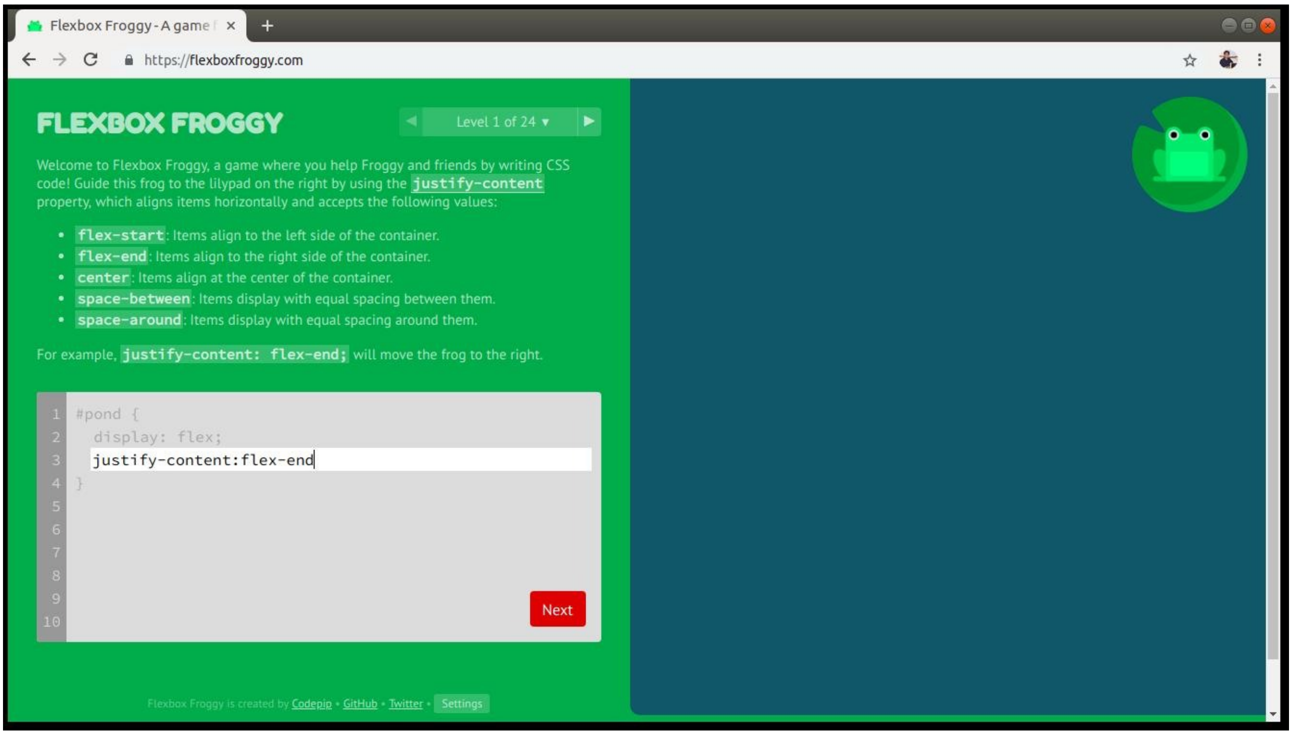
Question No:- 1



Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lilypad on the right by using the `justify-content` property, which aligns items horizontally and accepts the following values:

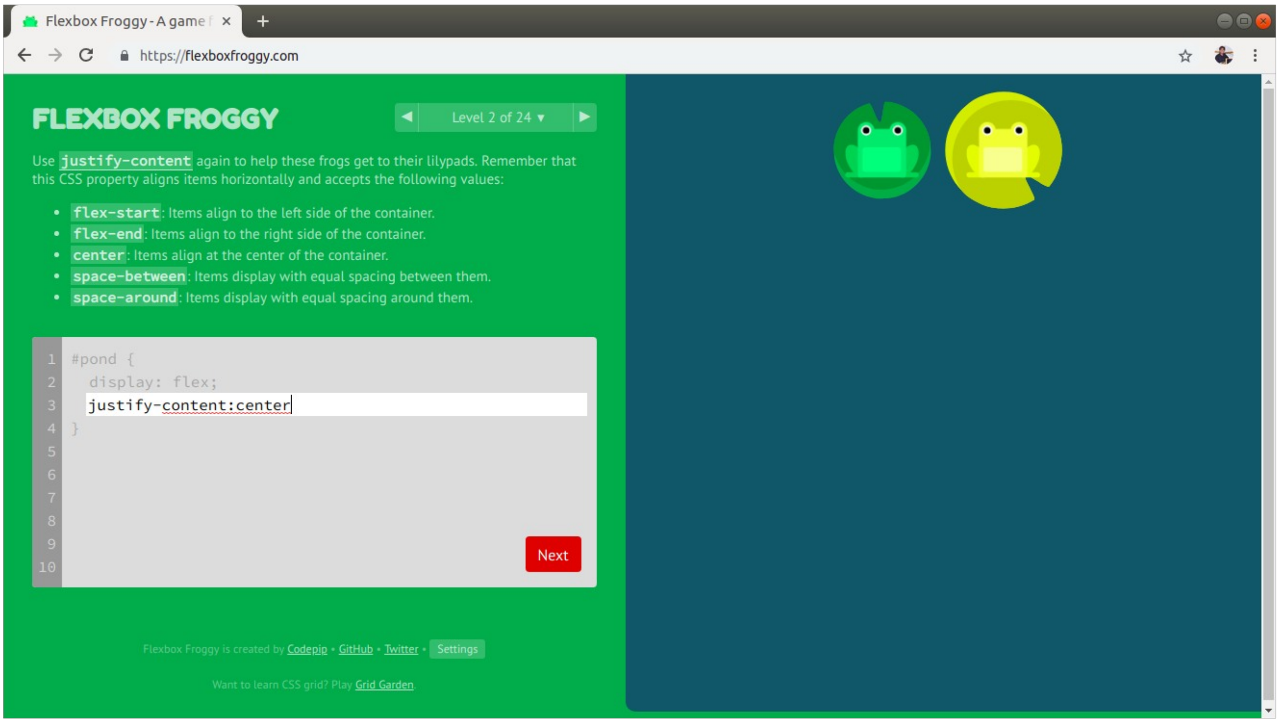
- `flex-start`: Items align to the left side of the container.
- `flex-end`: Items align to the right side of the container.
- `center`: Items align at the center of the container.
- `space-between`: Items display with equal spacing between them.
- `space-around`: Items display with equal spacing around them.

For example, `justify-content: flex-end;` will move the frog to the right.

```
#pond {
  display: flex;
 justify-content:flex-end
}
```

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Question No:- 2



Use `justify-content` again to help these frogs get to their lilypads. Remember that this CSS property aligns items horizontally and accepts the following values:

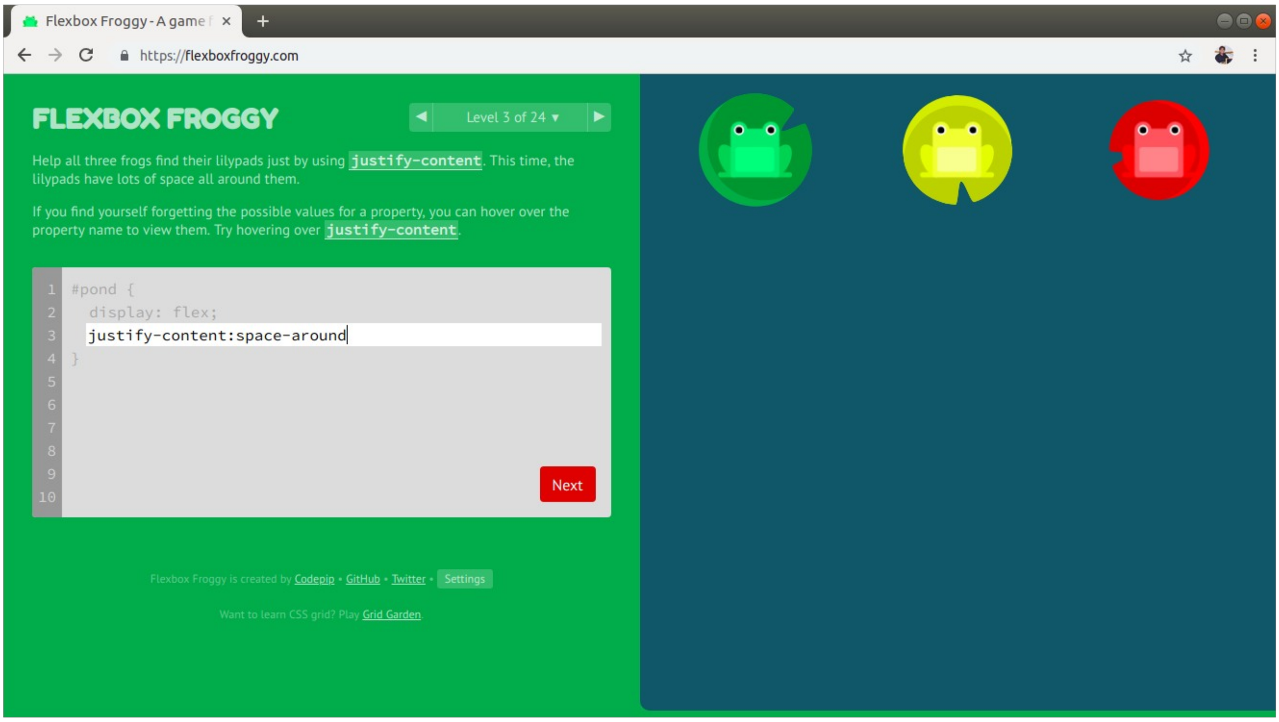
- `flex-start`: Items align to the left side of the container.
- `flex-end`: Items align to the right side of the container.
- `center`: Items align at the center of the container.
- `space-between`: Items display with equal spacing between them.
- `space-around`: Items display with equal spacing around them.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden

Question No:- 3



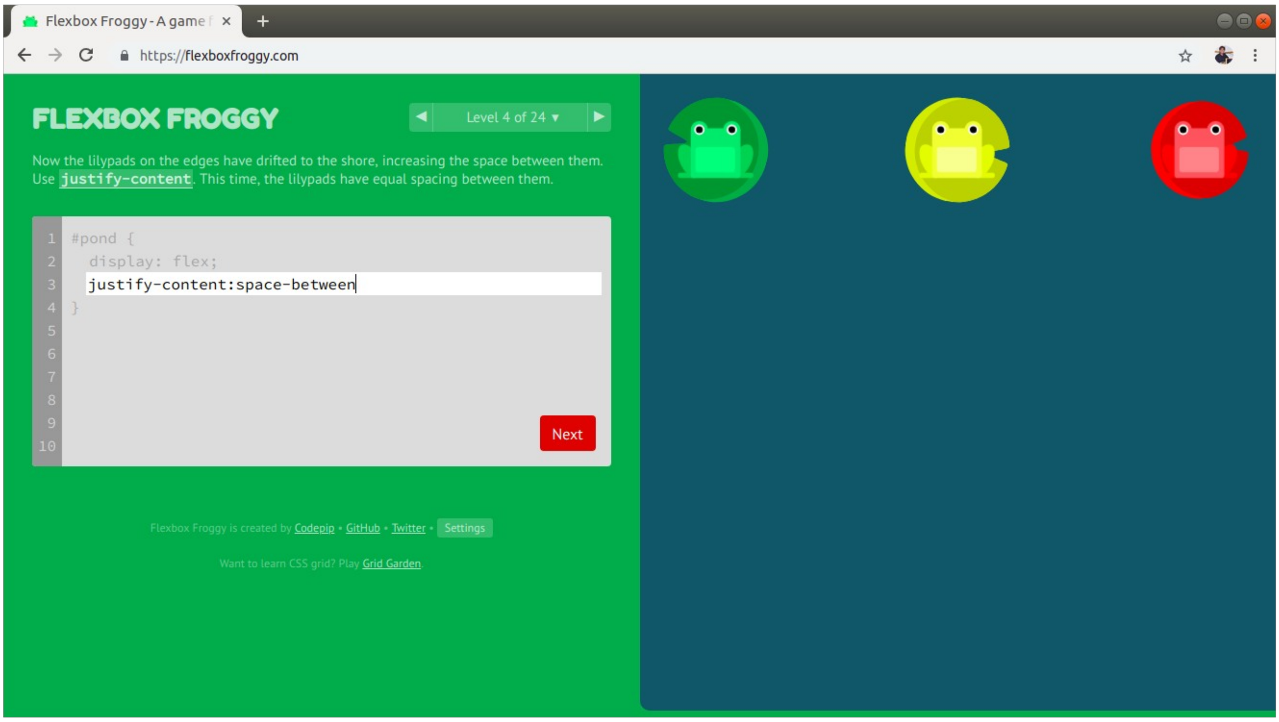
Flexbox Froggy - A game f ×
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 3 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden.

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 hover over the property name to view them. Try hovering over `justify-content`.

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

Question No:- 4

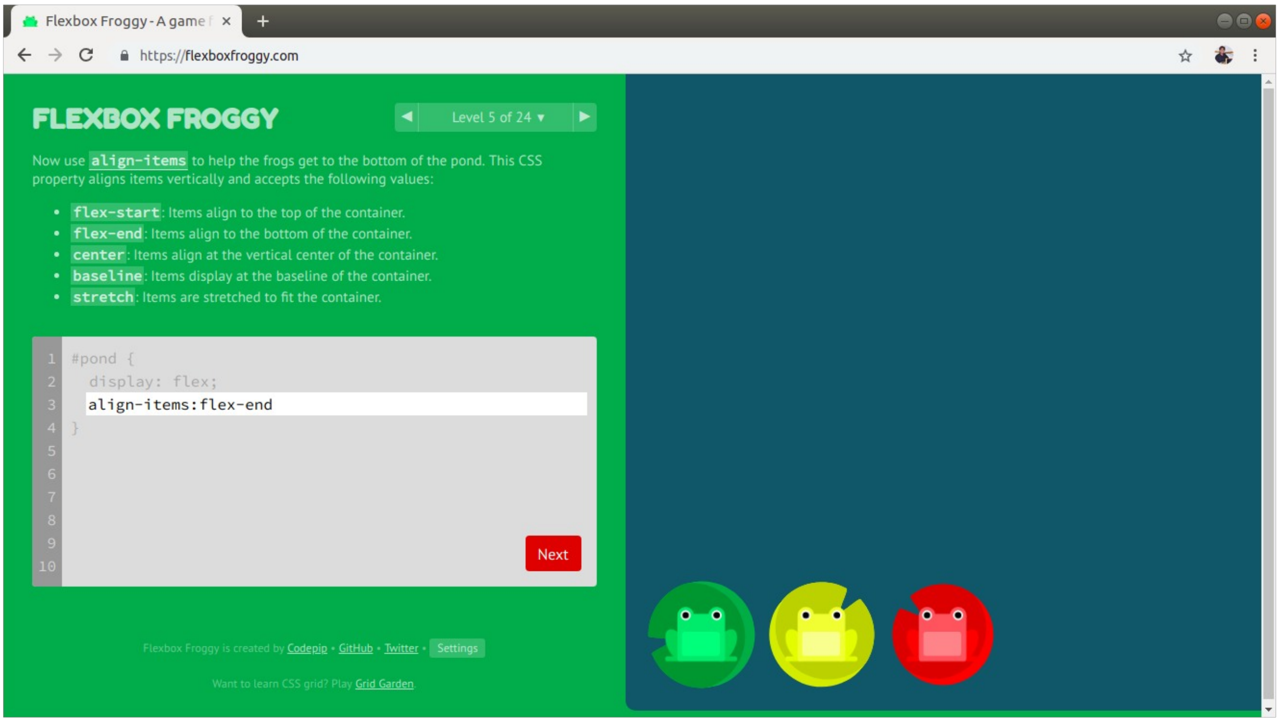


Flexbox Froggy - A game f ×
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 4 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden.

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.

```
#pond {
  display: flex;
  justify-content:space-between
}
```

Question No:- 5



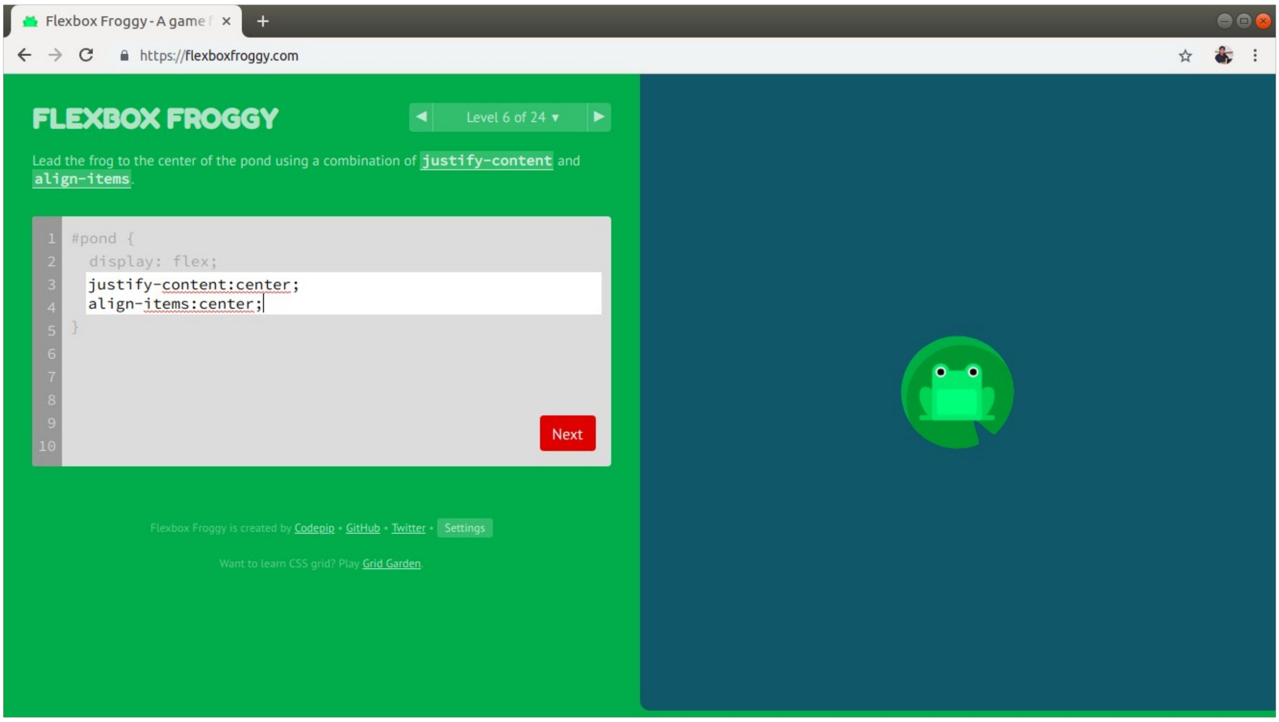
Now use `align-items` to help the frogs get to the bottom of the pond. This CSS property aligns items vertically and accepts the following values:

- `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.

```
#pond {
  display: flex;
  align-items:flex-end
}
```

Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

Question No:- 6



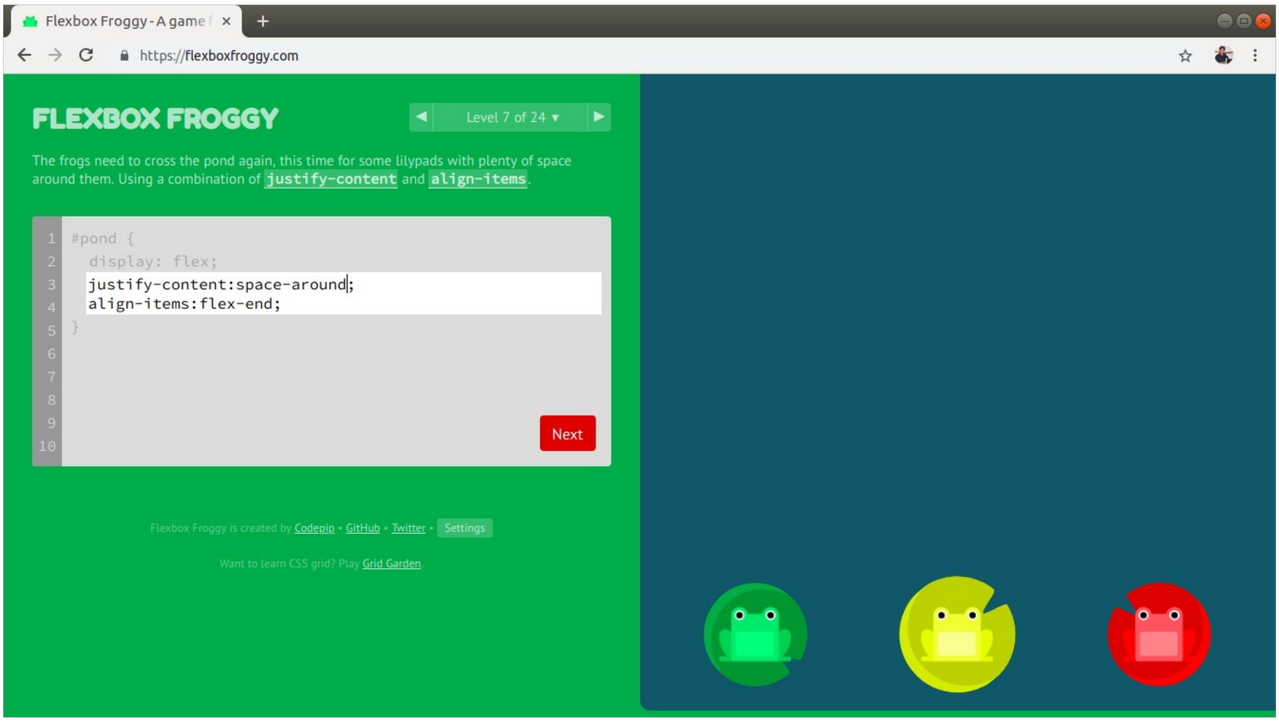
Flexbox Froggy - A game
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 6 of 24

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

```
#pond {
  display: flex;
  justify-content:center;
  align-items:center;
}
```

Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

Question No:- 7



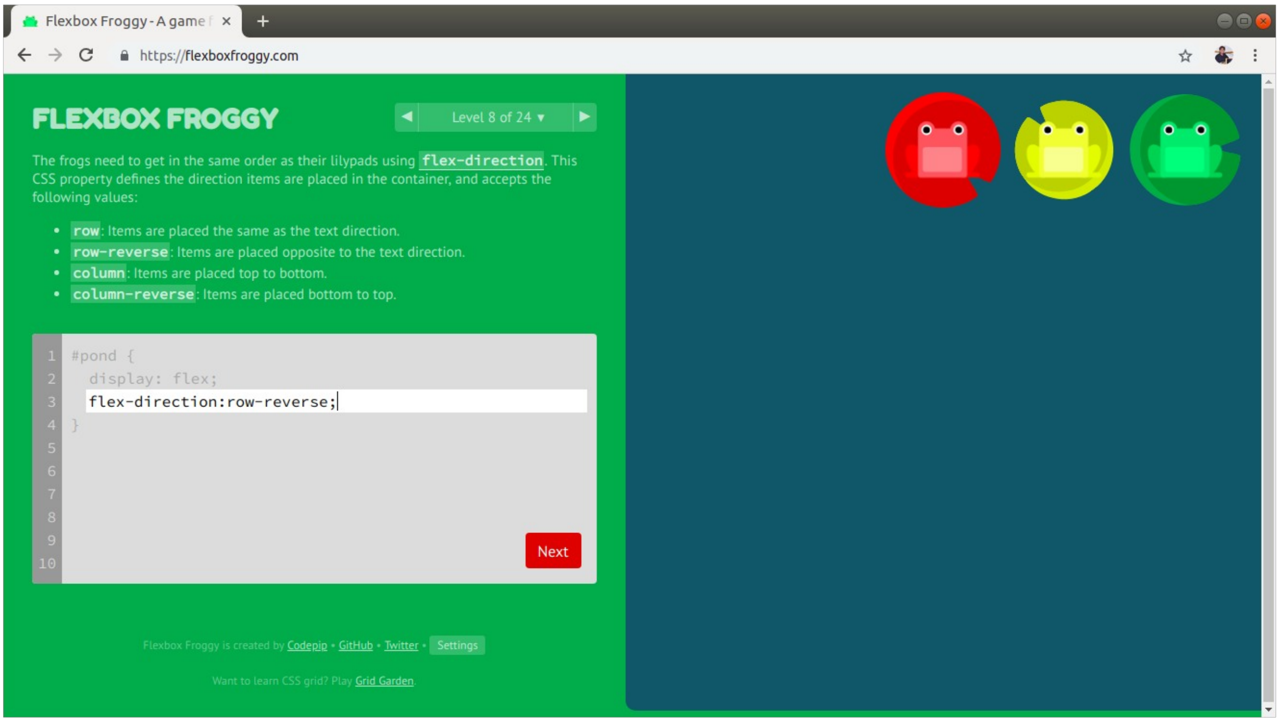
The frogs need to cross the pond again, this time for some lilypads with plenty of space around them. Using a combination of `justify-content` and `align-items`.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

Question No:- 8



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

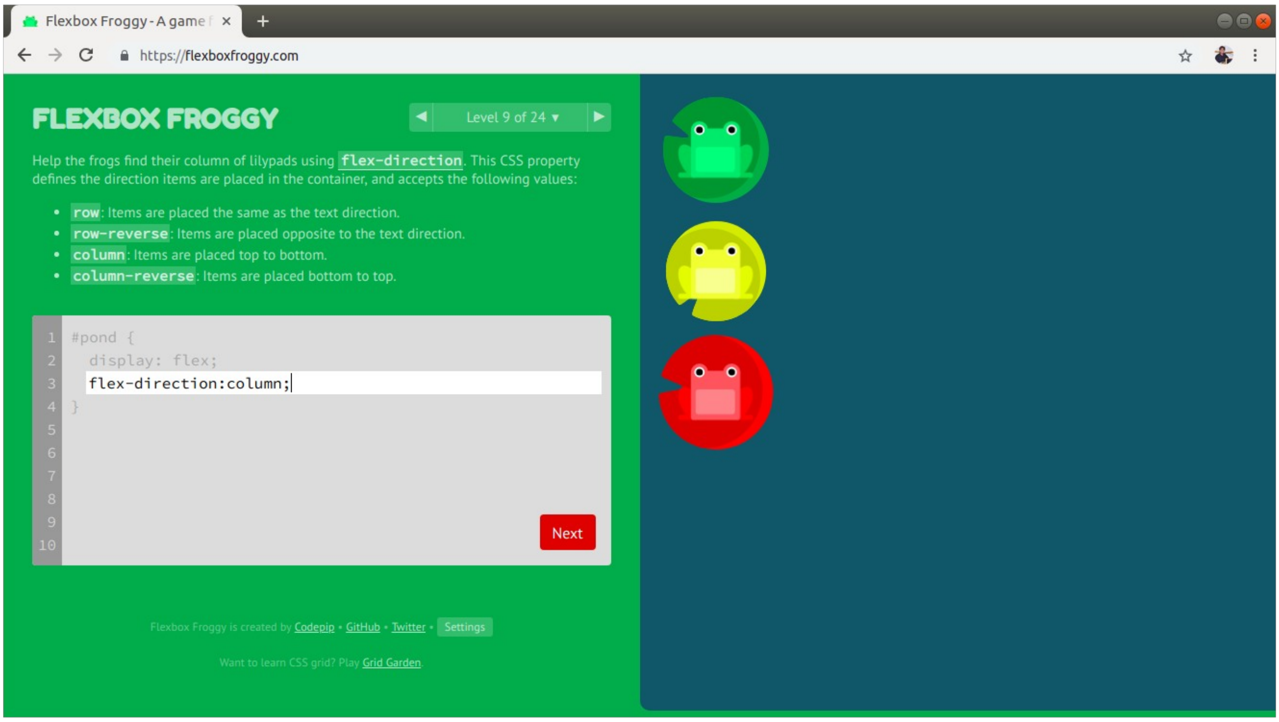
- `row`: Items are placed the same as the text direction.
- `row-reverse`: Items are placed opposite to the text direction.
- `column`: Items are placed top to bottom.
- `column-reverse`: Items are placed bottom to top.

```
#pond {
  display: flex;
  flex-direction:row-reverse;
}
```

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

Question No:- 9



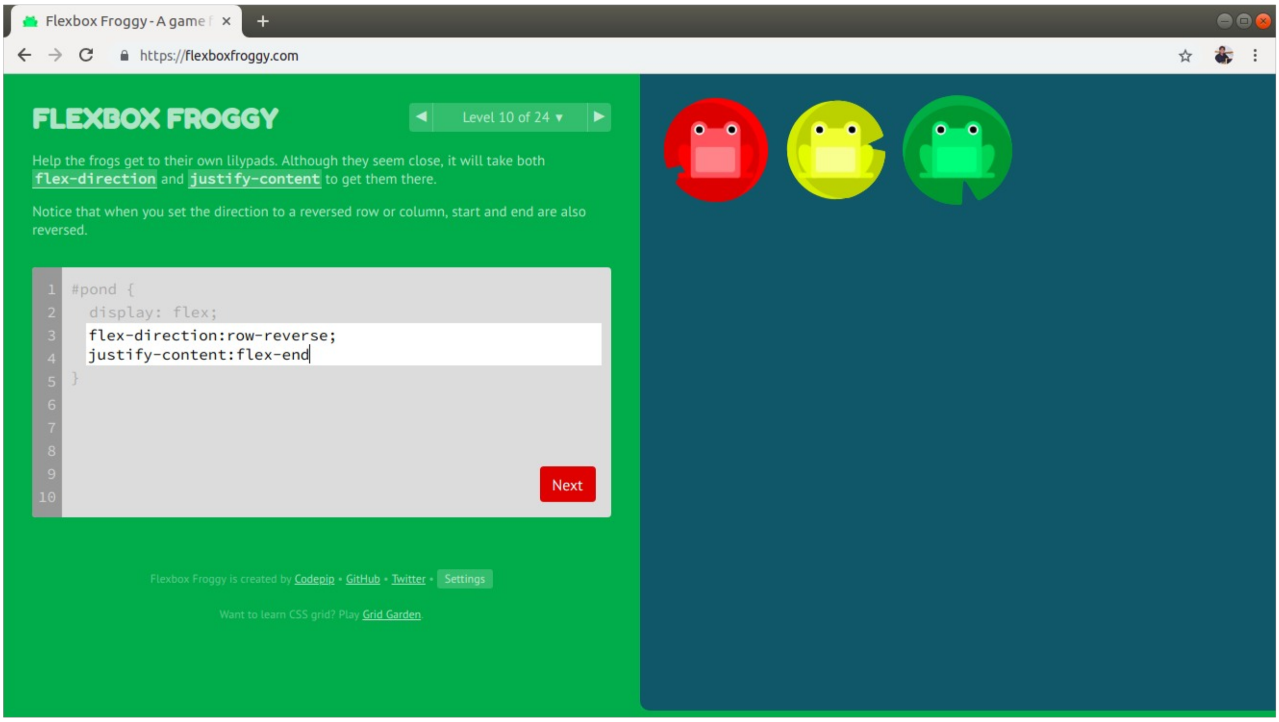
Help the frogs find their column of lilypads using `flex-direction`. This CSS property defines the direction items are placed in the container, and accepts the following values:

- `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.

```
#pond {
  display: flex;
  flex-direction:column;
}
```

Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

Question No:- 10



Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 10 of 24 ▾

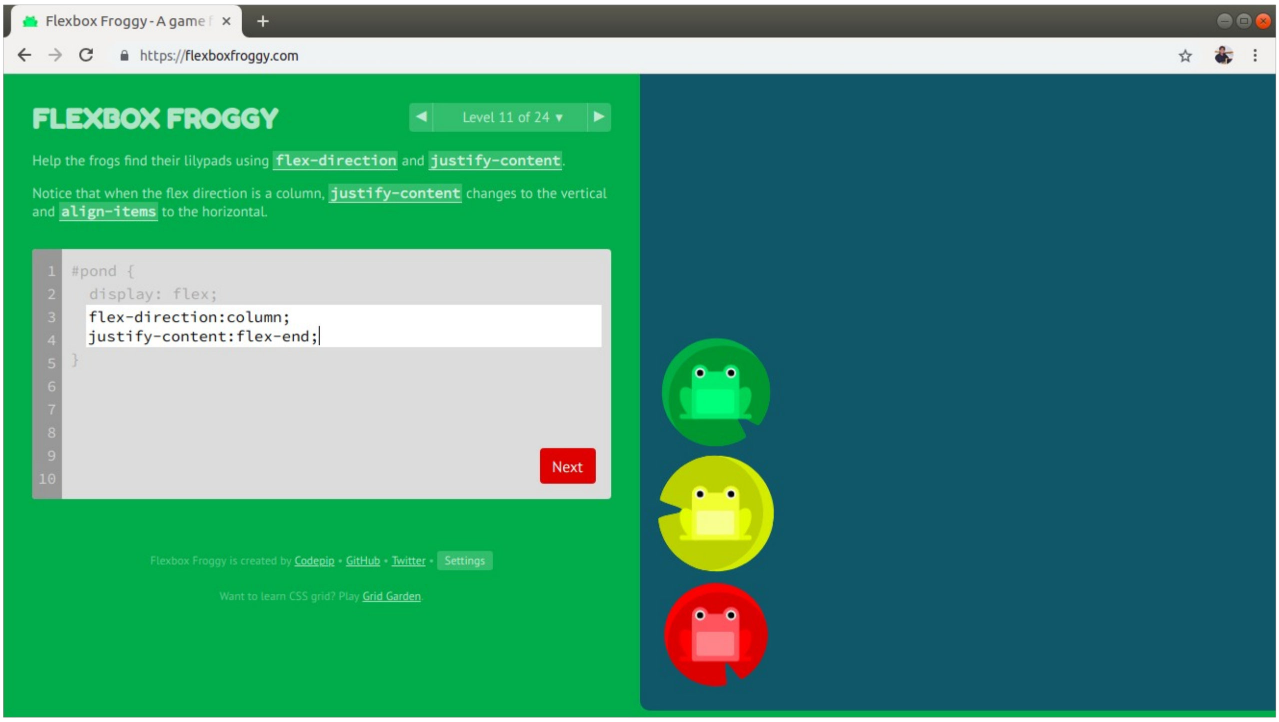
Help the frogs get to their own lilypads. Although they seem close, it will take both `flex-direction` and `justify-content` to get them there.

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

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

Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

Question No:- 11



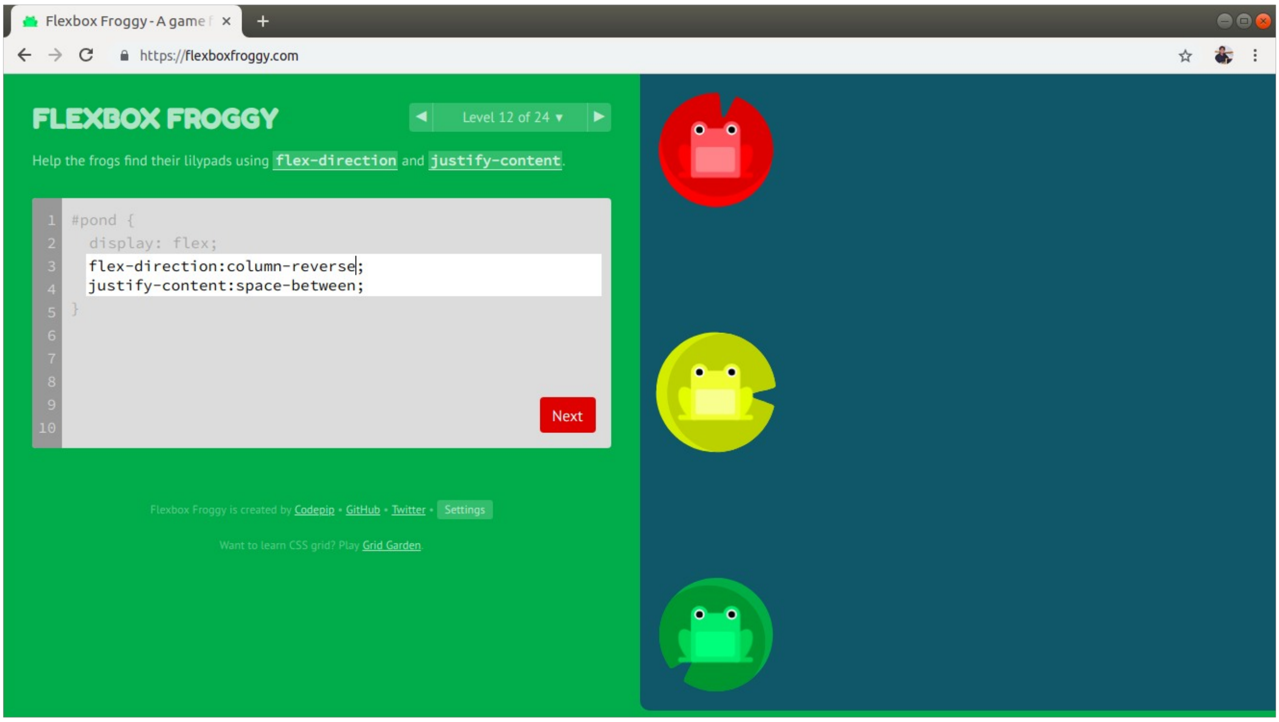
Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 11 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

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

Notice that when the flex direction is a column, `justify-content` changes to the vertical and `align-items` to the horizontal.

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

Question No:- 12

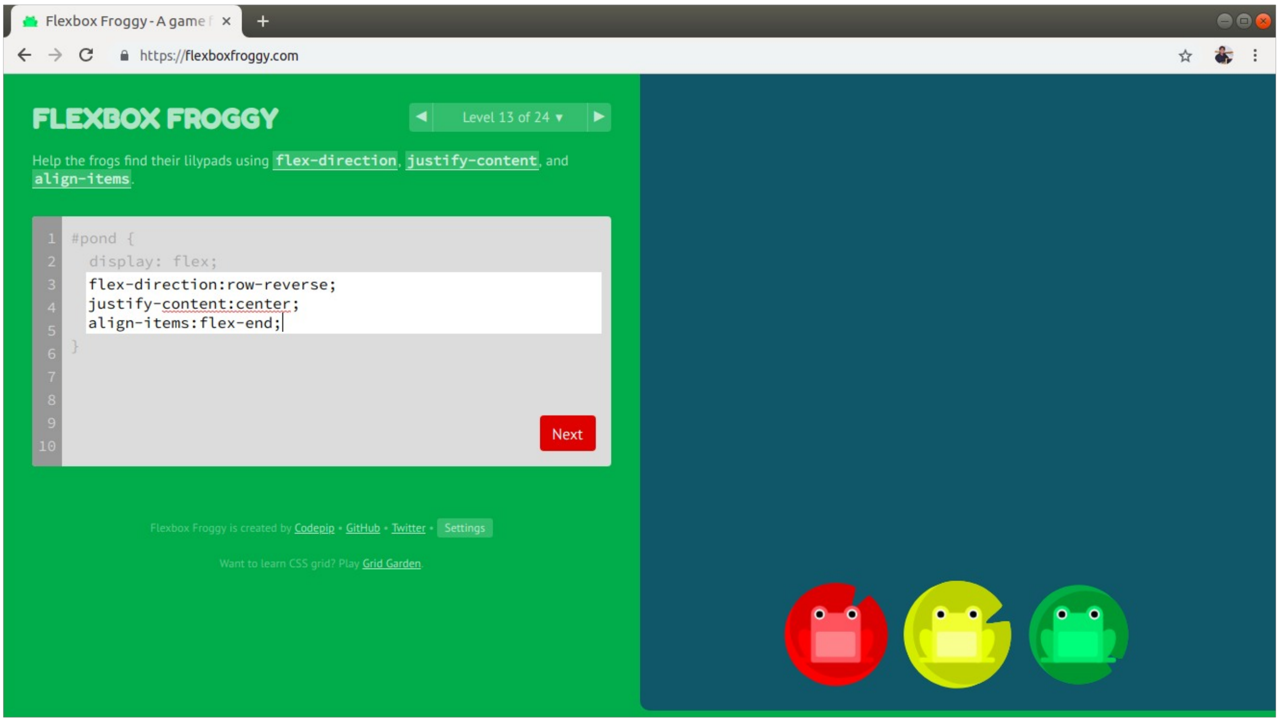


Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 12 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

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

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

Question No:- 13

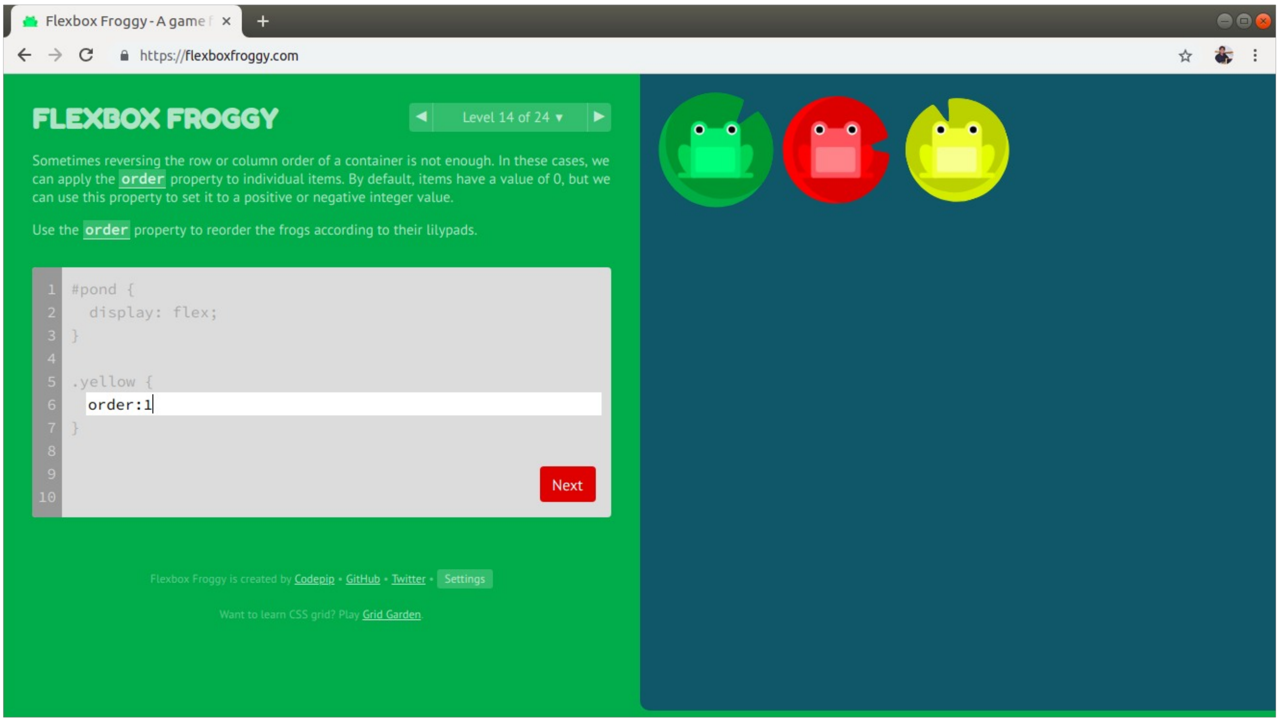


Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 13 of 24
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

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

```
#pond {
  display: flex;
  flex-direction:row-reverse;
  justify-content:center;
  align-items:flex-end;
}
```

Question No:- 14



Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 14 of 24
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

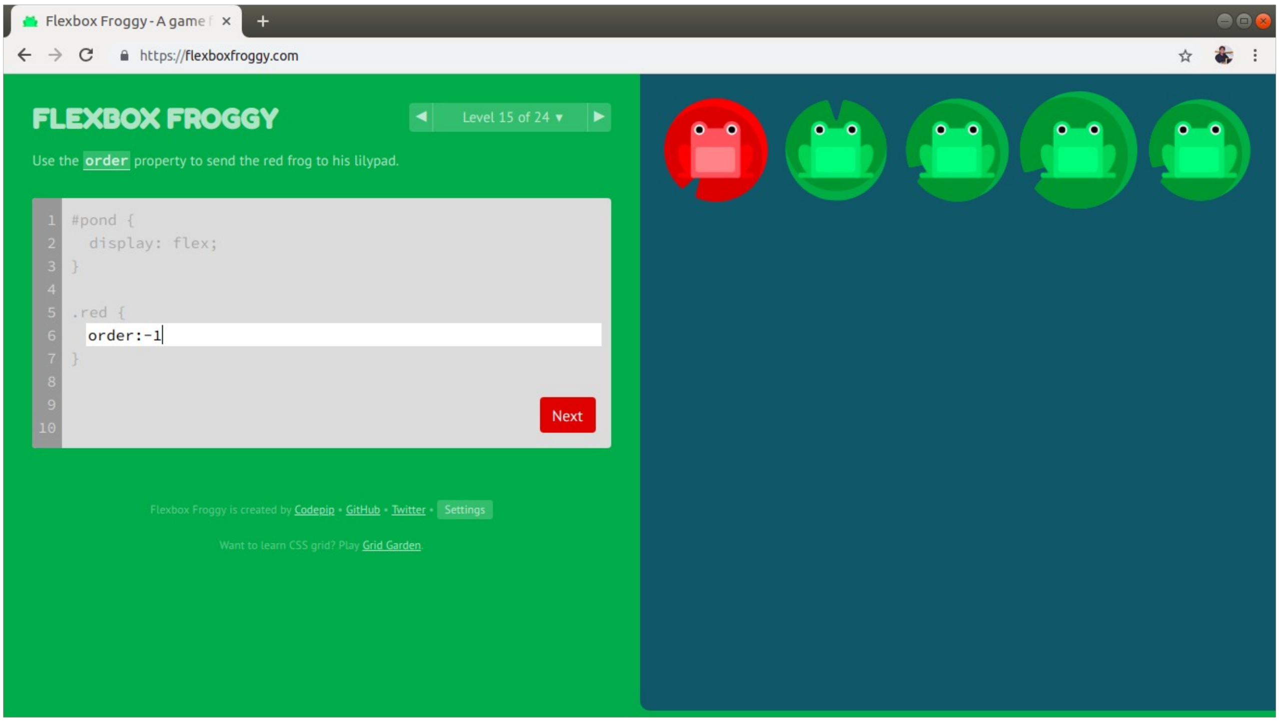
Sometimes reversing the row or column order of a container is not enough. In these cases, we can apply the `order` property to individual items. By default, items have a value of 0, but we can use this property to set it to a positive or negative integer value.

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

```
#pond {
  display: flex;
}

.yellow {
  order:1
}
```

Question No:- 15



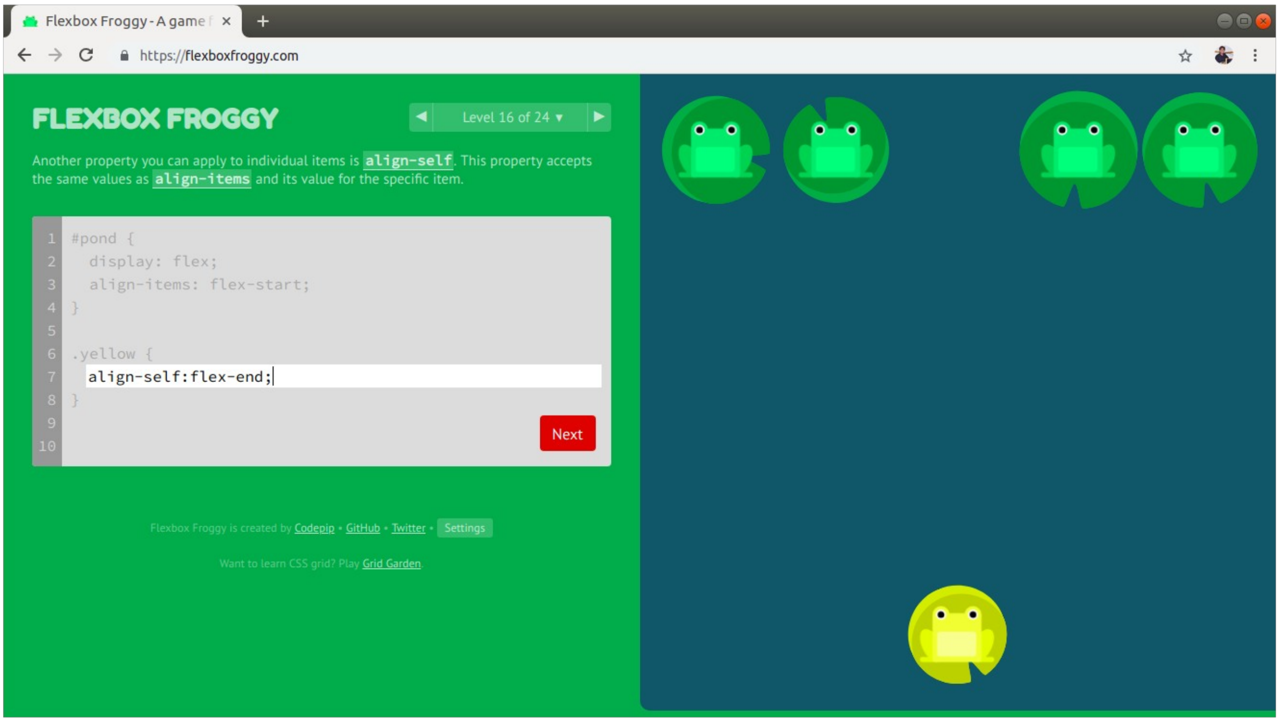
Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 15 of 24
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

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

```
#pond {
  display: flex;
}

.red {
 order:-1
}
```

Question No:- 16



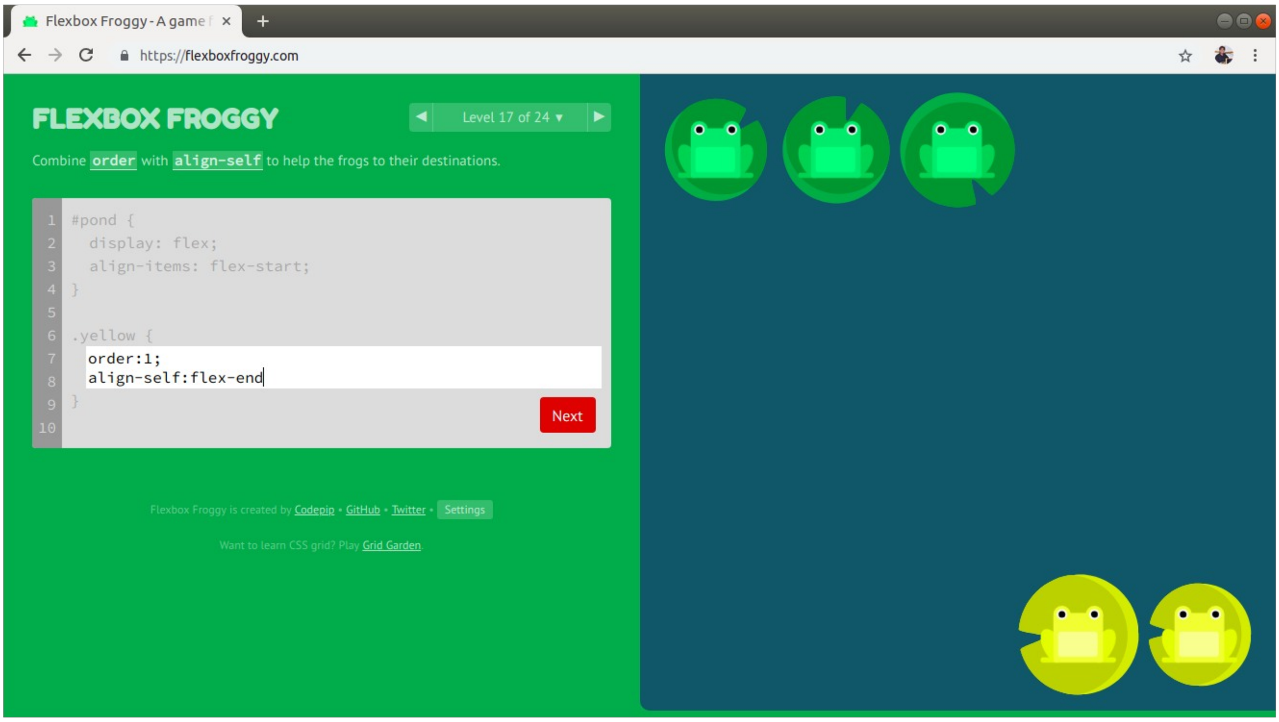
Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 16 of 24
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden

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.

```
#pond {
  display: flex;
  align-items: flex-start;
}

.yellow {
 align-self:flex-end;
}
```

Question No:- 17

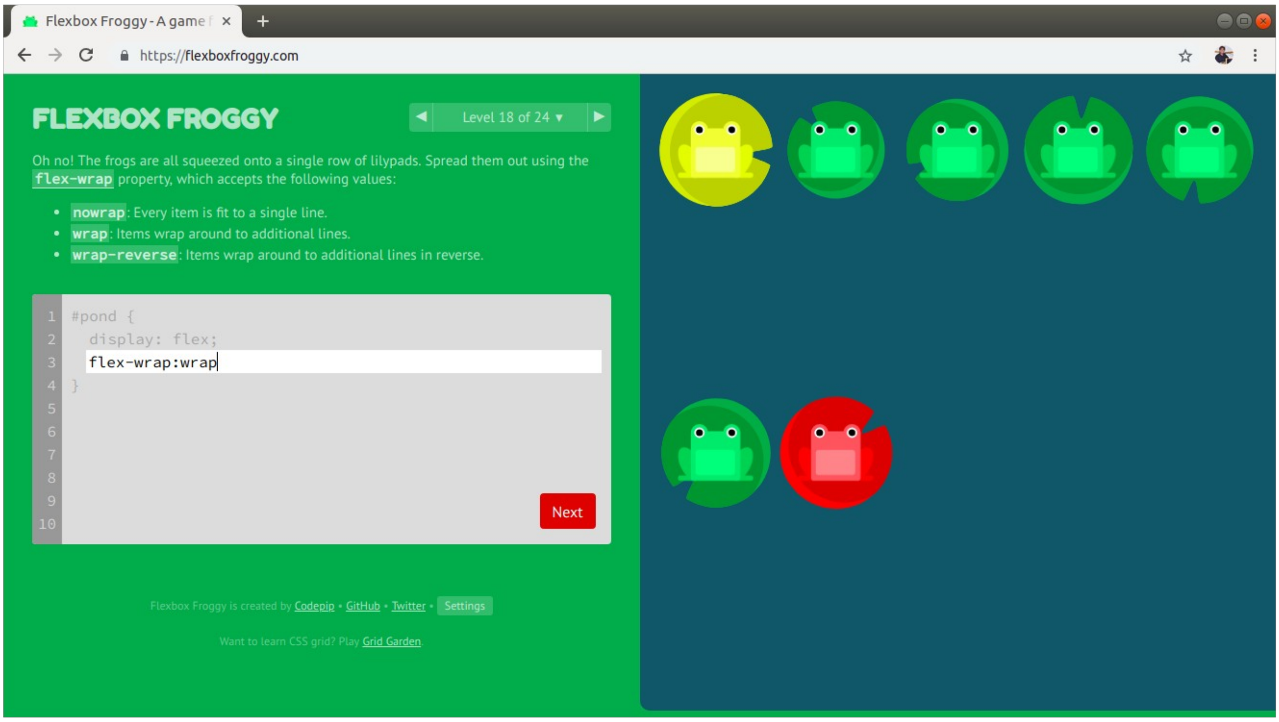

Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 17 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden.

Combine order with align-self to help the frogs to their destinations.

```
#pond {
  display: flex;
  align-items: flex-start;
}

.yellow {
  order:1;
  align-self:flex-end
}
```

Question No:- 18

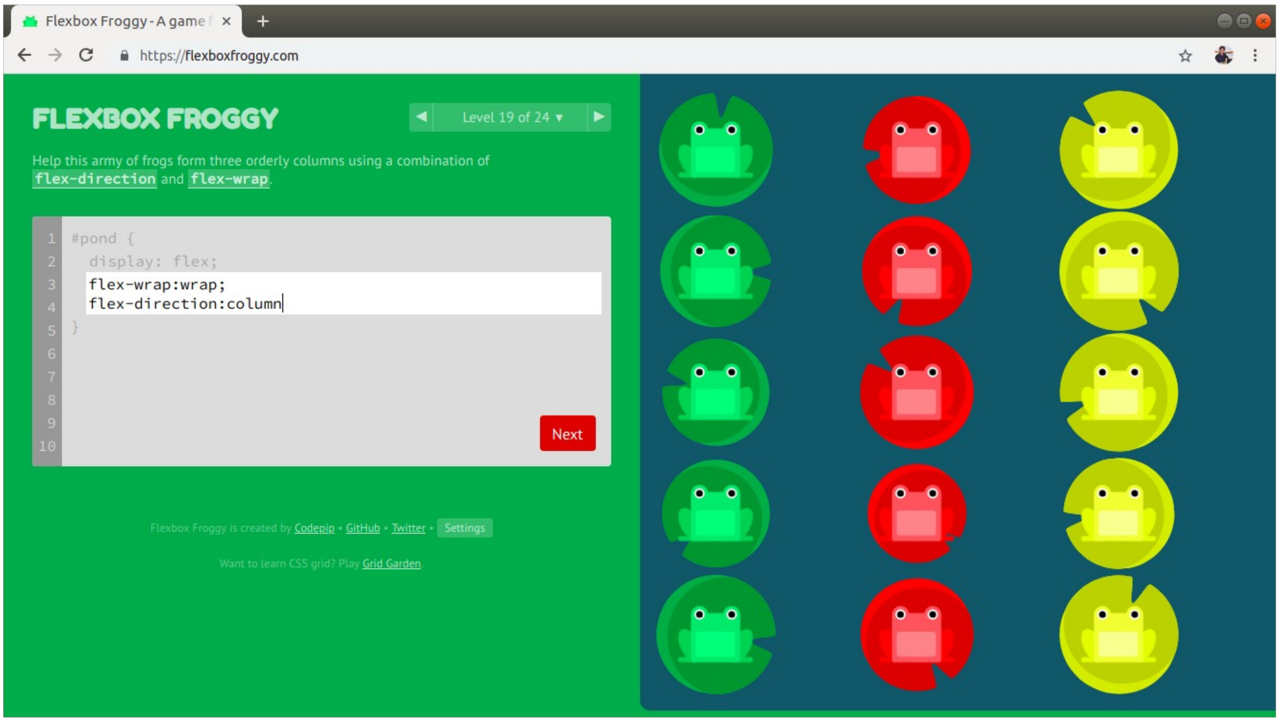

Flexbox Froggy - A game f
https://flexboxfroggy.com
FLEXBOX FROGGY
Level 18 of 24 ▾
Next
Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings
Want to learn CSS grid? Play Grid Garden.

Oh no! The frogs are all squeezed onto a single row of lilypads. Spread them out using the flex-wrap property, which accepts the following values:

- nowrap: Every item is fit to a single line.
- wrap: Items wrap around to additional lines.
- wrap-reverse: Items wrap around to additional lines in reverse.

```
#pond {
  display: flex;
  flex-wrap:wrap
}
```

Question No:- 19



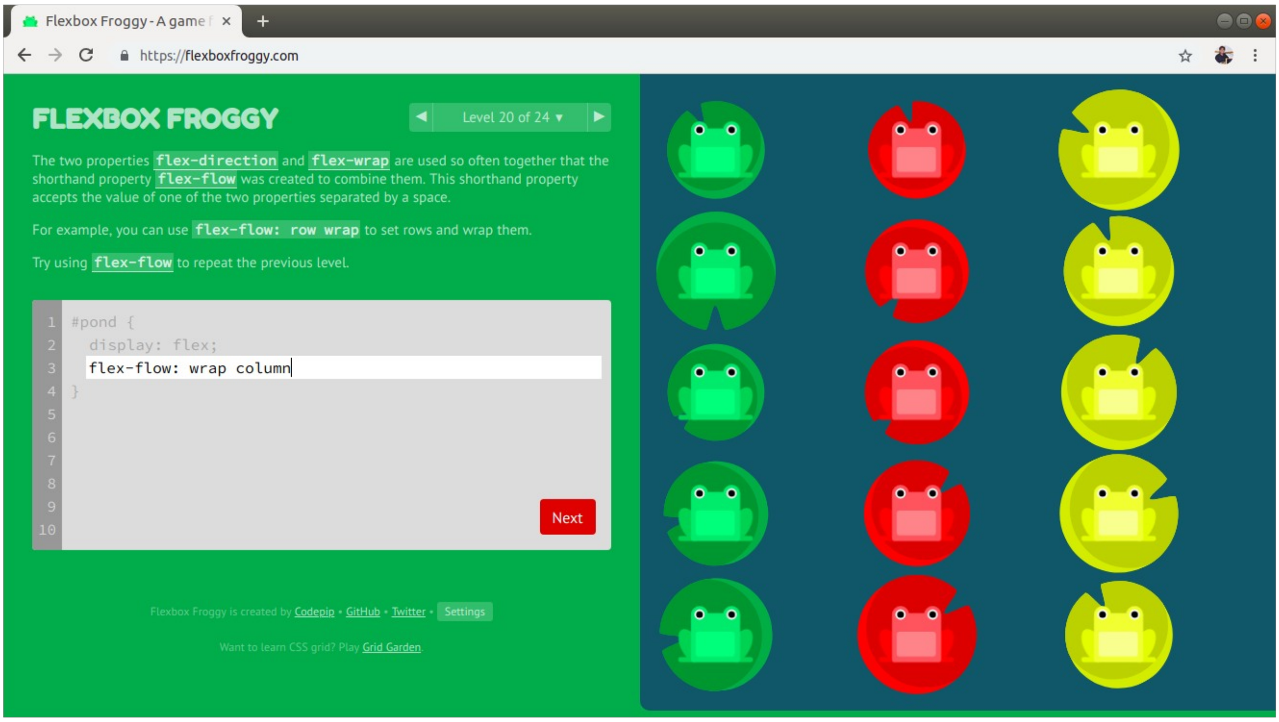
Help this army of frogs form three orderly columns using a combination of `flex-direction` and `flex-wrap`.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

Question No:- 20



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 one of the two properties separated by a space.

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

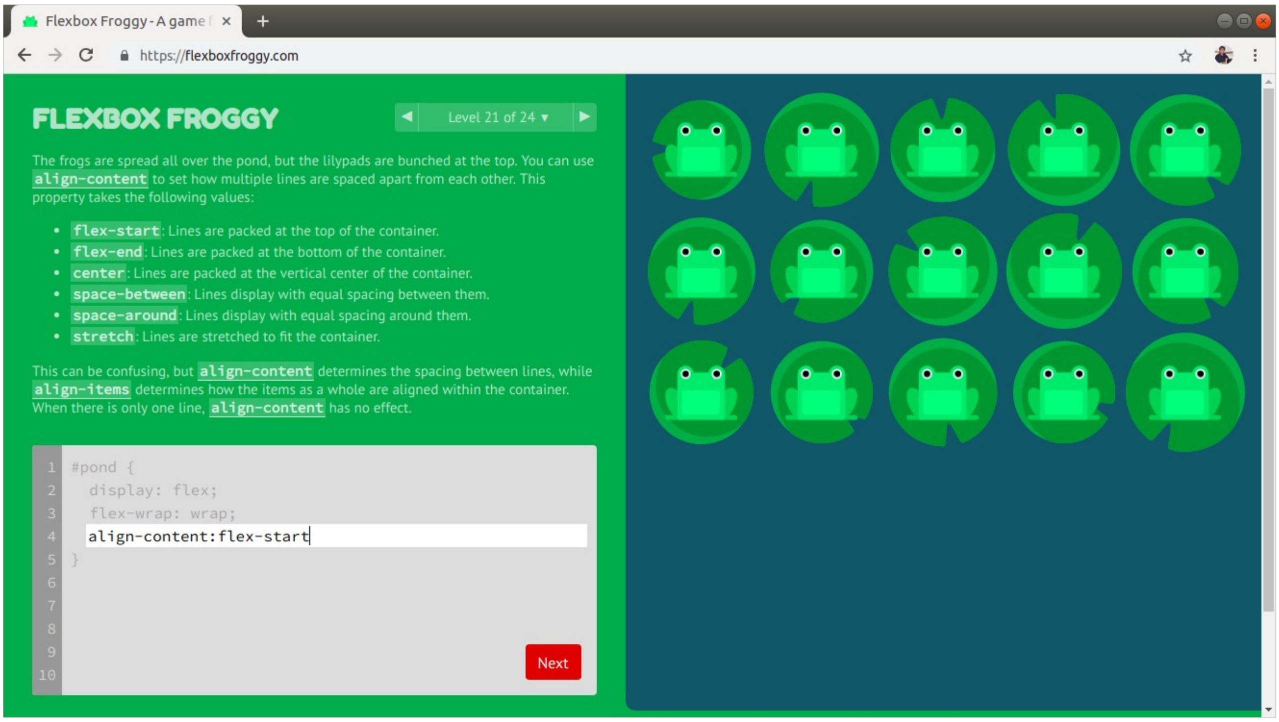
Try using `flex-flow` to repeat the previous level.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden.

Question No:- 21



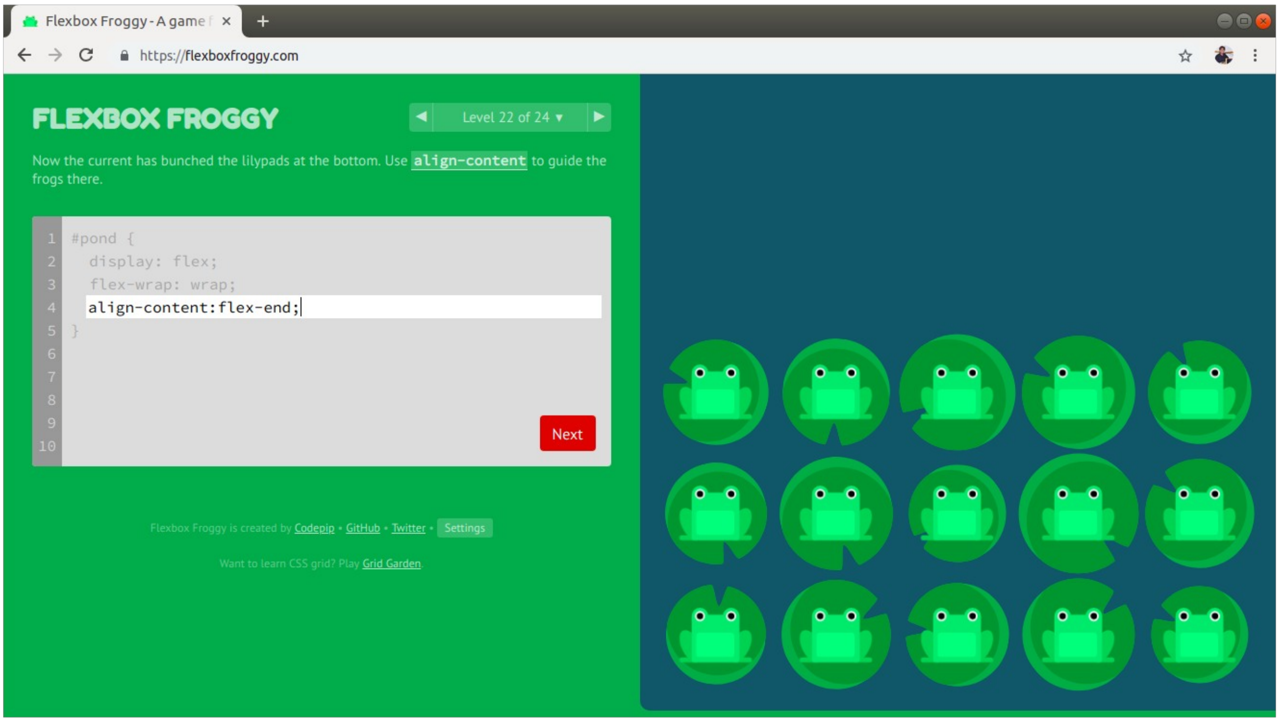
The frogs are spread all over the pond, but the lilypads are bunched at the top. You can use `align-content` to set how multiple lines are spaced apart from each other. This property takes the following values:

- `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.

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

Question No:- 22



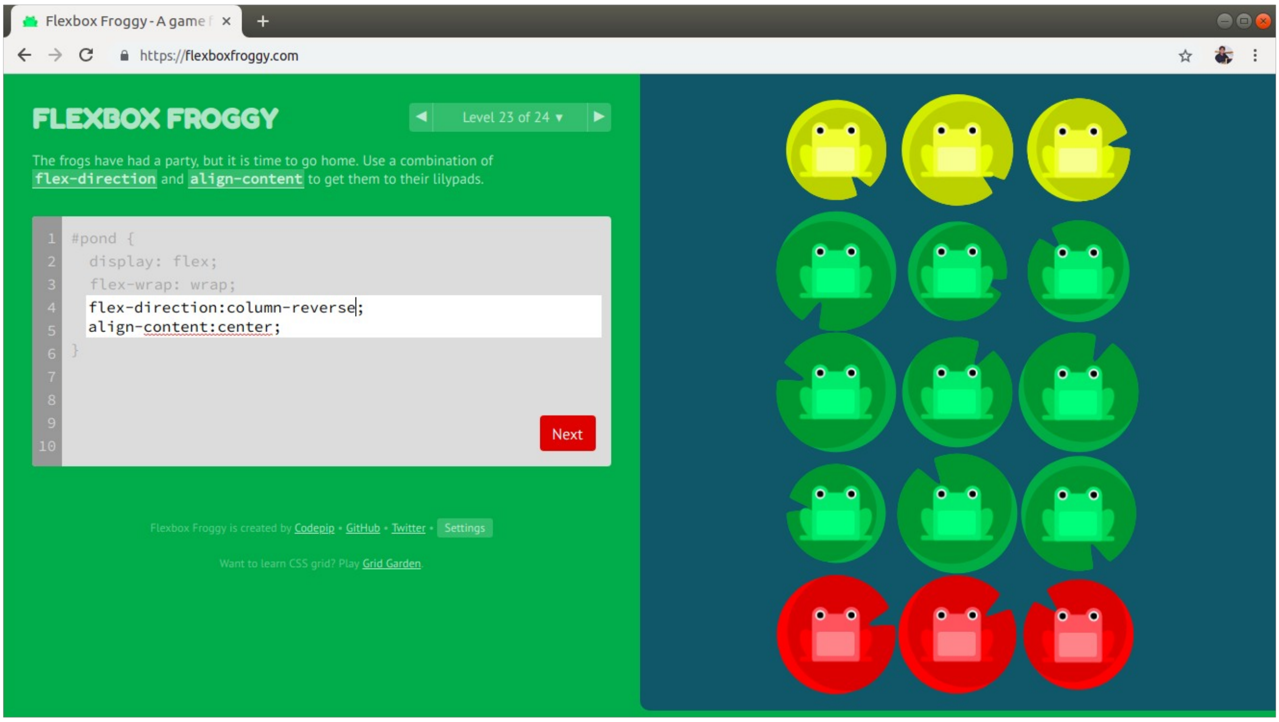
Now the current has bunched the lilypads at the bottom. Use `align-content` to guide the frogs there.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden

Question No:- 23



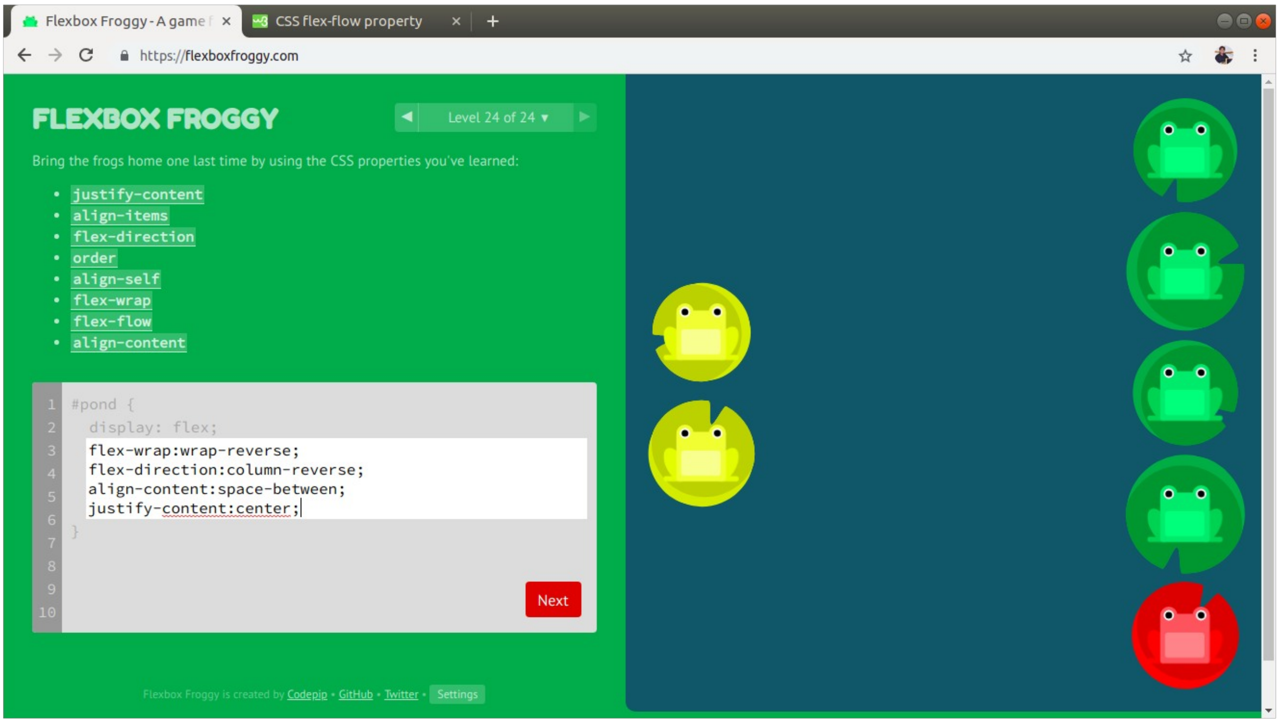
The frogs have had a party, but it is time to go home. Use a combination of `flex-direction` and `align-content` to get them to their lilypads.

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings

Want to learn CSS grid? Play Grid Garden

Question No:- 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`

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

Flexbox Froggy is created by Codepip • GitHub • Twitter • Settings