САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронд-энд разработка

Отчет

Практическая работа 1 на тему: "Знакомство с CSS Grid, Flexbox"

Выполнил: Безруков А.И. (К33402)

Проверил: Добряков Д.И.

Санкт-Петербург, 2022 г.

Задача

Познакомиться с основами CSS путем прохождения двух обучающих игр - Grid Garden, Flexbox Froggy

Ход работы

Grid garden

```
GRID GARDEN

Добро пожаловать в Grid Garden — место, где вы напишете CSS-код, чтобы вырастить морковный сад. Поливайте только те зоны, на которых есть морковь, используя свойство grid—column—start:

Например, grid—column—start: 3 покроет водой зону, начинающуюся на третьей grid-линии по вертикали, это как сказать "третья вертикальная граница на grid—сетке слева".

#garden {
display: grid;
grid—template—columns: 20% 20% 20% 20% 20%;
grid—template—rows: 20% 20% 20% 20% 20%;

#water {
grid—column—start: 3
}

##water {
grid—column—start: 3
}

##wate
```

```
GRID GARDEN

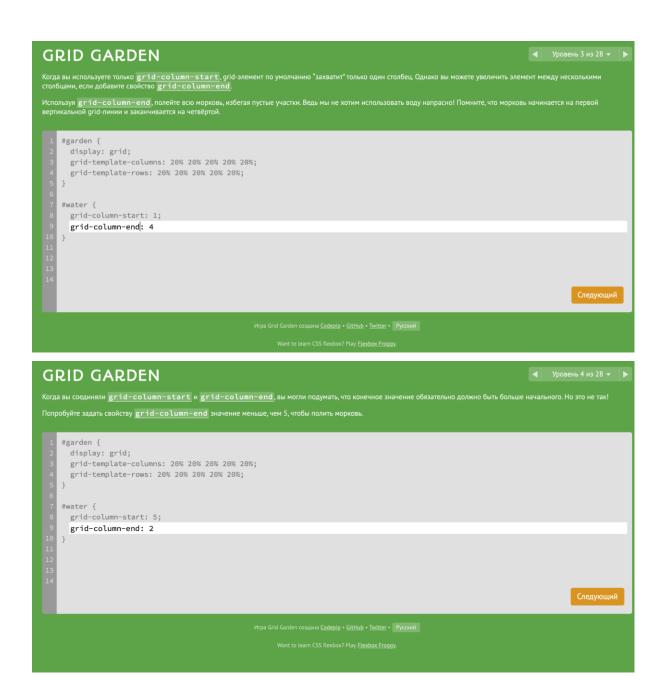
Oy, кажется, сорняки растут прямо в углу сада. Используйте grid-column-start, чтобы отравить их. Помните: сорняки начинаются на пятой вертикальной grid-линии.

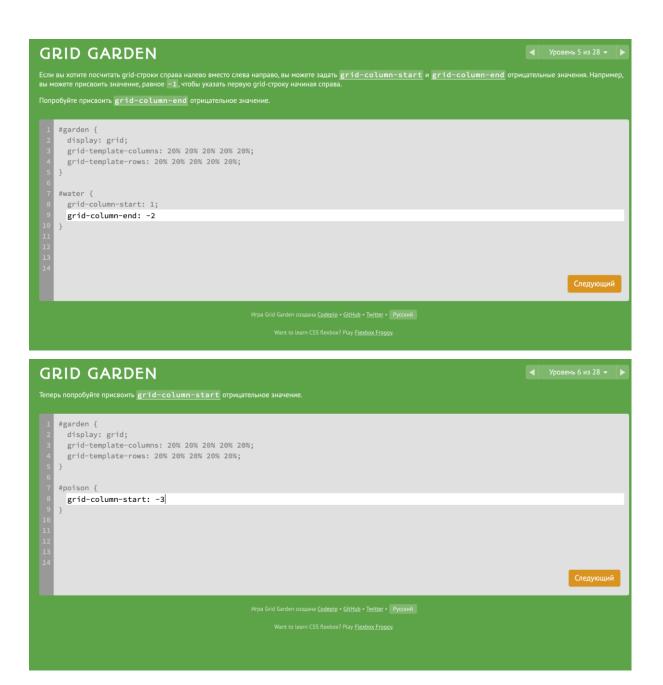
#garden {
    display: grid;
    grid-template-columns: 20% 20% 20% 20% 20%;
    grid-template-rows: 20% 20% 20% 20% 20%;
    }

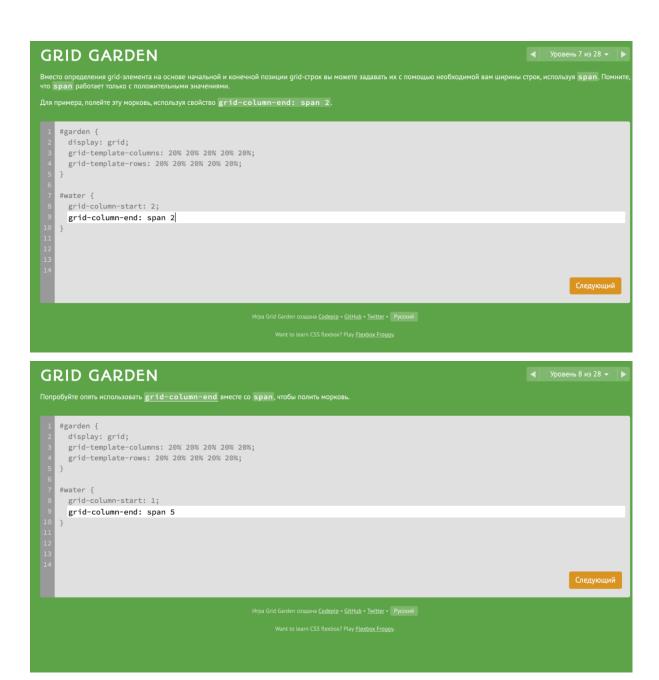
#poison {
    grid-column-start: 5 |
    }

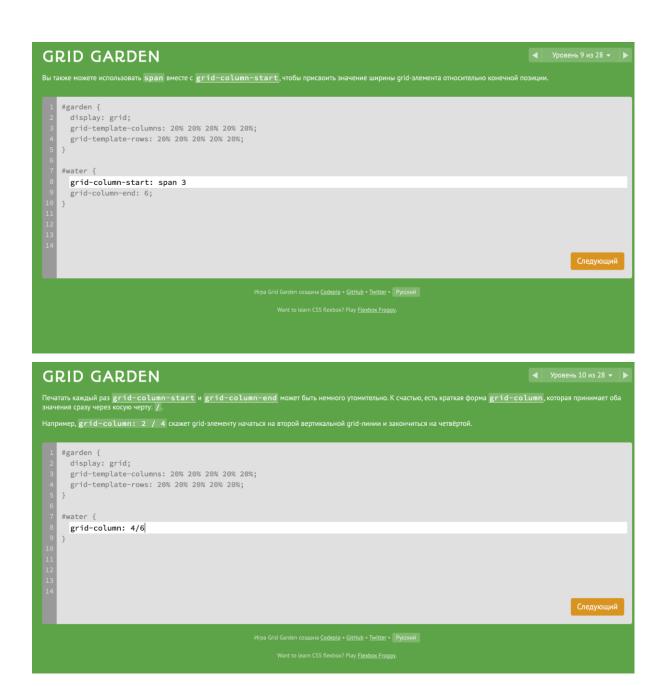
    Vrpa Grid Garden создана Себерір • GitHub • Tmitter • Русский

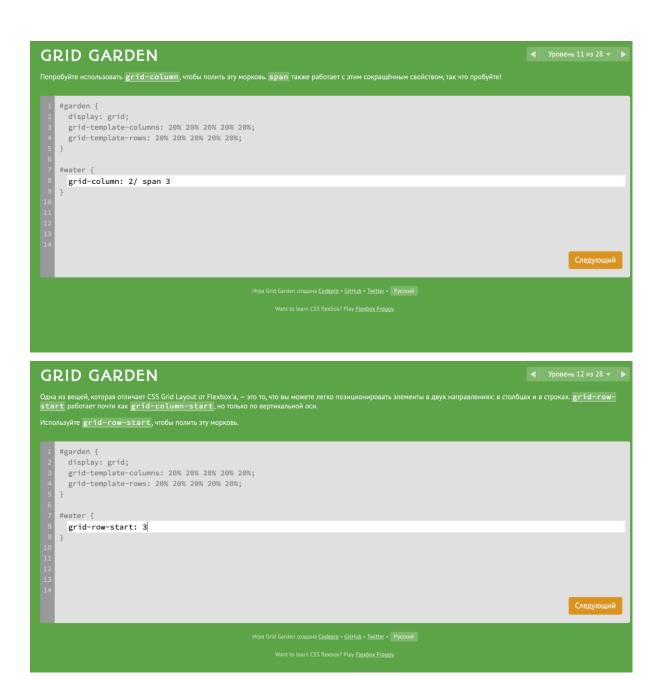
    Want to learn CSS fierboox? Play Elesbox Froagy.
```

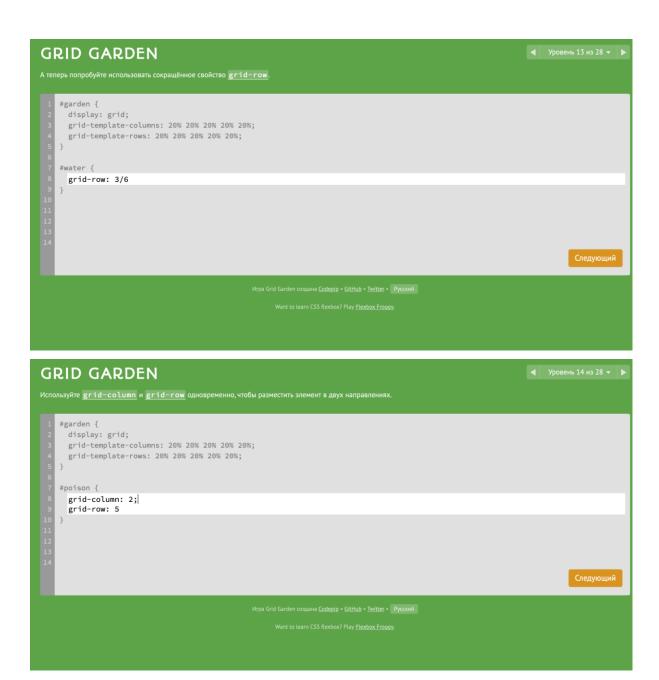


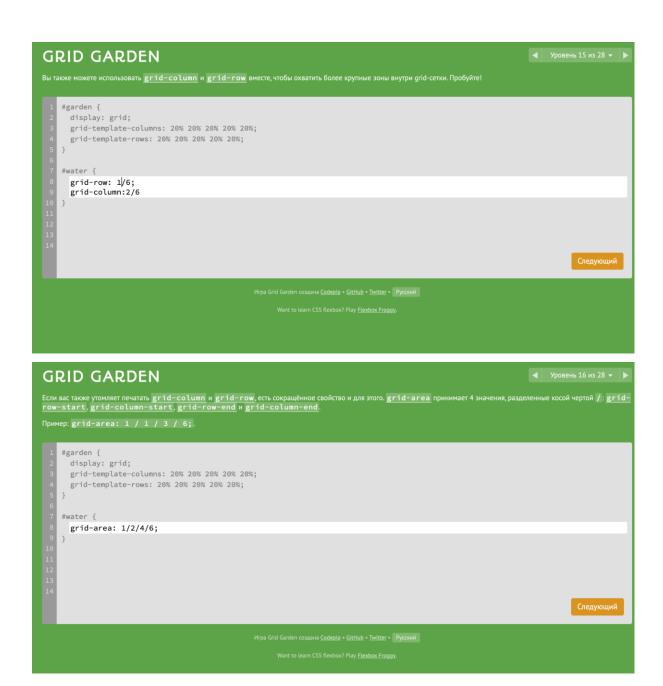


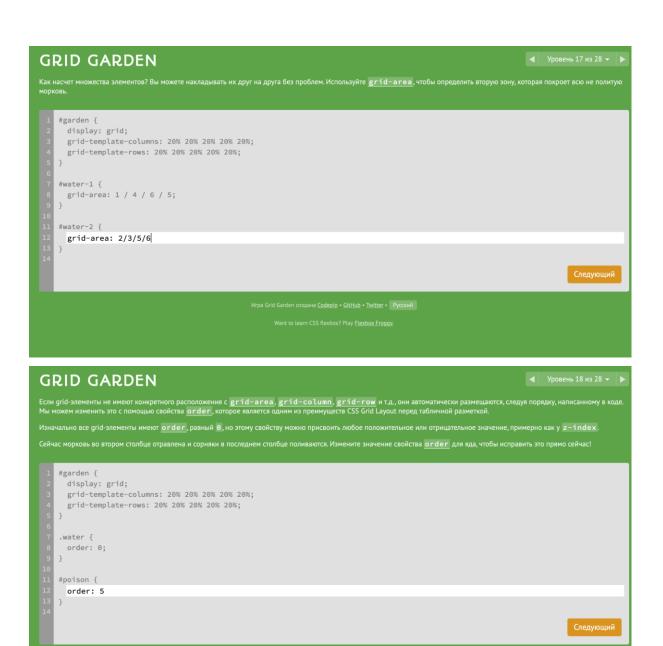


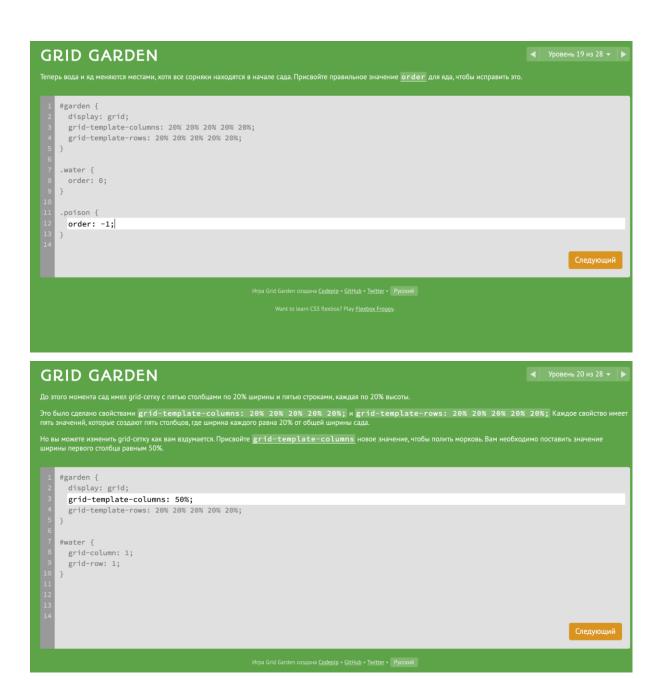


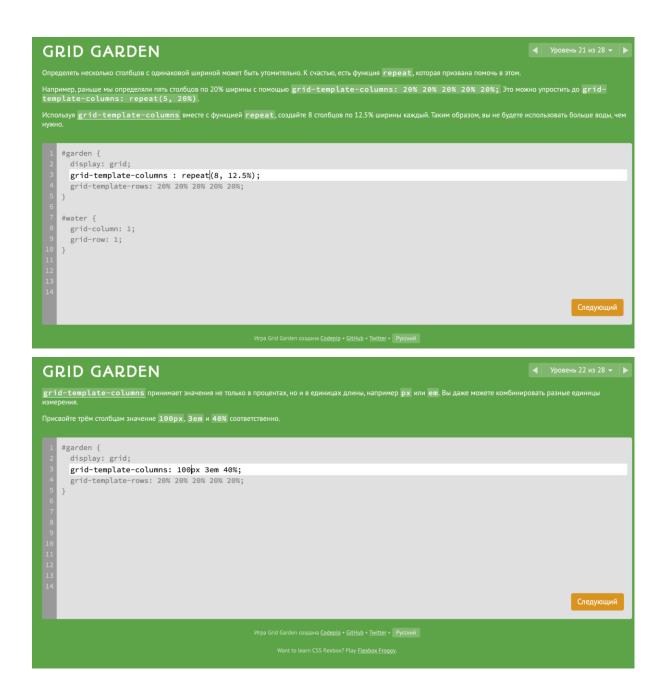


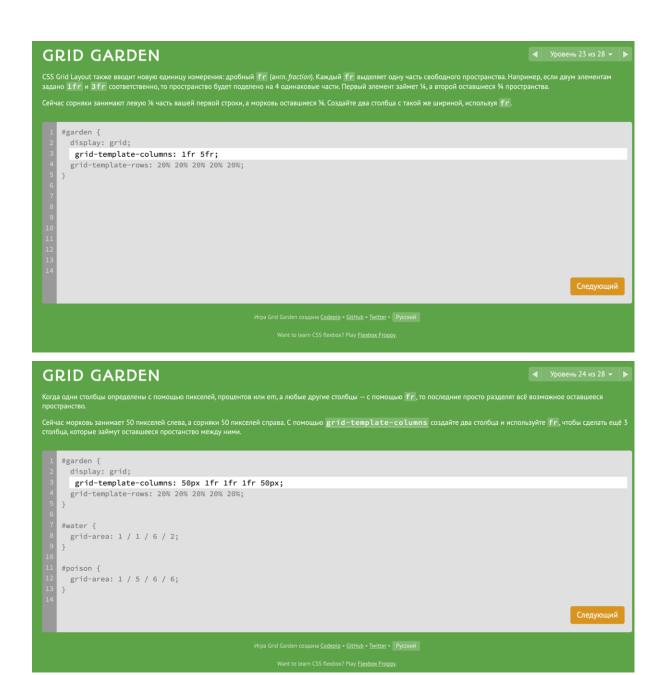


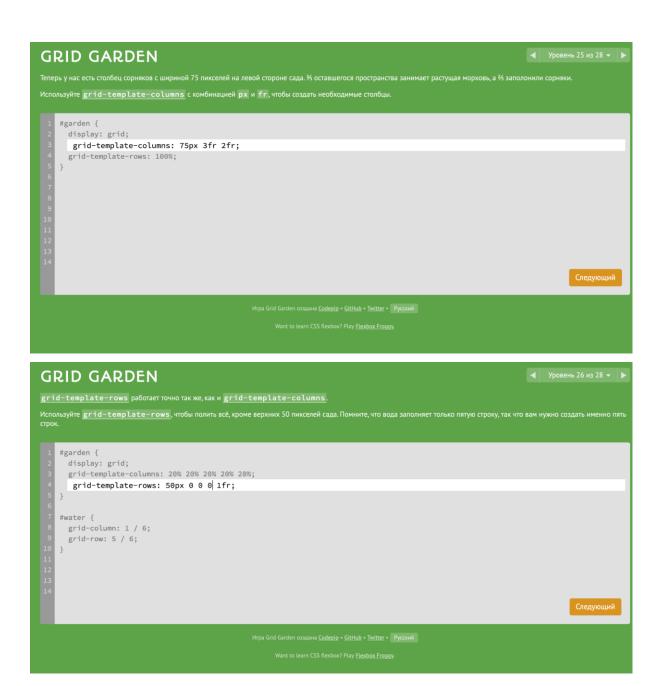






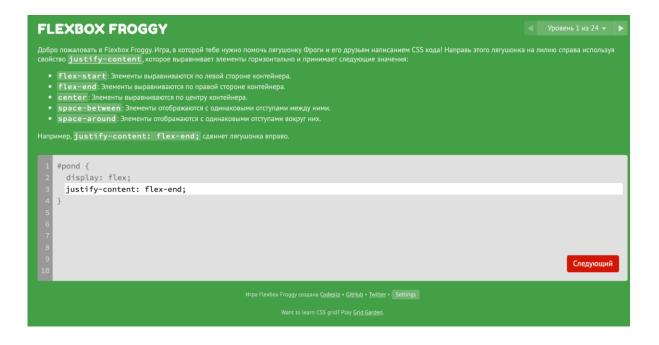






```
GRID GARDEN
grid-template — сокращённый вариант комбинации grid-template-rows и grid-template-columns.
   #garden {
     display: grid;
      grid-template: 60% 40%/ 200px;
   #water {
  grid-column: 1;
   grid-row: 1;
GRID GARDEN
   #garden {
     display: grid;
     grid-template: 1fr 50px / 1fr 4fr;
```

Flexbox Froggy



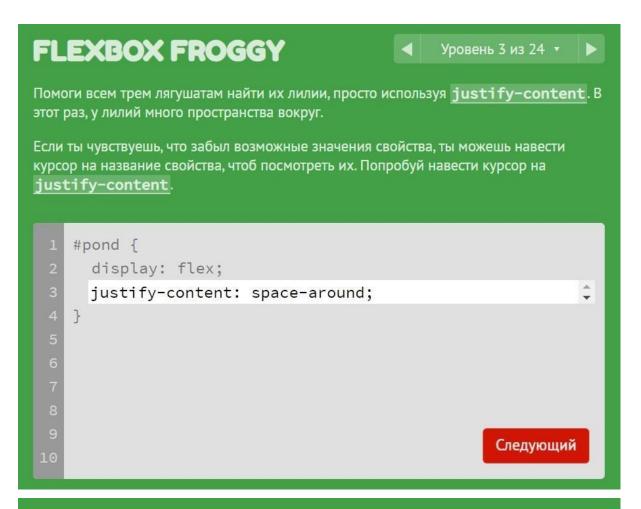


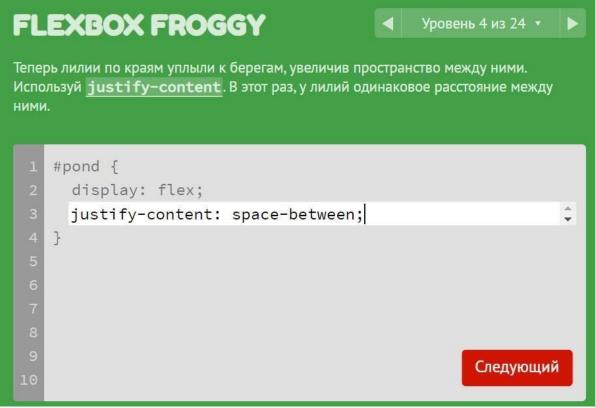
Используй **justify-content** еще раз, чтоб помочь этим лягушатам попасть на их лилии. Помни, что это свойство CSS выравнивает элементы горизонтально и принимает следующие значения:

- flex-start: Элементы выравниваются по левой стороне контейнера.
- flex-end: Элементы выравниваются по правой стороне контейнера.
- center: Элементы выравниваются по центру контейнера.
- space-between: Элементы отображаются с одинаковыми отступами между ними.
- space-around: Элементы отображаются с одинаковыми отступами вокруг

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

Cледующий
```





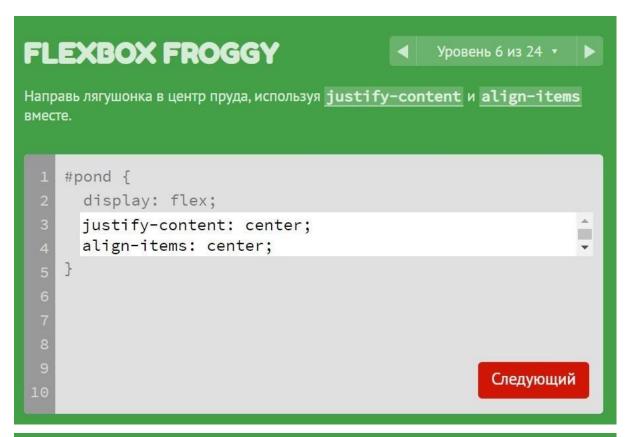


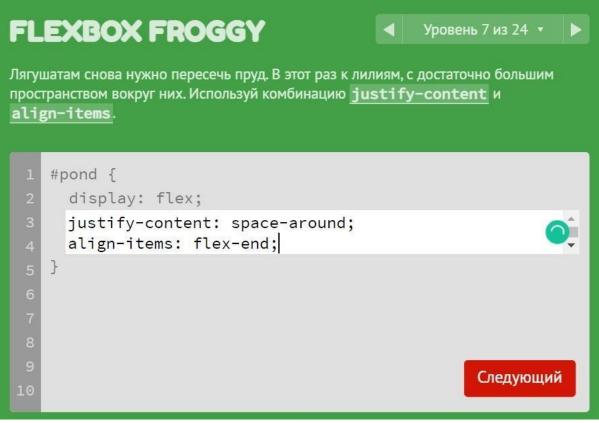
Теперь используй align-items чтоб помочь лягушатам добратся к нижней части пруда. Это CSS свойство выравнивает элементы вертикально и принимает следующие значения:

- flex-start: Элементы выравниваются по верхнему краю контейнера.
- flex-end: Элементы выравниваются по нижнему краю контейнера.
- center: Элементы выравниваются вертикально по центру контейнера.
- baseline: Элементы отображаются на базовой линии контейнера.
- stretch: Элементы растягиваются, чтоб заполнить контейнер.

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

Cледующий
```

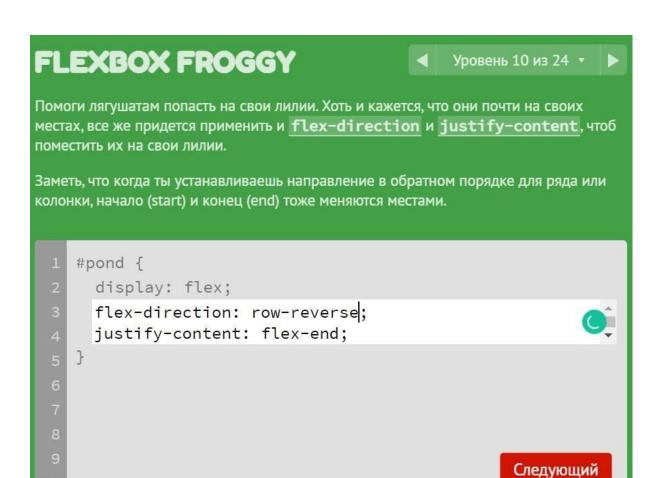


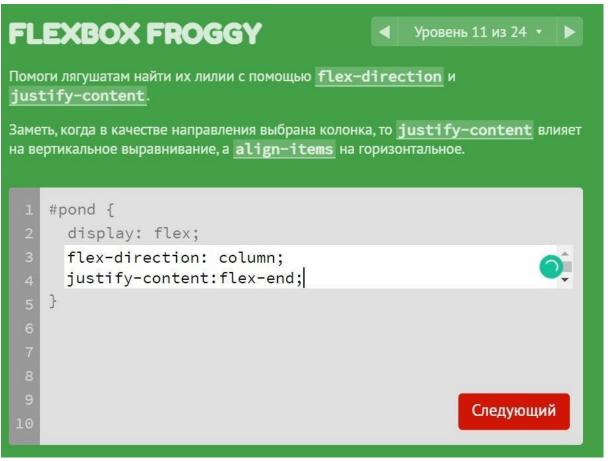


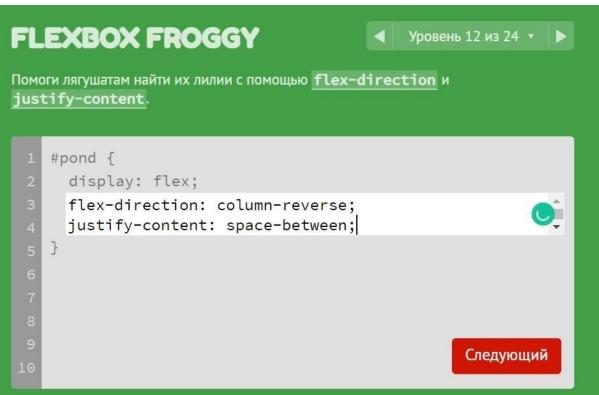
FLEXBOX FROGGY Лягушатам нужно выстроиться в порядке их лилий, используя flex-direction. Это CSS свойство задает направление, в котором будут расположены элементы в контейнере, и принимает следующие значения: • row: элементы размещаются по направлению текста. • row-reverse: элементы отображаются в обратном порядке к направлению текста. • column: элементы располагаются сверху вниз. • column-reverse: элементы располагаются снизу вверх. 1 #pond { 2 display: flex; flex-direction: row-reverse; } 5 6 7 8 9 10

FLEXBOX FROGGY ✓ Уровень 9 из 24 ▼ Помоги лягушатам расположиться на своих лилиях используя flex-direction. Это CSS свойство задает направление, в котором будут расположены элементы в контейнере и принимает следующие значения: • row: Элементы размещаются по направлению текста. • row-reverse: Элементы отображаются в обратном порядке к направлению текста. • column: Элементы распологаются сверху вниз. • column-reverse: Элементы распологаются снизу вверх. 1 #pond { 2 display: flex; 3 flex-direction: column; 4 } 5 6 7 8

Следующий







```
FLEXBOX FROGGY

Помоги лягушатам найти их лилии с помощью flex-direction, justify-content и align-items.

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

Следующий

Спедующий
```

◀ Уровень 14 из 24

Иногда изменения порядка отображения элементов в контейнере недостаточно. В таких случаях мы можем применить свойство <u>order</u> для конкретных элементов. По умолчанию, значение этого свойства у элементов равно 0, но мы можем задать положительное или отрицательное целое число этому свойству.

Используй свойство order, чтоб разместить лягушат на своих лилиях.

```
FLEXBOX FROGGY

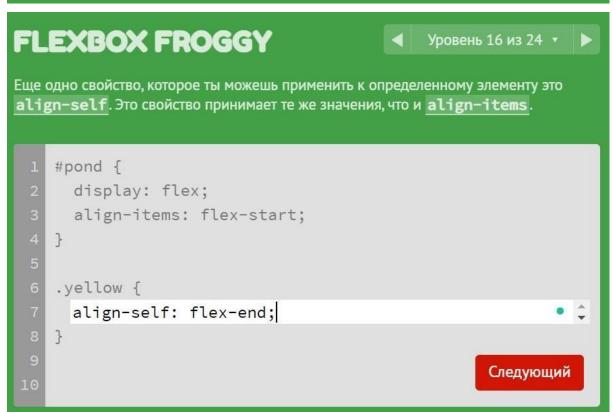
✓ Уровень 15 из 24 ▼

Используй свойство order, чтоб отправить красного лягушонка на его лилию.

1 #pond {
    display: flex;
    }

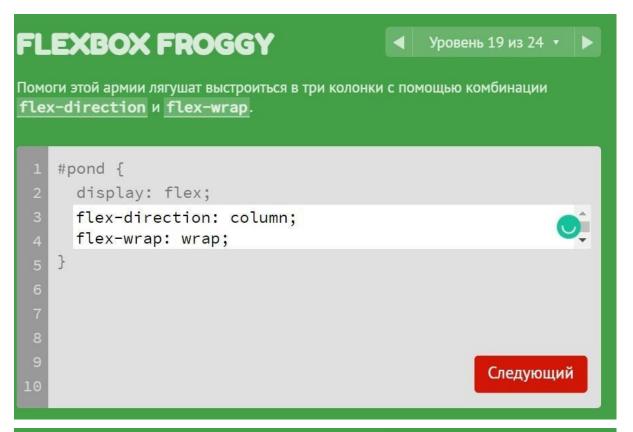
4 .red {
    order:-2;
    }

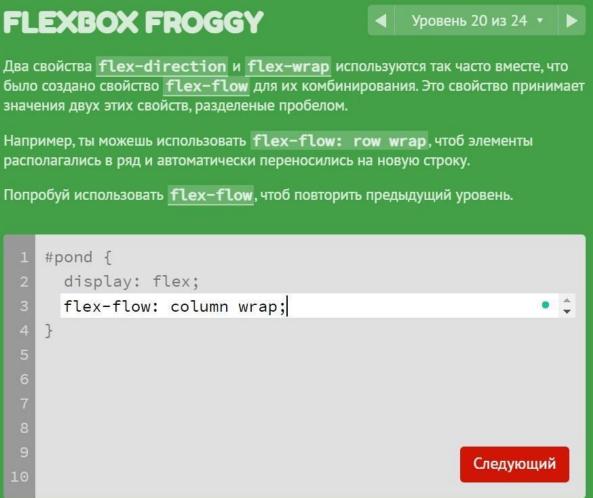
Следующий
```











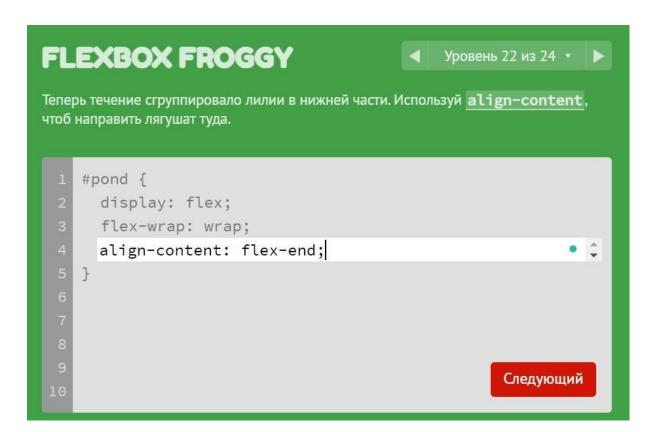


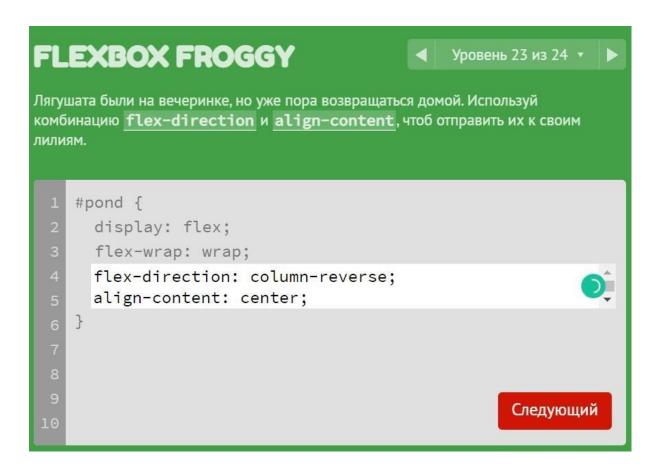
Лягушат раскидало по всему пруду, но лилии сгруппированы в верхней части. Ты можешь использовать align-content, чтобы указать, как несколько рядов должны отделяться друг от друга. Данное свойство принимает следующие значения:

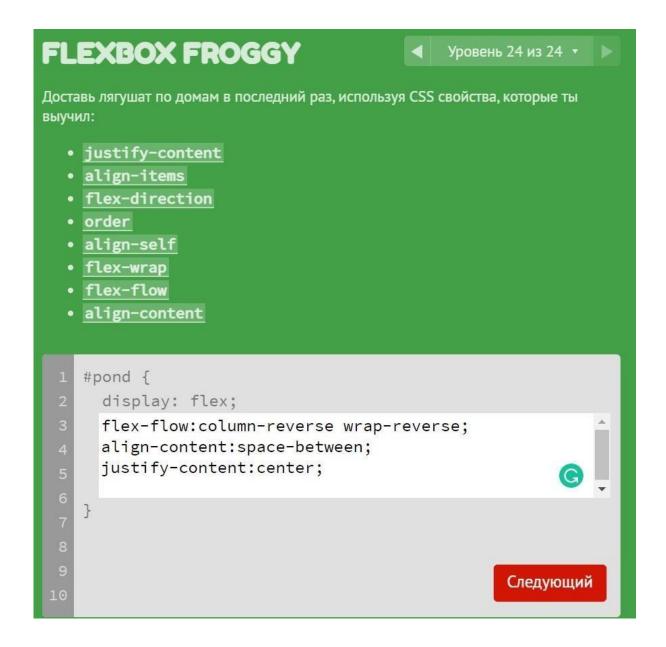
- flex-start: Ряды группируются в верхней части контейнера.
- flex-end: Ряды группируются в нижней части контейнера.
- center: Ряды группируются вертикально по центру контейнера.
- space-between: Ряды отображаются с одинаковыми расстояниями между ними.
- space-around: Ряды отображаются с одинаковыми расстояниями вокруг них.
- stretch: Ряды растягиваются, чтоб заполнить контейнер равномерно.

Это может запутать, но align-content отвечает за расстояние между рядами, в то время как align-items отвечает за то, как элементы в целом будут выровнены в контейнере. Когда только один ряд, align-content ни на что не влияет.

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   align-content: flex-start;
5 }
```







Вывод: Grid и Flexbox - два современных стандарта для идеальной CSS-вёрстки. Flexbox чаще работает в одном измерении, а Grid задействует сразу два ряды и колонки. Благодаря этим играм я убедился в удобстве данных технологий.