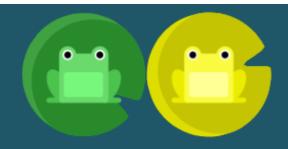


Use <u>justify-content</u> novamente para ajudar essas rãs a chegarem aos nenúfares. Lembre-se de que essa propriedade CSS alinha os itens horizontalmente e aceita os seguintes valores:

- flex-start: Os itens são alinhados à esquerda do contêiner.
- flex-end: Os itens são alinhados ao lado direito do contêiner.
- center: Os itens são alinhados no centro do contêiner.
- space-between: Os itens são exibidos com espaçamento igual entre eles.
- space-around: Os itens são exibidos com espaçamento igual ao redor deles.

```
#Lago {
    display: flex;
    justify-content: center
}

Próximo
```





Ajude os três sapos a encontrar seus nenúfares apenas usando **justify-content**. Desta vez, os nenúfares têm muito espaço ao seu redor.

Se você se esquecer dos valores possíveis para uma propriedade, passe o mouse sobre o nome da propriedade para visualizá-los. Experimente pairar sobre justify-content.

```
#Lago {
    display: flex;
    justify-content: space-around
}

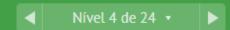
Próximo

#Lago {
    display: flex;
    justify-content: space-around
    }
}
```









Agora, os nenúfares nas bordas foram levados para a costa, aumentando o espaço entre eles. Use **justify-content**. Desta vez, os nenúfares têm espaçamento igual entre eles.







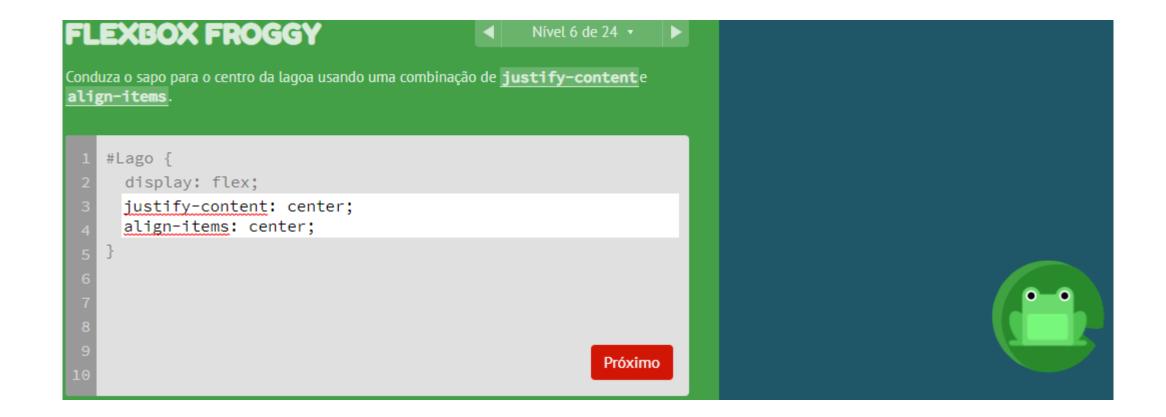
Agora use <u>align-items</u> para ajudar as rãs a chegarem ao fundo do tanque. Esta propriedade CSS alinha os itens verticalmente e aceita os seguintes valores:

- flex-start: Os itens são alinhados ao topo do contêiner.
- flex-end: Os itens são alinhados ao fundo do recipiente.
- center: Os itens são alinhados no centro vertical do contêiner.
- baseline: Os itens são exibidos na linha de base do contêiner.
- **stretch**: Os itens são esticados para caber no contêiner.

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

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>





Nível 7 de 24 ▼

Os sapos precisam cruzar o lago novamente, desta vez para alguns nenúfares com bastante espaço ao redor deles. Usando uma combinação de justify-contente align-items.

```
#Lago {
    display: flex;
    justify-content: space-around;
    align-items: flex-end;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>









As rãs precisam ficar na mesma ordem que seus nenúfares usando **flex-direction**. Esta propriedade CSS define a direção em que os itens são colocados no contêiner e aceita os seguintes valores:

- row: Os itens são colocados da mesma forma que a direção do texto.
- row-reverse: Os itens são posicionados opostos à direção do texto.
- column: Os itens são colocados de cima para baixo.
- column-reverse: Os itens são colocados de baixo para cima.



◀ Nível 9 de 24 ▼ ▶

Ajude as rãs a encontrar sua coluna de nenúfares usando **flex-direction**. Esta propriedade CSS define a direção em que os itens são colocados no contêiner e aceita os seguintes valores:

- row: Os itens são colocados da mesma forma que a direção do texto.
- row-reverse: Os itens são posicionados opostos à direção do texto.
- column: Os itens são colocados de cima para baixo.
- column-reverse: Os itens são colocados de baixo para cima.

```
#Lago {
    display: flex;

flex-direction: column;

}

Próximo
```





Ajude as rãs a pegar seus próprios nenúfares. Embora pareçam próximos, vai demorar tanto **flex-direction** e **justify-content** para levá-los lá.

Observe que, ao definir a direção para uma linha ou coluna invertida, o início e o fim também são invertidos.

```
#Lago {
    display: flex;
    flex-direction: row-reverse;
    justify-content: flex-end;
}

Próximo
```



■ Nível 11 de 24 • ■

Ajude as rãs a encontrar seus nenúfares usando **flex-direction** e **justify-content**.

Observe que quando a direção flexível é uma coluna, <u>justify-content</u> muda para vertical e align-items horizontal.

```
#Lago {
    display: flex;
    flex-direction: column;
    justify-content: flex-end;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>



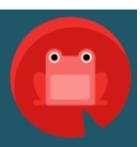


Ajude as rãs a encontrar seus nenúfares usando flex-direction e justify-content.

```
#Lago {
    display: flex;
    flex-direction: column-reverse;
    justify-content: space-between;
}

Próximo
```









Nível 13 de 24 ▼

Ajude as rãs encontrar seus lilypads usando **flex-direction**, **justify-content** e **align-items**.

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

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>

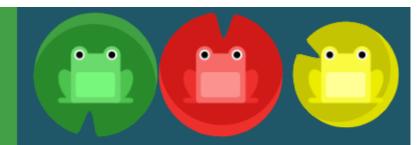




Às vezes, inverter a ordem das linhas ou colunas de um contêiner não é suficiente. Nesses casos, podemos aplicar a <u>order</u> propriedade a itens individuais. Por padrão, os itens têm um valor de 0, mas podemos usar essa propriedade para defini-la também como um valor inteiro positivo ou negativo (-2,-1,0,1,2).

Use a **order** propriedade para reordenar as rãs de acordo com seus nenúfares.

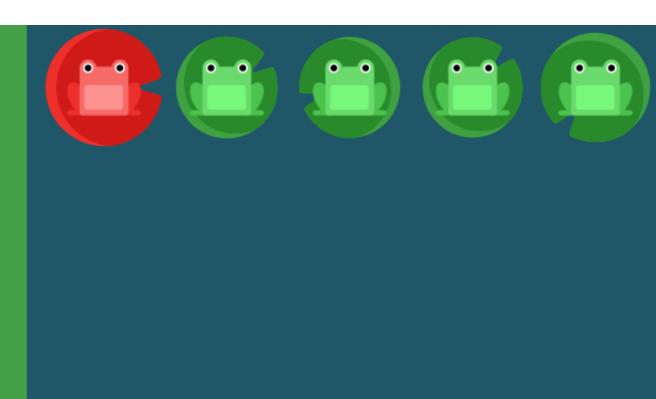
```
1 #Lago {
2   display: flex;
3 }
4 
5   .amarelo {
6   order:1;
7 }
8 
9 
10
Próximo
```



■ Nível 15 de 24 • )

Use a **order** propriedade para enviar o sapo vermelho ao seu nenúfar.

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





Outra propriedade que você pode aplicar a itens individuais é **align-self**. Esta propriedade aceita os mesmos valores **align-items** e seu valor para o item específico.

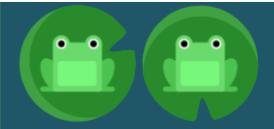
```
#Lago {
    display: flex;
    alinhar-itens: flex-start;

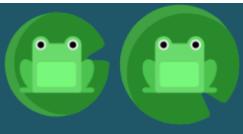
}

amarelo {
    align-self: flex-end;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>







Nível 17 de 24 ▼

Combine **order** com **align-self** para ajudar as rãs a seus destinos.

```
1 #Lago {
2  display: flex;
3  alinhar-itens: flex-start;
4 }
5
6  .amarelo {
7  order: 1;
8  align-self: flex-end;
9 }
10
Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <mark>Configurações</mark>



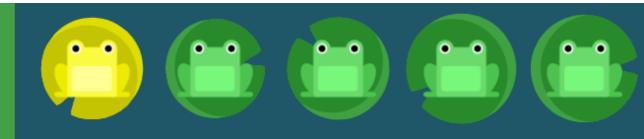




Ah não! Os sapos são todos espremidos em uma única fileira de nenúfares. Distribua-os usando a **flex-wrap** propriedade, que aceita os seguintes valores:

- nowrap: Cada item é adequado a uma única linha.
- wrap: Os itens são agrupados em linhas adicionais.
- wrap-reverse: Os itens são agrupados em linhas adicionais ao contrário.

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



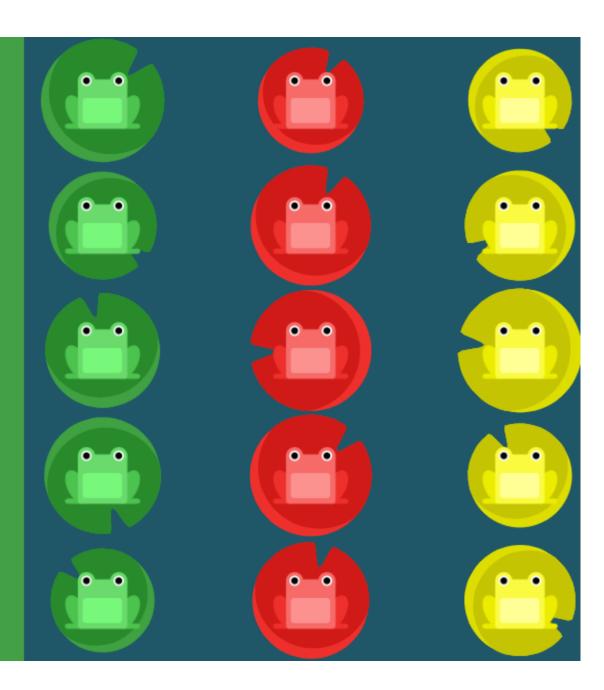


Ajude este exército de sapos a formar três colunas ordenadas usando uma combinação de **flex-direction** e **flex-wrap**.

```
#Lago {
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>



Nível 20 de 24 ▼

As duas propriedades **flex-direction** e **flex-wrap** são utilizadas tantas vezes juntos que a propriedade estenográfica **flex-flow** foi criado para combiná-los. Esta propriedade abreviada aceita o valor das duas propriedades separadas por um espaço.

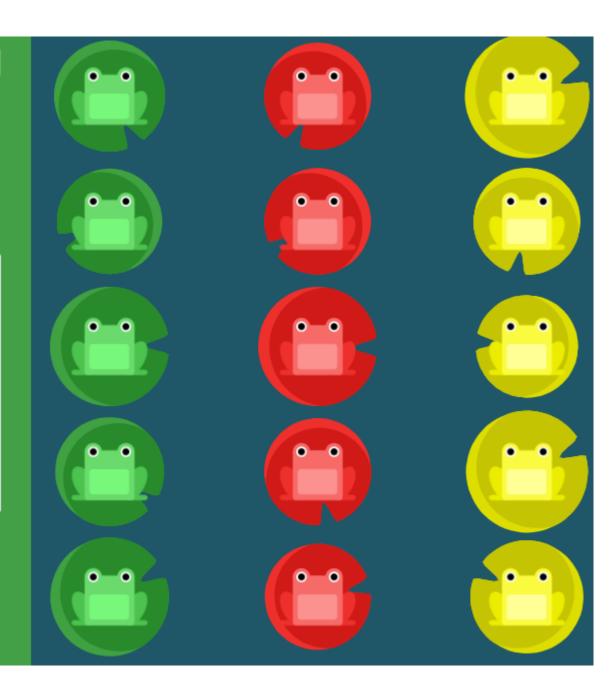
Por exemplo, você pode usar **flex-flow:** row wrap para definir linhas e agrupá-las.

Tente **flex-flow** repetir o nível anterior.

```
#Lago {
    display: flex;
    flex-flow: column wrap;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>



Nível 21 de 24 ▼

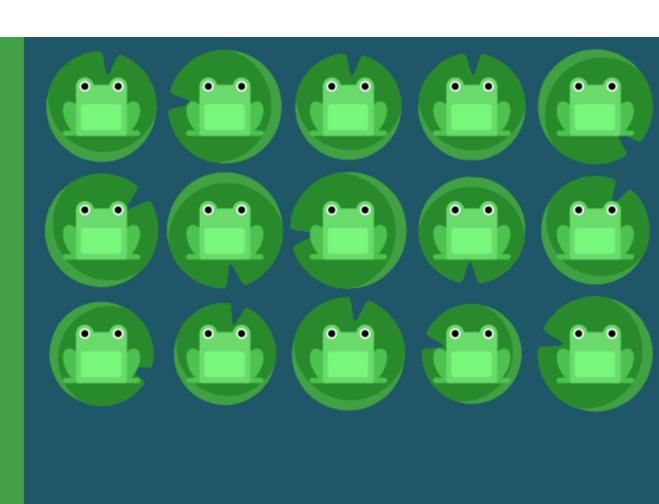
As rãs estão espalhadas por todo o tanque, mas os nenúfares estão agrupados no topo. Você pode usar align-content para definir como várias linhas são espaçadas umas das outras. Esta propriedade assume os seguintes valores:

- flex-start: As linhas são embaladas na parte superior do contêiner.
- flex-end: As linhas são embaladas no fundo do contêiner.
- center: As linhas são embaladas no centro vertical do contêiner.
- space-between: As linhas são exibidas com espaçamento igual entre elas.
- space-around: As linhas são exibidas com espaçamento igual ao redor delas.
- **stretch**: As linhas são esticadas para caber no contêiner.

Isso pode ser confuso, mas <u>align-content</u> determina o espaçamento entre as linhas, enquanto <u>align-items</u> determina como os itens como um todo são alinhados dentro do contêiner. Quando há apenas uma linha, align-content não tem efeito.

```
#Lago {
    display: flex;
    flex-wrap: wrap;
    align-content: flex-start;
}

Próximo
```





Agora, a correnteza amontoou os nenúfares na parte inferior. Use **align-content** para guiar os sapos até lá.

```
#Lago {
    display: flex;
    flex-wrap: wrap;
    align-content: flex-end;
}

Próximo
```

Flexbox Froggy é criado por Codepip • GitHub • Twitter • Configurações



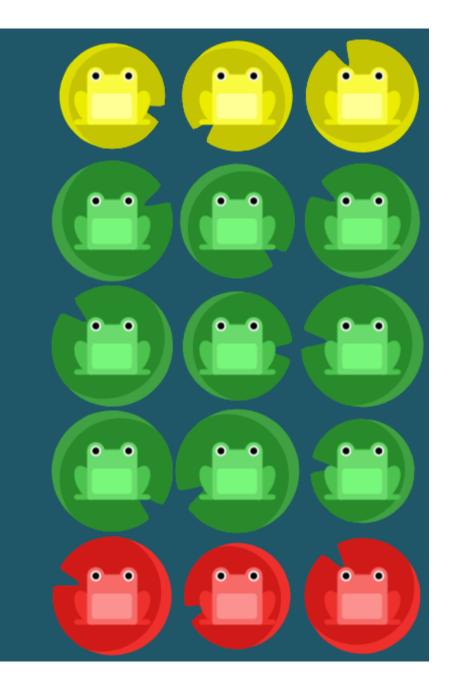
■ Nível 23 de 24 ■ ■

As rãs deram uma festa, mas é hora de voltar para casa. Use uma combinação de **flex-direction** e **align-content** para levá-los aos nenúfares.

```
#Lago {
    display: flex;
    flex-wrap: wrap;
    flex-direction: column-reverse;
    align-content: center;
}

Próximo
```

Flexbox Froggy é criado por <u>Codepip</u> • <u>GitHub</u> • <u>Twitter</u> • <u>Configurações</u>



■ Nível 24 de 24 ▼

Traga os sapos para casa uma última vez usando as propriedades CSS que você aprendeu:

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

```
#Lago {
    display: flex;
    flex-flow:column-reverse wrap-reverse;
    justify-content:center;
    align-content:space-between;
}

Próximo
```











