

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Домашняя работа 1

Выполнил:

Мосин З. И.  
Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Пройти три игры:

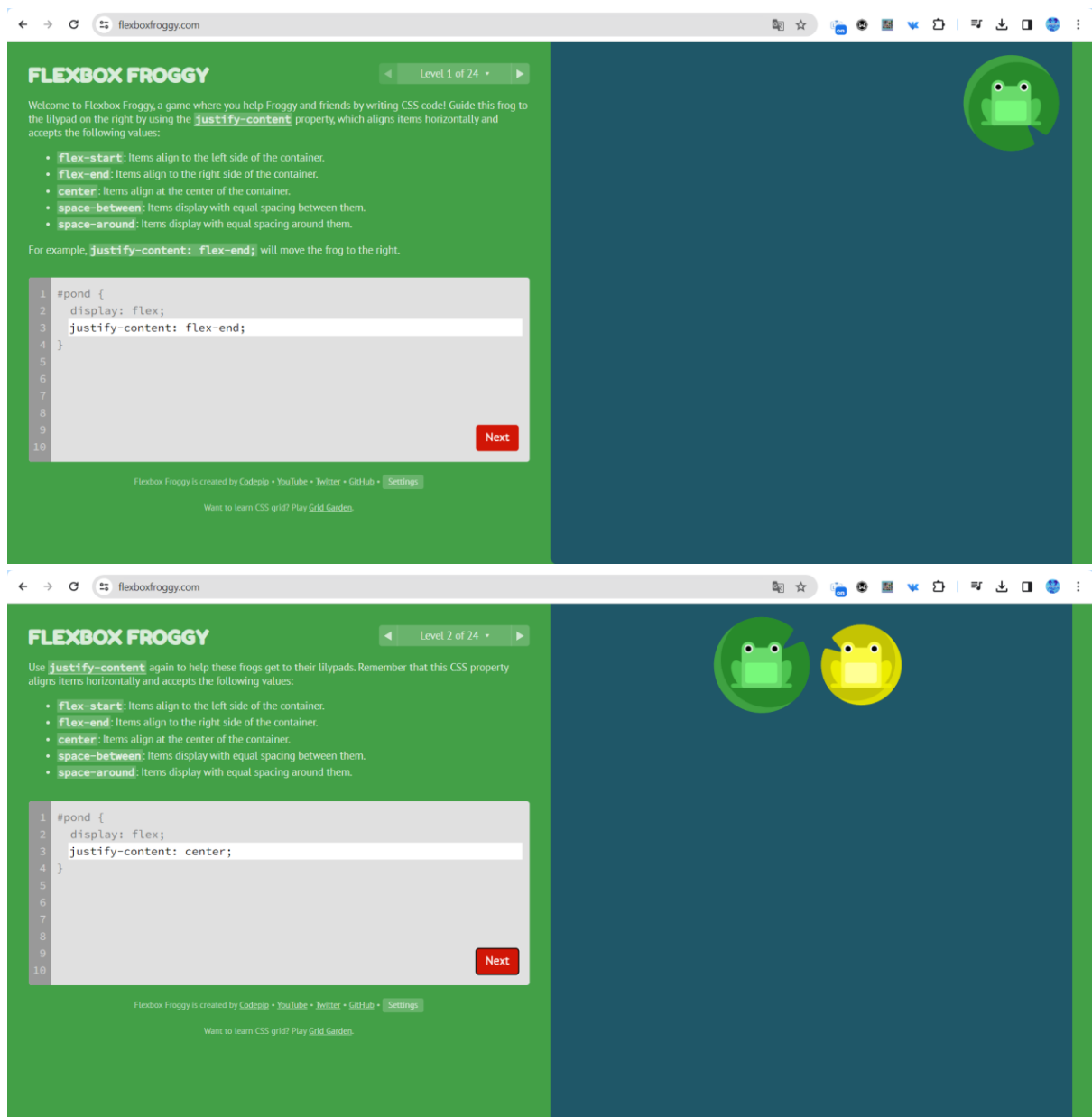
<https://flexboxfroggy.com/#ru>

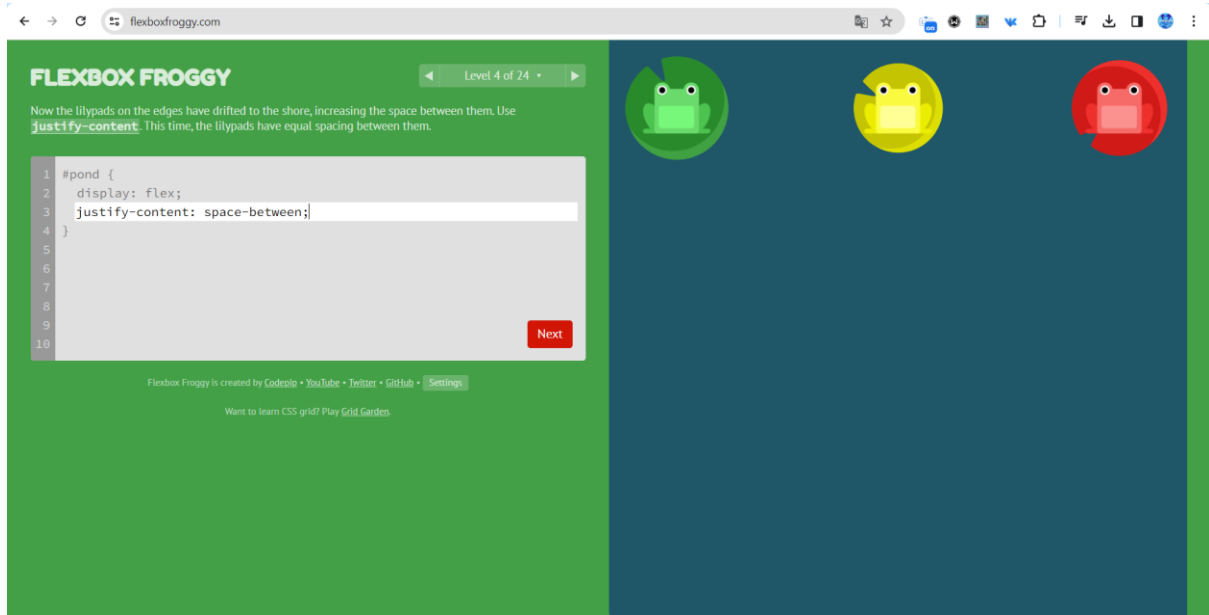
<https://cssgridgarden.com/#ru>

[https://learngitbranching.js.org/?locale=ru\\_RU](https://learngitbranching.js.org/?locale=ru_RU) (основы работы с git, достаточно выполнить первые 4 урока)

## Ход работы

Были пройдены три игры.







flexboxfroggy.com

Level 7 of 24

# FLEXBOX FROGGY


The frogs need to cross the pond again, this time for some lilypads 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 CSS grid? Play [Grid Garden](#).



flexboxfroggy.com

Level 8 of 24

# FLEXBOX FROGGY

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.

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





flexboxfroggy.com

Level 11 of 24

# FLEXBOX FROGGY

Help the frogs find their lily pads 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.

```
1 #pond {
2   display: flex;
3   flex-direction: column;
4   justify-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](#).



flexboxfroggy.com

Level 12 of 24

# FLEXBOX FROGGY




Help the frogs find their lily pads 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 CSS grid? Play [Grid Garden](#).



←

→

↻

flexboxfroggy.com

🔍

☆

🌐

🔗

📄

📱

🔧

🌐

⋮

# 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   flex-direction: row-reverse;
5   align-items: flex-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](#).



←

→

↻

flexboxfroggy.com

🔍

☆

🌐

🔗

📄

📱

🔧

🌐

⋮

# 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
```

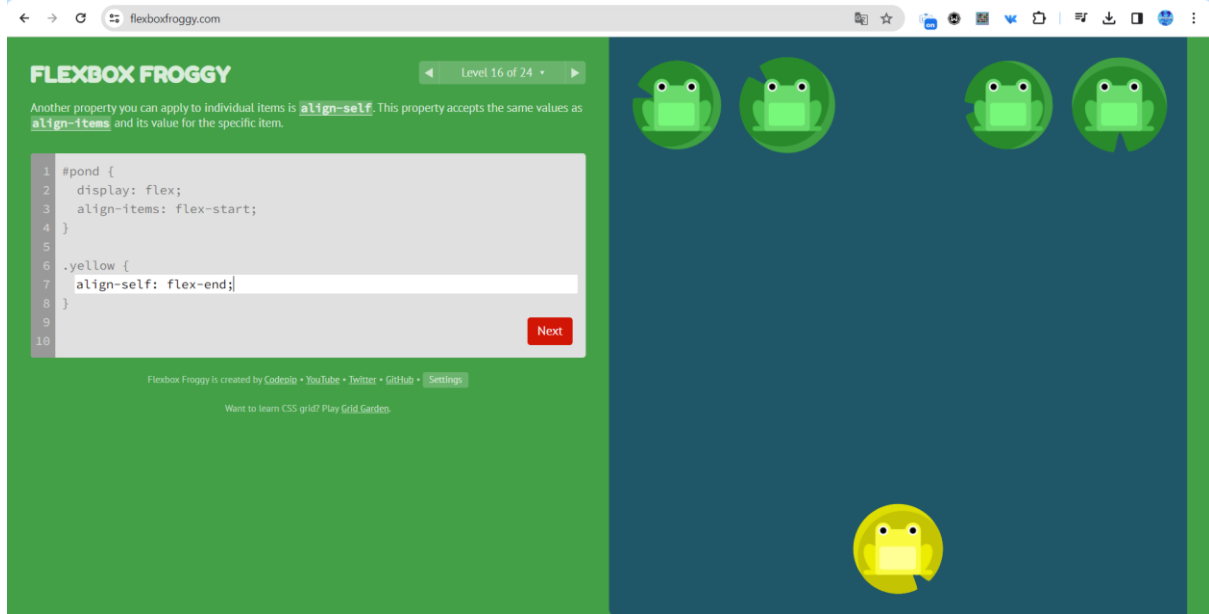
Next

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

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







flexboxfroggy.com

Level 17 of 24

# FLEXBOX FROGGY


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


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

Next

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

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





flexboxfroggy.com

Level 18 of 24

# FLEXBOX FROGGY

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


```
1 #pond {
2   display: flex;
3   flex-wrap: 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](#).





flexboxfroggy.com

Level 19 of 24

# FLEXBOX FROGGY


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](#).









































flexboxfroggy.com

Level 20 of 24

# FLEXBOX FROGGY

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](#)

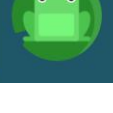
Want to learn CSS grid? [Play Grid Garden](#).









































flexboxfroggy.com

Level 21 of 24

## FLEXBOX FROGGY

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-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](#).



flexboxfroggy.com

Level 22 of 24

## FLEXBOX FROGGY


Now the current has bunched the lily pads at the bottom. Use **align-content** to guide the frogs there.

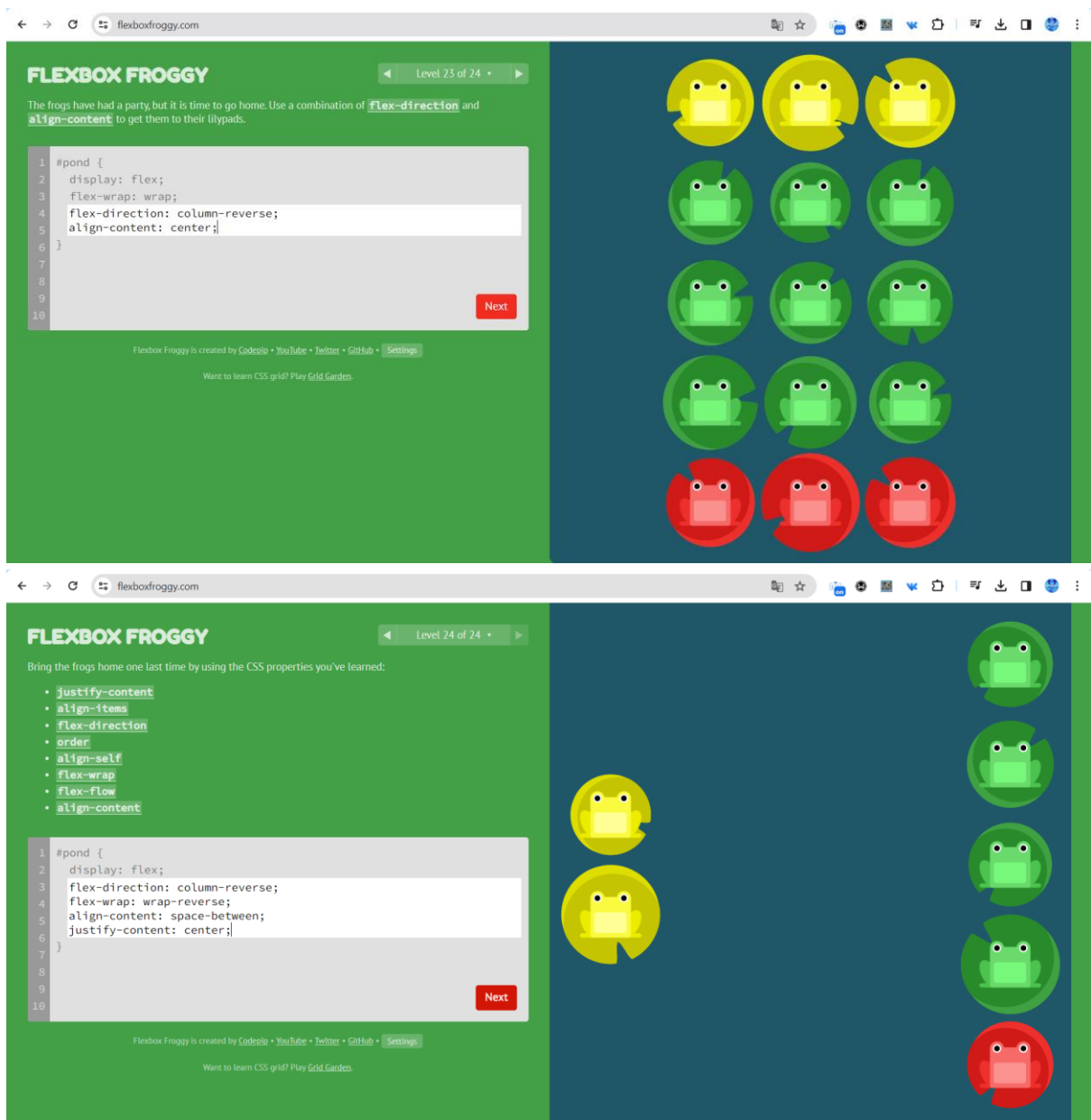
```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   align-content: flex-end;
5 }
6
7
8
9
10
```

Next

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

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





## Вывод

В ходе работы над первым домашним заданием я изучил FlexBox, Grid, освоил стили, а также углубил значения для взаимодействия с git.