

FLEXBOX FROGGY

◀ Nível 2 de 24 ▶

Use `justify-content` 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.

```
1 #Lago {  
2   display: flex;  
3   justify-content: center  
4 }
```

Próximo



FLEXBOX FROGGY

◀ Nível 3 de 24 ▶

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`.

```
1 #Lago {  
2   display: flex;  
3   justify-content: space-around  
4 }
```

Próximo



FLEXBOX FROGGY

◀ Nível 4 de 24 ▶

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.

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

Próximo



FLEXBOX FROGGY

◀ Nível 5 de 24 ▶

Agora use **align-items** 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 }
```

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY



Nível 6 de 24 ▾



Conduza o sapo para o centro da lagoa usando uma combinação de `justify-content` e `align-items`.

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

Próximo



FLEXBOX FROGGY

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-content` e `align-items`.

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ Nível 8 de 24 ▶

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.

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

Próximo



FLEXBOX FROGGY



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.

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

5
6
7
8
9
10

Próximo



FLEXBOX FROGGY

◀ Nível 10 de 24 ▶

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.

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

Próximo



FLEXBOX FROGGY

◀ 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, `justify-content` muda para vertical e `align-items` horizontal.

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ Nível 12 de 24 ▶

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

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ 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;  
4   justify-content: center;  
5   align-items: flex-end;  
6 }
```

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ Nível 14 de 24 ▶

Às vezes, inverter a ordem das linhas ou colunas de um contêiner não é suficiente. Nesses casos, podemos aplicar a `order` 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



FLEXBOX FROGGY

◀ 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



FLEXBOX FROGGY

◀ Nível 16 de 24 ▶

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.

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#) .



FLEXBOX FROGGY

◀ 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 [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ Nível 18 de 24 ▶

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 }
```

Próximo



FLEXBOX FROGGY

◀ Nível 19 de 24 ▶

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

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

Próximo

Flexbox Froggy é criado por [CodePip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ 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.

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY



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 `align-content` determina o espaçamento entre as linhas, enquanto `align-items` determina como os itens como um todo são alinhados dentro do contêiner. Quando há apenas uma linha, `align-content` não tem efeito.

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

Próximo



FLEXBOX FROGGY



Nível 22 de 24 ▾



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

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY

◀ 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.

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

Quer aprender a grade CSS? Jogue [Grid Garden](#).



FLEXBOX FROGGY



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`

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

Próximo

Flexbox Froggy é criado por [Codepip](#) • [GitHub](#) • [Twitter](#) • [Configurações](#)

