

FLEXBOX FROGGY

◀ Level 3 of 24 ▶

Help all three frogs find their lilypads just by using `justify-content`. This time, the lilypads have lots of space all around them.

If you find yourself forgetting the possible values for a property, you can click on the property name to view them. Try clicking on `justify-content`.

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

Next

FLEXBOX FROGGY

◀ Level 4 of 24 ▶

Now the lilypads on the edges have drifted to the shore, increasing the space between them. Use `justify-content`. This time, the lilypads have equal spacing between them.

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

Next



FLEXBOX FROGGY

Level 5 of 24

Now use `align-items` to help the frogs get to the bottom of the pond. This CSS property aligns items vertically and accepts the following values:

- `flex-start`: Items align to the top of the container.
- `flex-end`: Items align to the bottom of the container.
- `center`: Items align at the vertical center of the container.
- `baseline`: Items display at the baseline of the container.
- `stretch`: Items are stretched to fit the container.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



#

FLEXBOX FROGGY

Level 6 of 24

Lead the frog to the center of the pond using a combination of `justify-content` and `align-items`.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



FLEXBOX FROGGY

◀ Level 7 of 24 ▶

The frogs need to cross the pond again, this time for some lily pads with plenty of space around them. Use a combination of `justify-content` and `align-items`.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



FLEXBOX FROGGY

◀ Level 8 of 24 ▶

The frogs need to get in the same order as their lily pads using `flex-direction`. This CSS property defines the direction items are placed in the container, and accepts the following values:

- `row`: Items are placed the same as the text direction.
- `row-reverse`: Items are placed opposite to the text direction.
- `column`: Items are placed top to bottom.
- `column-reverse`: Items are placed bottom to top.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



FLEXBOX FROGGY

◀ Level 12 of 24 ▶




Help the frogs find their lilypads using `flex-direction` and `justify-content`.


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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).





FLEXBOX FROGGY

◀ Level 13 of 24 ▶




Help the frogs find their lilypads using `flex-direction`, `justify-content`, and `align-items`.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



FLEXBOX FROGGY

◀ Level 9 of 24 ▶

Help the frogs find their column of lilypads using **flex-direction**. This CSS property defines the direction items are placed in the container, and accepts the following values:

- **row**: Items are placed the same as the text direction.
- **row-reverse**: Items are placed opposite to the text direction.
- **column**: Items are placed top to bottom.
- **column-reverse**: Items are placed bottom to top.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).



FLEXBOX FROGGY

◀ Level 10 of 24 ▶

Help the frogs get to their own lilypads. Although they seem close, it will take both **flex-direction** and **justify-content** to get them there.

Notice that when you set the direction to a reversed row or column, start and end are also reversed.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)



FLEXBOX FROGGY

Level 14 of 24

Sometimes reversing the row or column order of a container is not enough. In these cases, we can apply the `order` property to individual items. By default, items have a value of 0, but we can use this property to also set it to a positive or negative integer value (-2, -1, 0, 1, 2).


Use the `order` property to reorder the frogs according to their lilypads.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchorem](#).



FLEXBOX FROGGY


Level 15 of 24

Use the `order` property to send the red frog to his lilypad.

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

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)



FLEXBOX FROGGY

Level 16 of 24


Another property you can apply to individual items is `align-self`. This property accepts the same values as `align-items` and its value for the specific item.

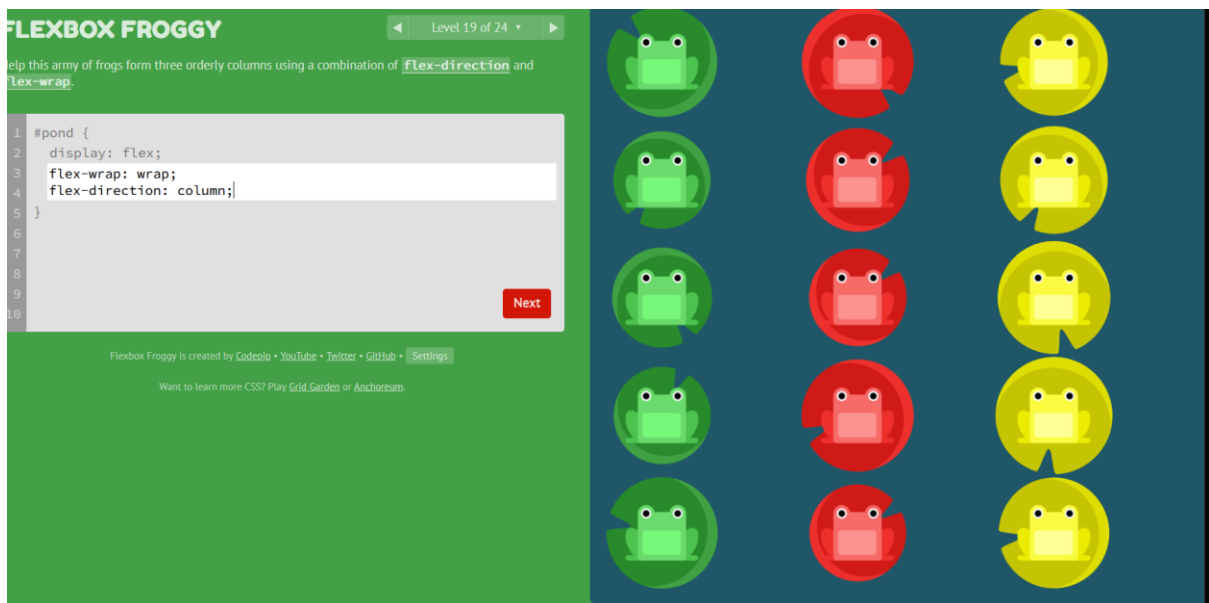
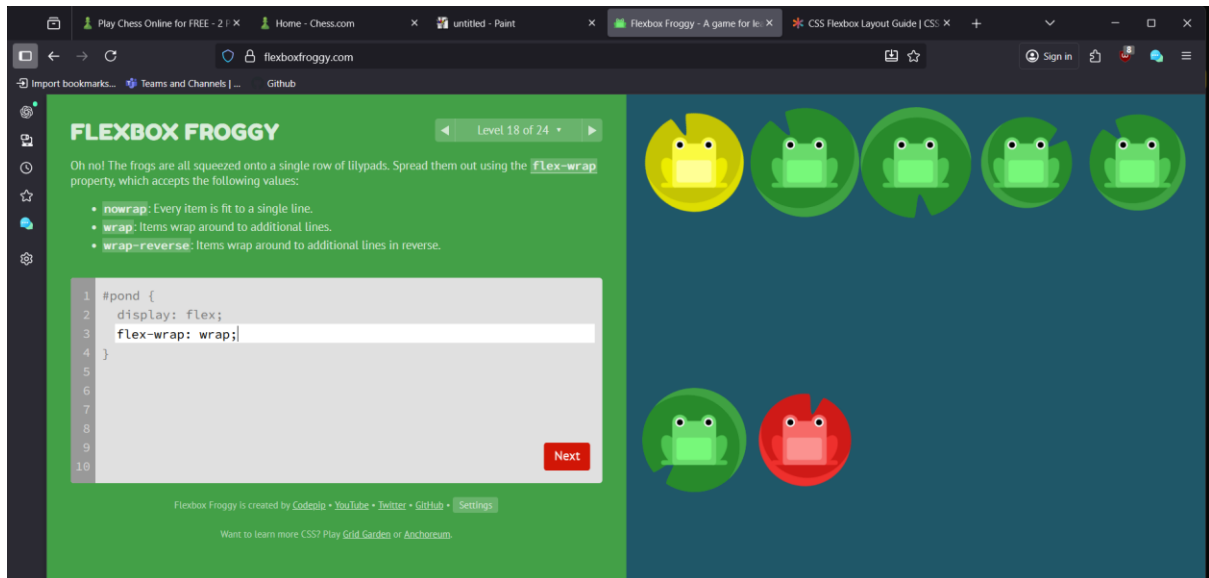
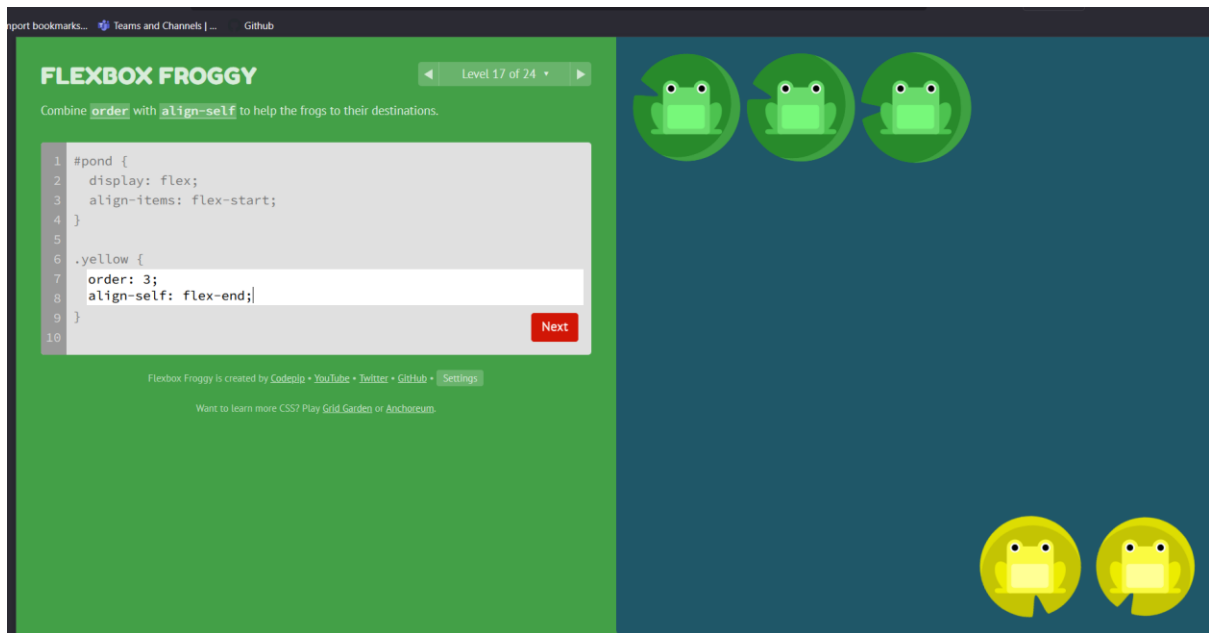
```
1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   align-self: flex-end;
8 }
9
10
```

Next

Flexbox Froggy is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchorem](#).





port bookmarks... Teams and Channels | ... Github

FLEXBOX FROGGY

Level 20 of 24

The two properties **flex-direction** and **flex-wrap** are used so often together that the shorthand property **flex-flow** was created to combine them. This shorthand property accepts the value of the two properties separated by a space.

For example, you can use **flex-flow: row wrap** to set rows and wrap them.

Try using **flex-flow** to repeat the previous level.

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

Next

[Flexbox Froggy](#) is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [Github](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).

← → ↺

flexboxfroggy.com

Import bookmarks... Teams and Channels | ... Github

FLEXBOX FROGGY

Level 21 of 24

The frogs are spread all over the pond, but the lily pads are bunched at the top. You can use **align-content** to set how multiple lines are spaced apart from each other. This property takes the following values:

- **flex-start**: Lines are packed at the top of the container.
- **flex-end**: Lines are packed at the bottom of the container.
- **center**: Lines are packed at the vertical center of the container.
- **space-between**: Lines display with equal spacing between them.
- **space-around**: Lines display with equal spacing around them.
- **stretch**: Lines are stretched to fit the container.

This can be confusing, but **align-content** determines the spacing between lines, while **align-items** determines how the items as a whole are aligned within the container. When there is only one line, **align-content** has no effect.

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

Next

[Flexbox Froggy](#) is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [Github](#) • [Settings](#)

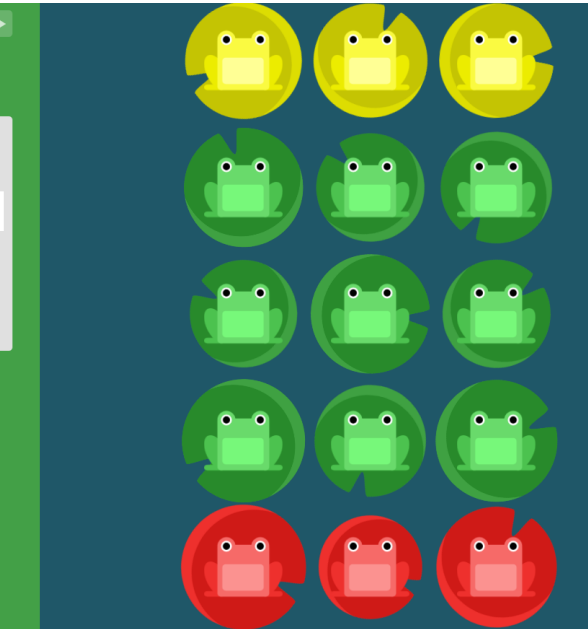
Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).

ENG DE

12:58 15/10/2025


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

Next




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

Next

FLEXBOX FROGGY

◀ Level 24 of 24 ▶

Bring the frogs home one last time by using the CSS properties you've learned:

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

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

Next

Flexbox Froggy is created by [Codetly](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn more CSS? Play [Grid Garden](#) or [Anchoreum](#).

