

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

**Отчет**

Домашняя работа 1: CSS Flexbox, CSS Grid

Выполнила:  
Пронина Александра

Группа K33392

Проверил:  
Добряков Д. И.

Санкт-Петербург

2023 г.

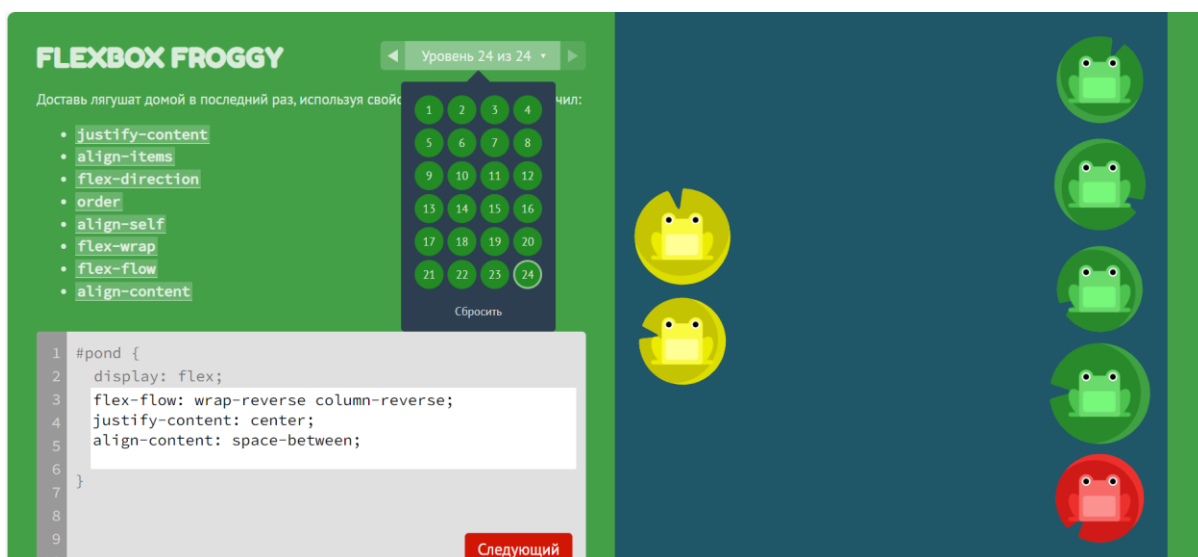
## Задача:

Пройти три игры:

- <https://flexboxfroggy.com/#ru>
- <https://cssgridgarden.com/#ru>
- [https://learnitbranching.js.org/?locale=ru\\_RU](https://learnitbranching.js.org/?locale=ru_RU) (основы работы с git, достаточно выполнить первые 4 урока)

## Ход работы:

### Задание 1.



Свойство `justify-content` выравнивает элементы горизонтально и принимает 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.

Далее мы использовали `align-items`. Это свойство CSS выравнивает элементы по вертикали и принимает 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, чтобы заполнить контейнер.

Flex-direction 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.

Далее мы меняли порядок – order – для лягушек. По умолчанию стоит «0».

Раздвинуть «сплюснутых» лягушек можно с помощью свойства flex-wrap, которое принимает следующие значения:

- nowrap: размеры элементов устанавливаются автоматически, чтобы 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.

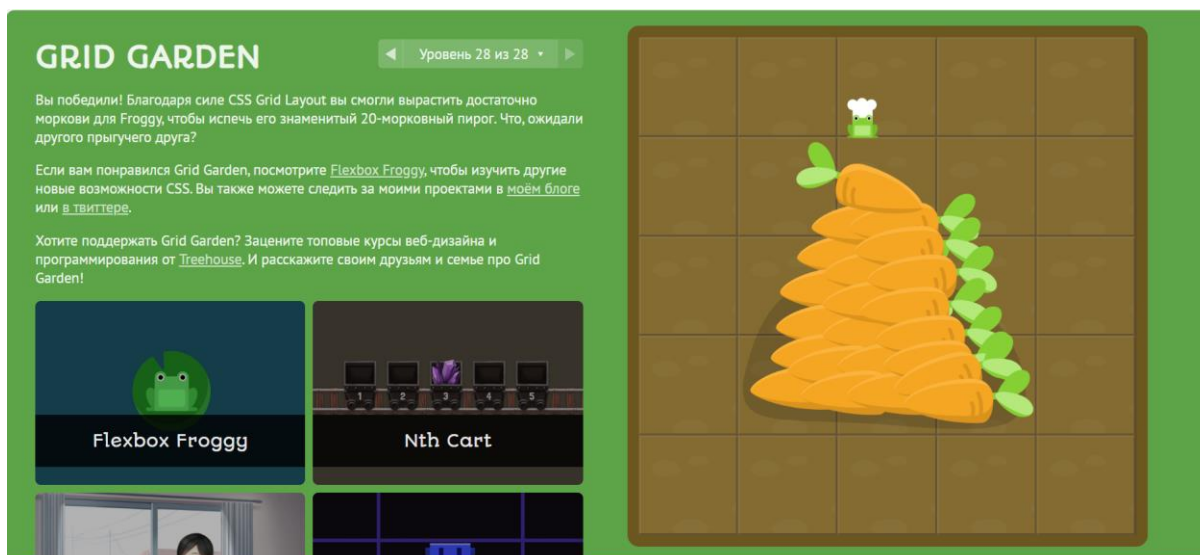
Два свойства flex-direction и flex-wrap используются так часто вместе, что было создано свойство flex-flow для их комбинирования. Это свойство принимает их значения, разделённые пробелом.

Можно 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, чтобы заполнить контейнер равномерно.

Align-content отвечает за расстояние между рядами, а align-items отвечает за то, как элементы в целом будут выровнены в контейнере.

## 2.



Далее была похожая на первую игра **CSS Grid**.

Когда вы используется `grid-column-start`, то `grid`-элемент по умолчанию выделит только один столбец. Однако можно увеличить элемент между несколькими столбцами, если добавите свойство `grid-column-end`.

Вместо определения `grid`-элемента на основе начальной и конечной позиции `grid`-строк вы можете задавать их с помощью необходимой вам ширины строк, используя `span`. Помните, что `span`, в отличие от `grid`, работает только с положительными значениями. Также есть краткая форма `grid-column`, которая принимает оба значения сразу через косую черту: `/`.

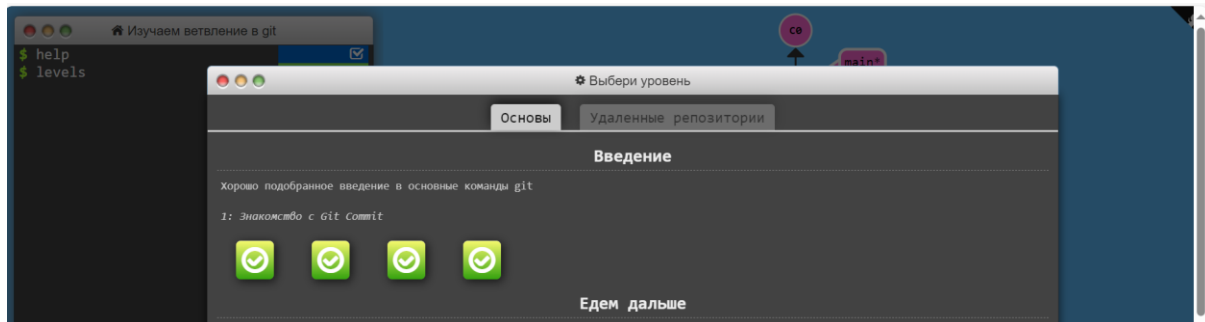
`Grid-row-start` работает почти как `grid-column-start`, но только по вертикальной оси. Есть краткая форма `grid-row`.

Бывает лень печатать `grid-column` и `grid-row`, для этого есть ещё более краткое свойство `grid-area`, которое принимает 4 значения, разделенные косой чертой - `/`: `grid-row-start`, `grid-column-start`, `grid-row-end` и `grid-column-end`.

Можно изменить `grid`-сетку как захочется, для этого нужно использовать `grid-template-columns` : новое значение(по умолчанию 20/20/20/20/20).

### 3.

При выполнении третьего задания работали с командами системы контроля версий Git: commit, branch, checkout, rebase, merge.



### Вывод:

В ходе выполнения домашней работы были получены базовые знания по Flexbox CSS и Grid. Также были получены знания по работе с Git'ом, а точнее познакомились с командами commit, branch, checkout, rebase, merge.