

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Практическая работа 1

Выполнила:

Герулайтите Габриэля

К33401

Проверил:

Добряков Д. И.

Санкт-Петербург

2021 г.

Задача

Пройти две игры на изучение CSS Flexbox, CSS Grid.

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

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

Ход работы

1) Flexbox Froggy

Игра, в которой мне нужно помочь лягушонку Фроги и его друзьям написанием CSS кода! Направляю лягушат на лилии, используя свойства:

justify-content, которое 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 них.

align-items – 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.

align-self – для конкретного элемента

align-content – выравнивает ряды flex-контейнера внутри него (работает если элементы расположены более, чем в один ряд). 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.

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.

order – указывает порядок flex-элемента

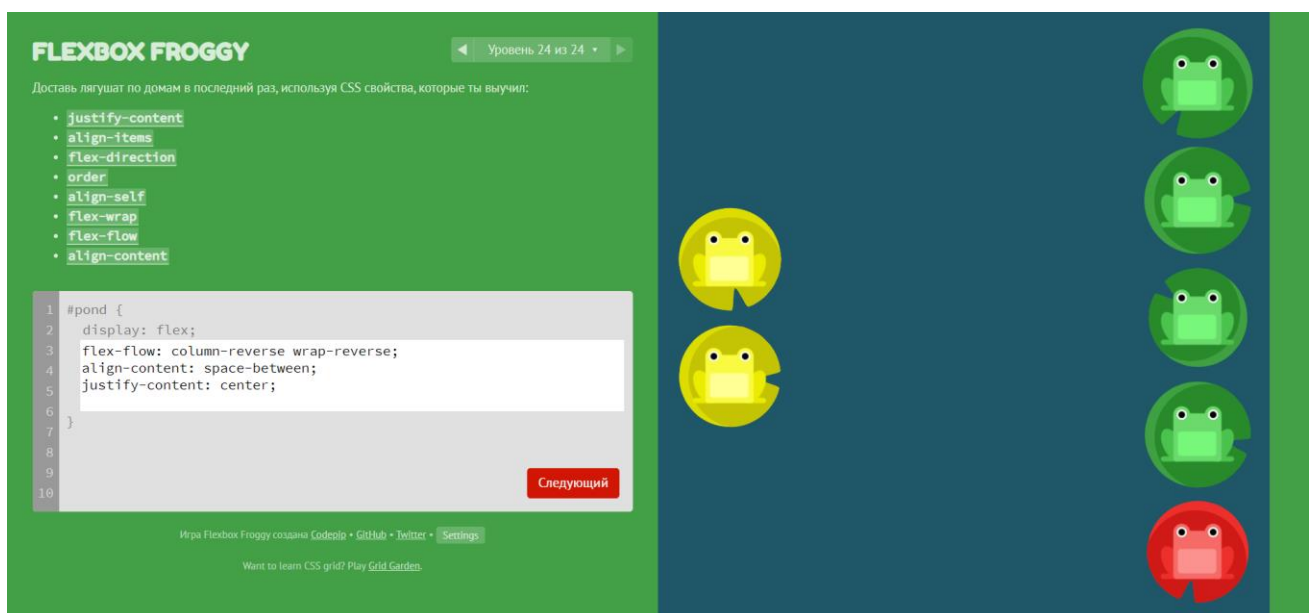
flex-wrap – указывает, нужно ли чтобы элементы принудительно находились в одном ряду или автоматически переносились.

nowrap: Размеры элементов устанавливаются автоматически, чтоб они поместились в один ряд.

wrap: Элементы автоматически переносятся на новую строку.

wrap-reverse: Элементы автоматически переносятся на новую строку, но строки расположены в обратном порядке.

flex-flow – свойство для быстрой записи <flex-direction> и <flex-wrap>.



2) Grid Garden.

Это место, где я писала CSS-код, чтобы вырастить морковный сад.

grid-column-start – определяет начальную позицию grid-элемента внутри grid-столбцов

grid-column-end - определяет конечную позицию grid-элемента внутри grid-столбцов

Функция **span** – ширина строк

grid-column – позиция grid-элемента внутри grid-столбцов `<grid-column-start>/<grid-column-end>`

Аналогично со строками **row**.

grid-area – определяет позицию и размер grid-элемента внутри grid-сетки, grid-area принимает 4 значения, разделенные косой чертой `/`: `grid-row-start`, `grid-column-start`, `grid-row-end` и `grid-column-end`.

order – определяет порядок grid-элемента

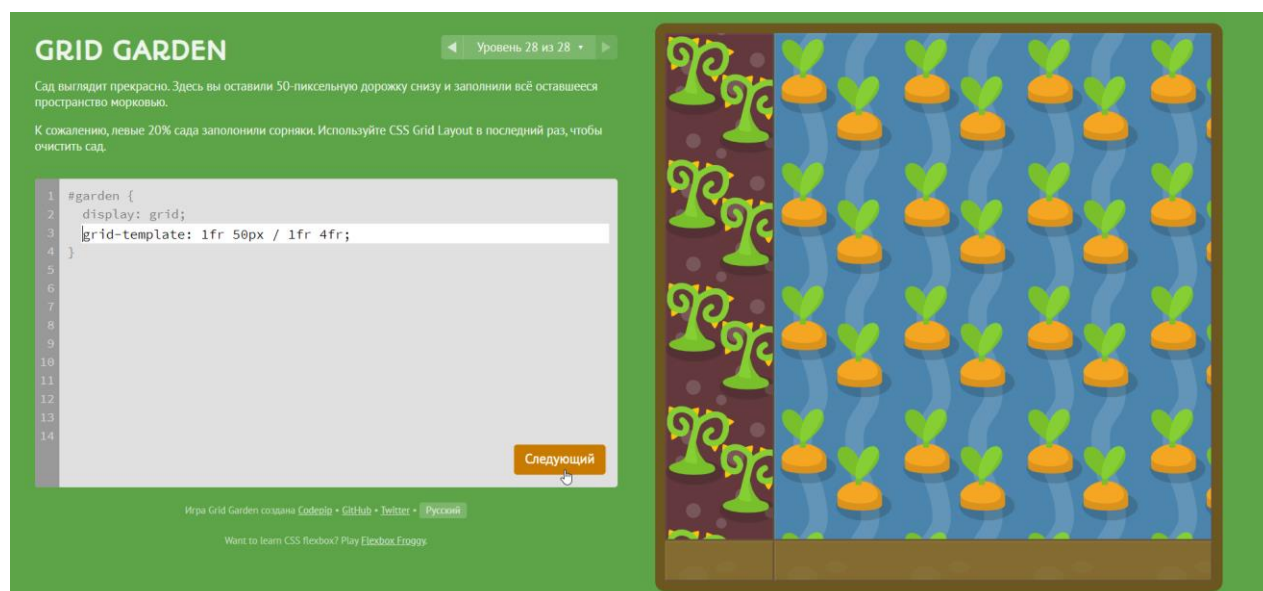
grid-template-columns – размер и названия для grid-столбцов

grid-template-rows – размер и названия для grid-строк

Функция **repeat** – определяет несколько столбцов с одинаковой шириной (пример: `grid-template-columns: repeat(5, 20%)`)

Заметка: можно комбинировать единицы измерения.

grid-template – сокращённый вариант комбинации `grid-template-rows` и `grid-template-columns`.



Вывод

В ходе работы было пройдено две игры и изучена работа CSS Flexbox и CSS Grid.