

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Практическая работа №1  
“CSS Flexbox, CSS Grid”

Выполнил:  
Стукалов Артем

Группа  
К33392

Проверил:  
Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Пройти три игры

- 1) <https://flexboxfroggy.com/#ru>
- 2) <https://cssgridgarden.com/#ru>
- 3) [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

### FLEXBOX FROGGY

Первые 5 заданий являются простой отработкой правил выравнивания элементов вдоль одной из осей. Заострять внимание особо не на чем.

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

Следующий

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

Следующий

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

Следующий

```
1 #pond {
2   display: flex;
3   align-items: flex-end;
4 }
```

Следующий

Далее идет практика комбинирования этих стилей с добавлением еще одного, который позволяет изменять направление главной оси(flex-direction).

```
1 #pond {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5 }
```

Следующий

[Меню](#) [Flexbox](#) [Flexbox-свойства](#) [Codepen](#) [GitHub](#) [Twitter](#) [Наш блог](#)

```
1 #pond {
2   display: flex;
3   justify-content: space-around;
4   align-items: flex-end;
5 }
```

Следующий

```
1 #pond {
2   display: flex;
3   flex-direction: row-reverse;
4 }
```

Следующий

[Меню](#) [Flexbox](#) [Flexbox-свойства](#) [Codepen](#) [GitHub](#) [Twitter](#) [Наш блог](#)

```
1 #pond {
2   display: flex;
3   flex-direction: column;
4 }
```

Следующий

[Меню](#) [Flexbox](#) [Flexbox-свойства](#) [Codepen](#) [GitHub](#) [Twitter](#) [Наш блог](#)

```
1 #pond {  
2   display: flex;  
3   flex-direction: row-reverse;  
4   justify-content: flex-end;  
5 }
```

Следующий

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

Следующий

```
1 #pond {  
2   display: flex;  
3   flex-direction: column-reverse;  
4   justify-content: space-between;  
5 }
```

Следующий

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

Следующий

После идет практика изменения стилей отдельных элементов: порядок элемента на вдоль главной оси и выравнивание элемента вдоль побочной оси.

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

Следующий

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

Следующий

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

Следующий

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

Следующий

Следующим свойством является возможность переноса элементов вдоль главной оси при невозможности уложиться в заданную ширину.

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

[Следующий](#)

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

[Следующий](#)

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

[Следующий](#)

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

[Следующий](#)

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

Следующий

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap;
4   flex-direction: column-reverse;
5   align-content: center;
6 }
```

Следующий

Последнее задание является наиболее интересным, так как требует применения 4 стилей одновременно. Разберем более подробно.

```
1 #pond {
2   display: flex;
3   flex-direction: column-reverse;
4   flex-wrap: wrap-reverse;
5   justify-content: center;
6   align-content: space-between;
7 }
```

Следующий

## GRID GARDEN

Все задания являются практикой достаточно простых CSS свойств.

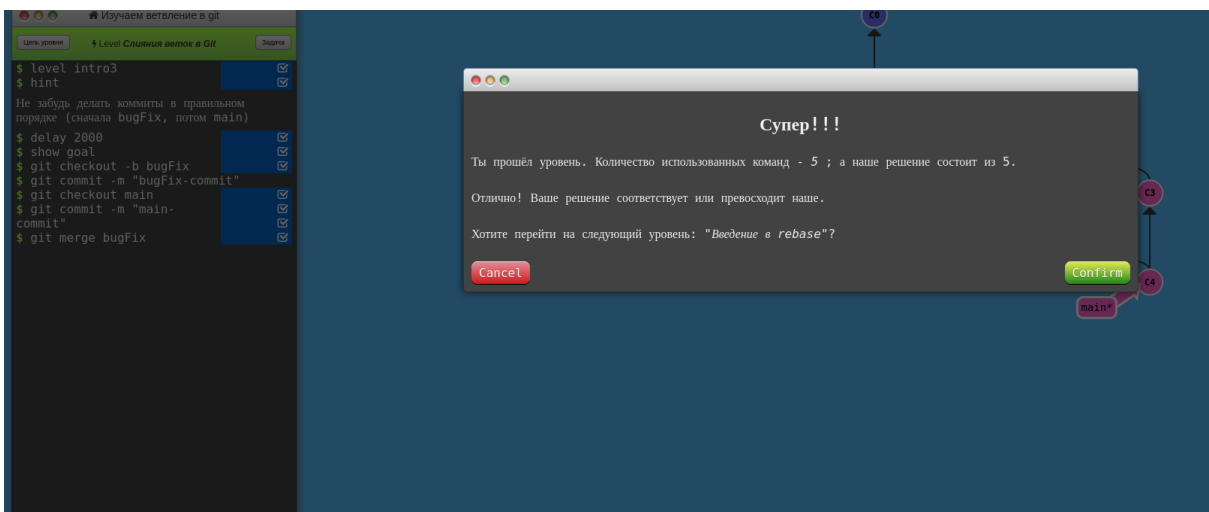
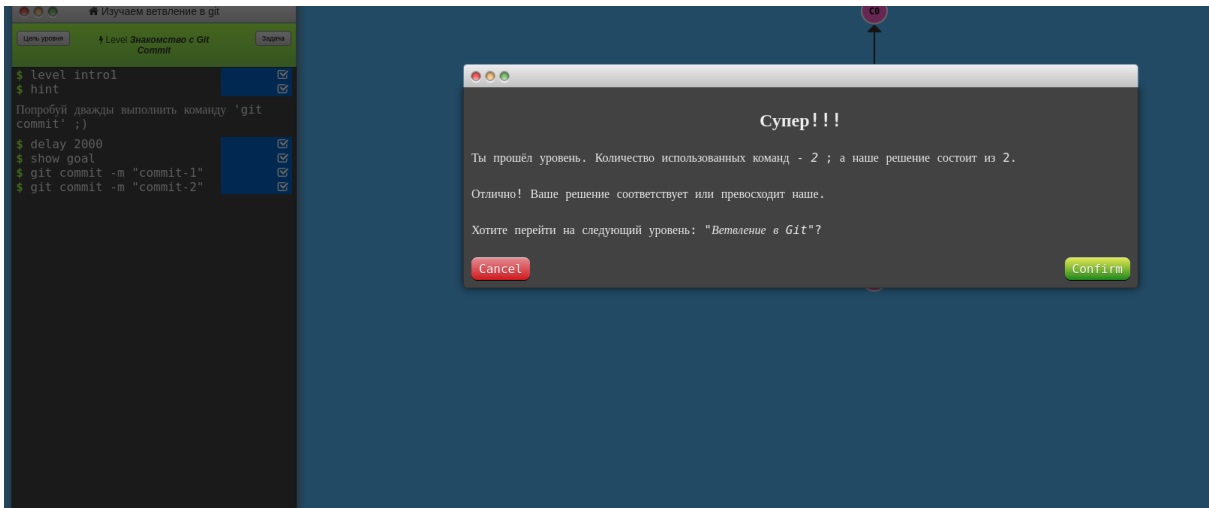
- 1) grid-column-start: 3;
- 2) grid-column-start: 5;
- 3) grid-column-end: 4;
- 4) grid-column-end: 2;
- 5) grid-column-end: -2;
- 6) grid-column-start: -3;

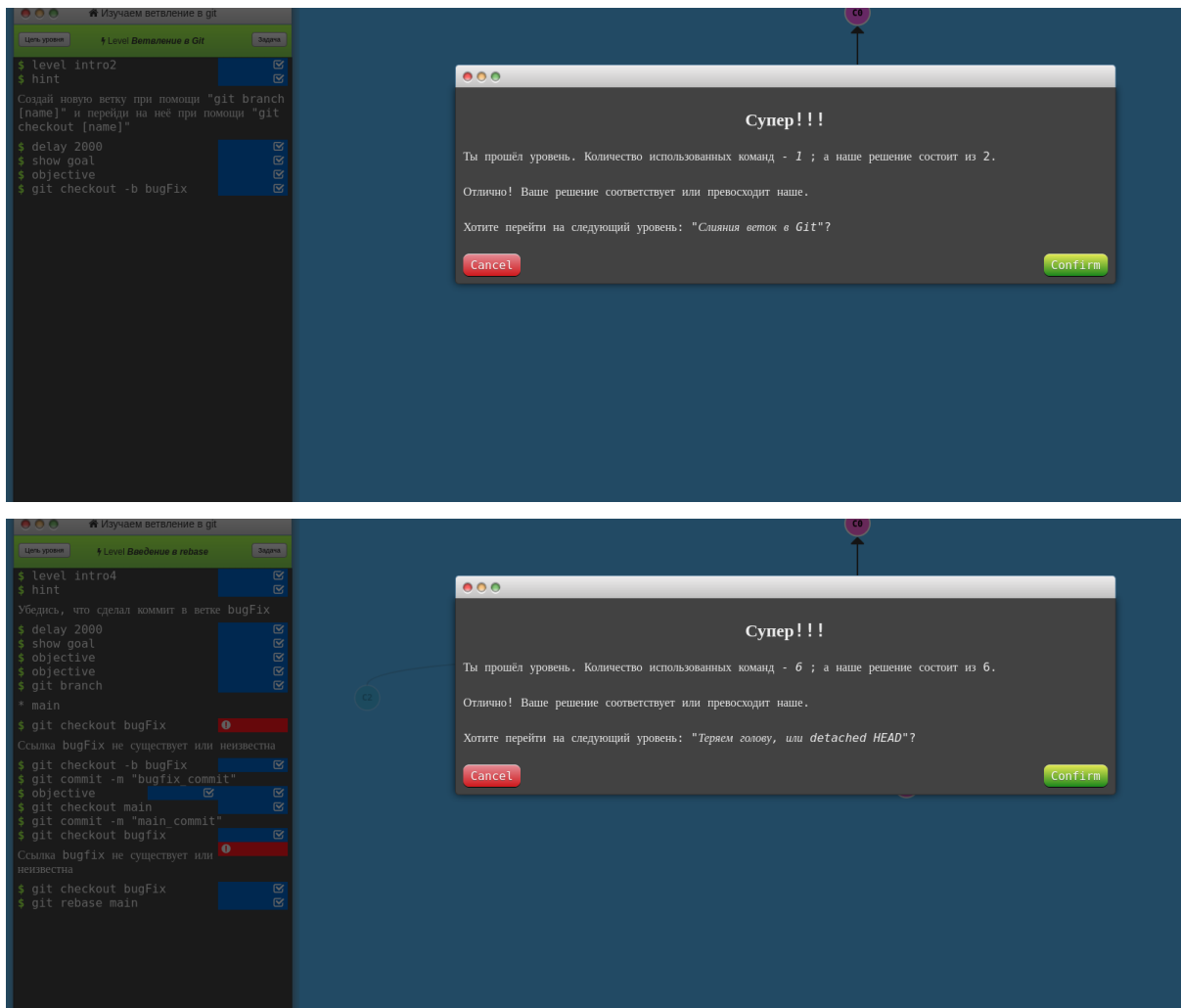


- 7) `grid-column-end: span 2;`
- 8) `grid-column-end: span 5;`
- 9) `grid-column-start: span 3;`
- 10) `grid-column: -3/-1;`
- 11) `grid-column: span 3/5;`
- 12) `grid-row-start: 3;`
- 13) `grid-row: 3/6;`
- 14) `grid-row: 5; grid-column: 2;`
- 15) `grid-column: 2/6; grid-row: 1/6;`
- 16) `grid-area: 1/2/4/6;`
- 17) `grid-area: 2/3/5/6;`
- 18) `order: 1;`
- 19) `order: -1;`
- 20) `grid-template-columns: 50%;`
- 21) `grid-template-columns: repeat(8, 12.5%);`
- 22) `grid-template-columns: 100px 3em 40%;`
- 23) `grid-template-columns: 1fr 5fr;`
- 24) `grid-template-columns: 50px 1fr 1fr 1fr 50px;`
- 25) `grid-template-columns: 75px 3fr 2fr;`
- 26) `grid-template-rows: repeat(4, 12.5px);`
- 27) `grid-template: 60% 60% / 200px;`
- 28) `grid-template: 1fr 50px / 20% 80%;`

## LearnGitBranching

Первые 2 стиля позволят нам совершить перенос элементов против главной оси в обратном порядке. Таким образом мы добьемся верного порядка лягушек. Останется лишь выровнять их вдоль главной оси и расстояние между рядами.





## Вывод

Пройдя 3 тематические игры, удалось овладеть базовыми навыками использования css flex-box и grid. Также удалось овладеть базовыми навыками работы с git ветками: checkout, rebase, merge.