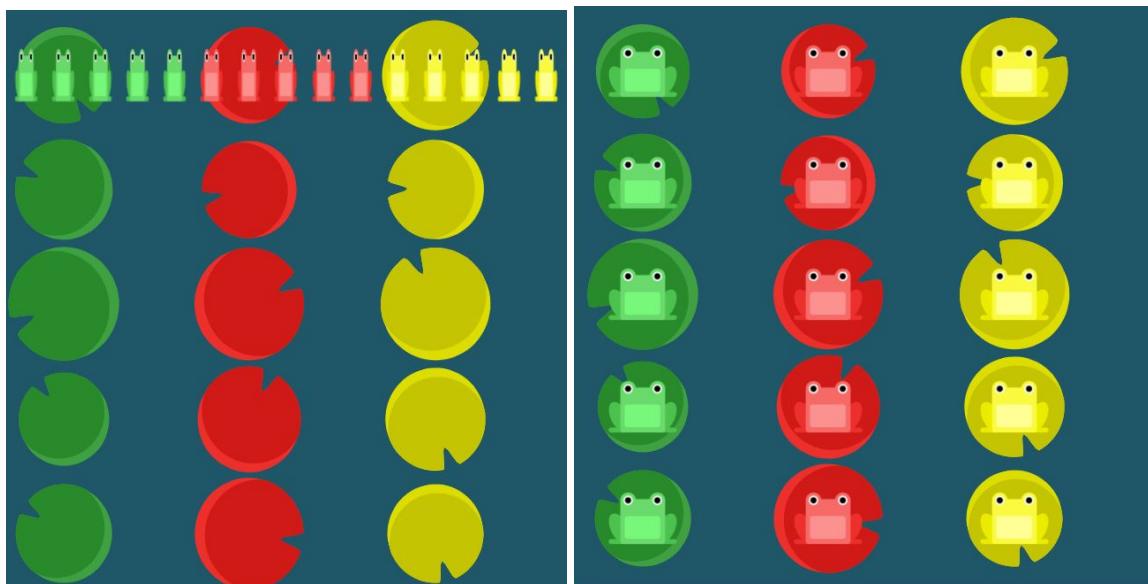
Código utilizado:

```
flex-direction: column;  
flex-wrap: wrap;
```

```
1 #pond {  
2   display: flex;  
3   flex-direction: column;  
4   flex-wrap: wrap;  
5 }  
6  
7  
8  
9  
10
```

[Siguiente](#)

En este nivel tenemos que acomodarlas en forma de columnas y en envoltura o algo así.
Antes y después:



Nivel 20

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.

Código utilizado:

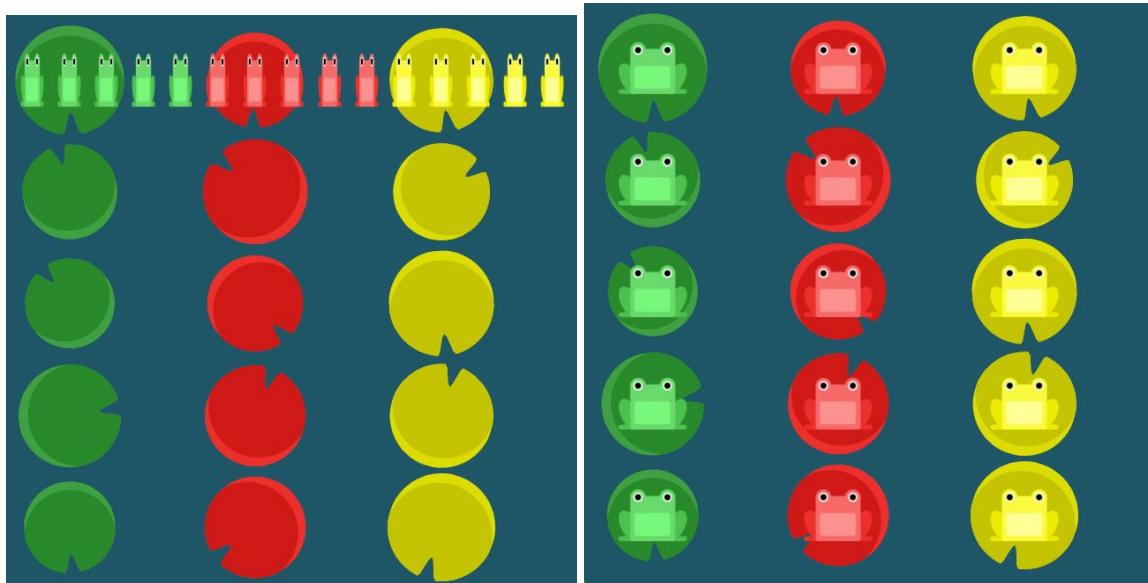
```
flex-flow: column wrap;
```

```
1 #pond {  
2   display: flex;  
3   flex-flow: column wrap;  
4 }  
5  
6  
7  
8  
9  
10
```

[Siguiente](#)

Con un grado mas de dificultad ahora toca combinar propiedades en una sola.

Antes y después:



Nivel 21

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.

Código utilizado:

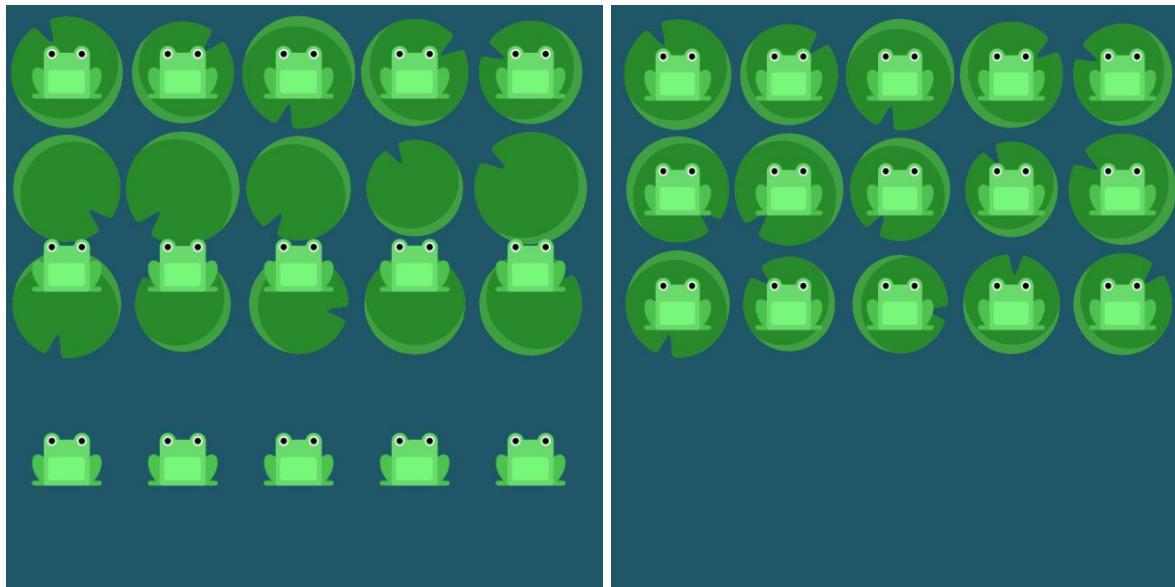
```
align-content: flex-start;
```

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

[Siguiente](#)

Toca aprender otro align algo confuso pero esta vez se usa para establecer múltiples líneas para que queden ordenadas las ranitas bien bonitas

Antes y después:



Nivel 22

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

Código utilizado:

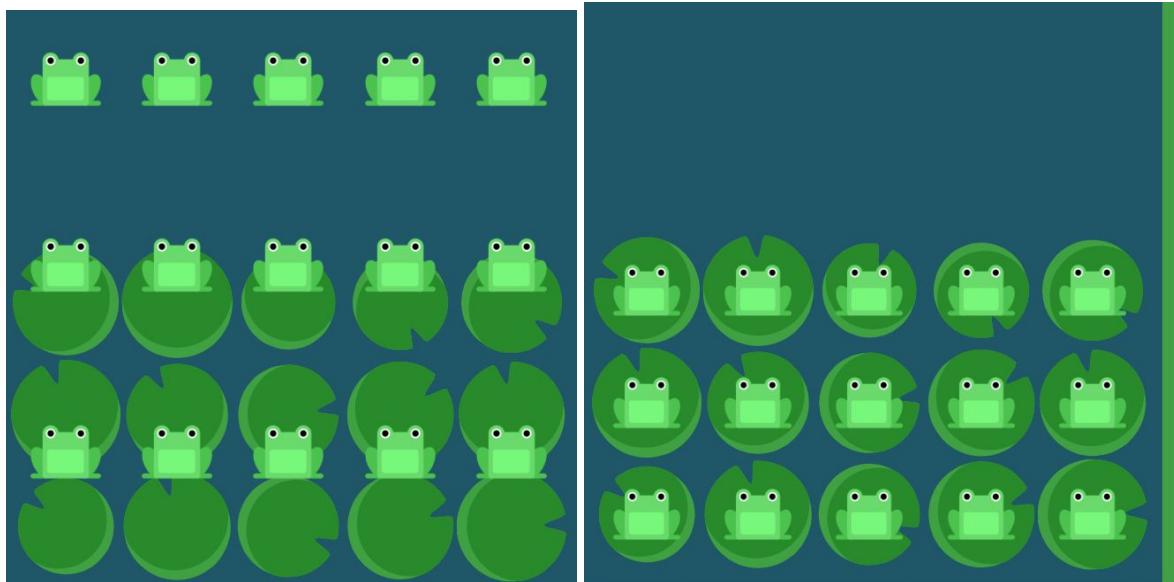
```
align-content: flex-end;
```

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

[Siguiente](#)

Estas es igual que la anterior pero más sencilla, solo las bajamos y se ordenan bien bonitas.

Antes y después:



Nivel 23

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.

Código utilizado:

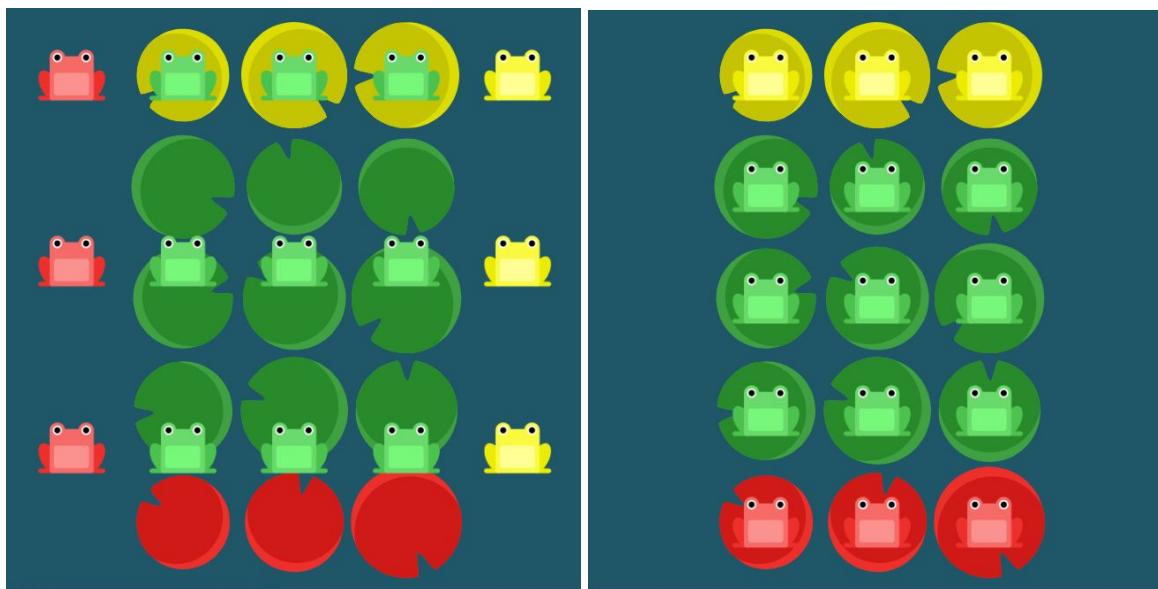
```
flex-direction: column-reverse;  
align-content: center;
```

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   flex-direction: column-reverse;  
5   align-content: center;  
6 }  
7  
8  
9  
10
```

[Siguiente](#)

Facilito, sólo se ocupa de acomodarlas en forma de columna en reversa y después centrar, ambas cosas ya se han hecho en niveles anteriores.

Antes y después:



Nivel 24

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

Código utilizado:

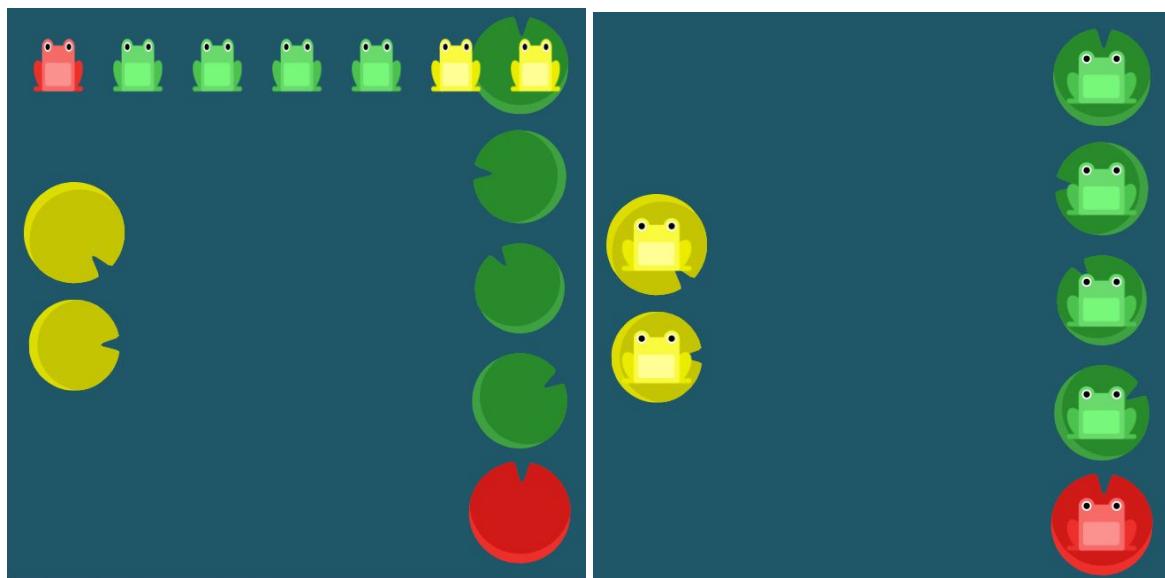
```
flex-flow:column-reverse wrap-reverse;  
justify-content: center;  
align-content: space-between;
```

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

[Siguiente](#)

Para el último nivel toca aplicar todo lo aprendido en los niveles anteriores pero con más dificultad porque te hace pensar. Después de muchos intentos y con un poco de ayuda de google recordé que se pueden combinar las flex-direction y las flex-wrap en uno solo y ya con eso se acomodan bien..

Antes y después:



Finalizado

Facilito
pantallazo final.

