

FLEXBOX ASSIGNMENT

Name: Aman Kumar

Batch: Full Stack Development (May Batch) – 2024

Assignment: FlexBox Froggy

Screenshots:

FLEXBOX FROGGY Level 1 of 24

Welcome to Flexbox Froggy, a game where you help Froggy across the pond by writing CSS code! Guide this frog to the lily pad 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 Froggy to the right.

```
1 #pond {
2   display: flex;
3   justify-content: flex-end;
4 }
```

Next

Flexbox Froggy is created by Codecademy • YouTube • Twitter • GitHub • Settings

Want to learn CSS grid? Play Grid Garden.

FLEXBOX FROGGY Level 2 of 24

Use `justify-content` again to help these frogs get to the lily pads in the center of the pond. What 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.

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

Next

Flexbox Froggy is created by Codecademy • YouTube • Twitter • GitHub • Settings

Want to learn CSS grid? Play Grid Garden.

FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 3 of 24

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




If you find yourself forgetting the possible values for a 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 is created by [Codecademy](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).



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

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 5 of 24

Now use `align-items` to help the frogs get to the bottom of the pond. The `align-items` 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 align 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 CSS grid? Play [Grid Garden](#).



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 CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 7 of 24

The frogs need to cross the pond again, this time for some lilypad. Use a combination of `justify-content` and `align-items` to get the frogs in the right order.

```
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 Codepip • YouTube • Twitter • GitHub • Settings

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX FROGGY

Level 8 of 24

The frogs need to get in the same order as their lilypads using this CSS property defines the direction items are placed in the container. 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 Codepip • YouTube • Twitter • GitHub • Settings

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 9 of 24

Help the frogs find their column of lilypads using `flex-direction`. `flex-direction` defines the direction items are placed in the container, and also the order they are placed in:

- `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 [Codepip](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).

Calculator: 1-24, Reset

Three frogs (green, yellow, red) are shown on the right, each with a lilypad icon.

FLEXBOX FROGGY

Level 10 of 24

Help the frogs get to their own lilypads. Although they seem to be in the right order, you need to use `flex-direction` and `justify-content` to get them to the right lilypad.

Notice that when you set the direction to a reversed row or column, the order of items is also reversed.

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).

Calculator: 1-24, Reset

Three frogs (red, yellow, green) are shown on the right, each with a lilypad icon.

FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 11 of 24

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

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

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



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 [Codepip](#) • [YouTube](#) • [Twitter](#) • [GitHub](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 13 of 24

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

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX FROGGY

Level 14 of 24

Sometimes reversing the row or column order of a container, we can apply the `order` property to individual items. By default, items are ordered from 0 to 1, 2, 3, etc. If you set `order` to a negative number, it will move the item to the beginning of the container. If you set it to a positive number, it will move the item to the end of the container.

Use the `order` property to reorder the frogs according to the instructions.

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

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: -3;
7 }
8
9
10
```

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX FROGGY

Level 16 of 24

Another property you can apply to individual items is `align`. It has the same values as `align-items` and its value for the specified axis.

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Combine `order` with `align-self` to help the frogs to the

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

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

Want to learn CSS grid? Play [Grid Garden](#).

Level 17 of 24

Reset

Next



FLEXBOX FROGGY

Oh no! The frogs are all squeezed onto a single row of lily pads. Use 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.

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

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

Want to learn CSS grid? Play [Grid Garden](#).

Level 18 of 24

Reset

Next



FLEXBOX ASSIGNMENT

FLEXBOX FROGGY

Level 19 of 24

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

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

Next

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

Want to learn CSS grid? Play [Grid Garden](#).



FLEXBOX FROGGY

Level 20 of 24

The two properties `flex-direction` and `flex-wrap` are combined in the shorthand property `flex-flow`. `flex-flow` was created to combine the two properties into one. The shorthand property `flex-flow` accepts the value of the two properties separated by a space. For example, you can use `flex-flow: row wrap` to set `flex-direction: row` and `flex-wrap: wrap`.

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 CSS grid? Play [Grid Garden](#).



FLEXBOX FROGGY

Level 21 of 24

The frogs are spread all over the pond, but the lilypads are all at the bottom. Use `align-content` to set how multiple lines are spaced. 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 how the container is aligned, 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-content: flex-start;
5 }
6
7
8
9
10
```

Next

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

[Want to learn CSS grid? Play Grid Garden.](#)



FLEXBOX FROGGY

Level 22 of 24

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

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

Next

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

[Want to learn CSS grid? Play Grid Garden.](#)



FLEXBOX ASSIGNMENT

