

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа 1

Выполнила:
Кормановская Дарина
K33401

Проверил:
Добряков Д. И.

Санкт-Петербург

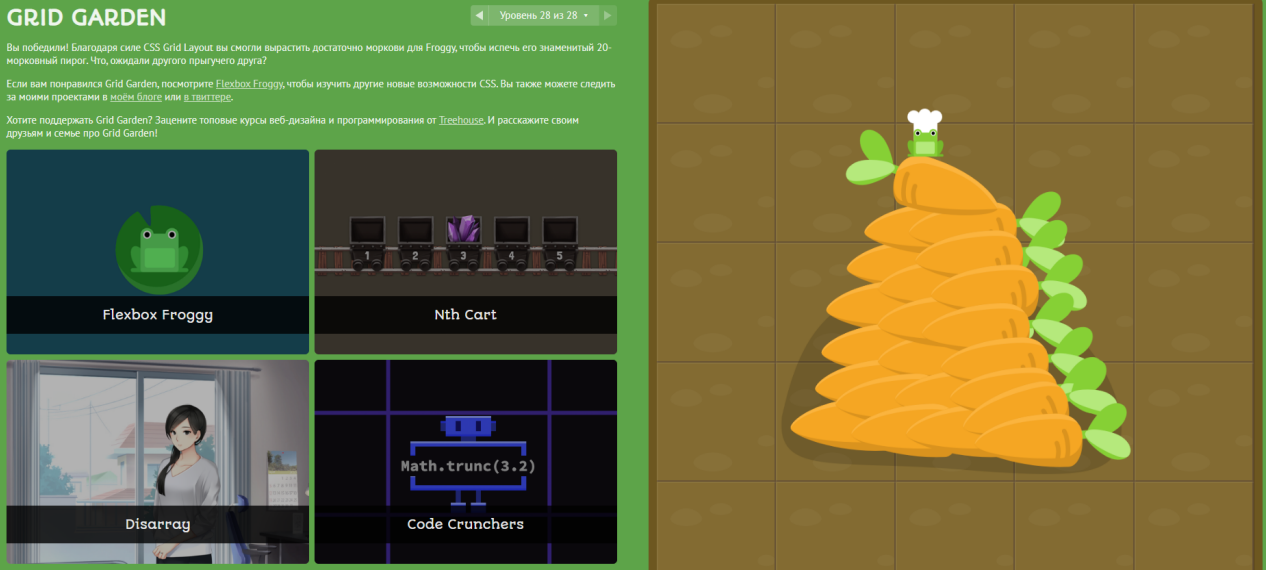
2022 г.

Задача

Изучить работу CSS grid и CSS flexbox в игровой форме, пройдя две игры:

1. <https://cssgridgarden.com/#ru>
2. <https://flexboxfroggy.com/#ru>

Ход работы

<div>https://cssgridgarden.com/</div> <div></div>	
#	скриншот
1	<div><p>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 <code>grid-column-start</code>.</p><p>Например, <code>grid-column-start: 3</code> покрывает водой зону, начинающуюся на третьей grid-линии по вертикали, это как сказать "третья вертикальная граница на grid-сетке слева".</p></div> <div><pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column-start: 3; 9 }</pre></div>

2	<pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #poison { 8 grid-column-start: 5; 9 }</pre>
3	<pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column-start: 1; 9 grid-column-end: 4; 10 }</pre>
4	<pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column-start: 5; 9 grid-column-end: 2; 10 }</pre>
5	<pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column-start: 1; 9 grid-column-end: -2; 10 }</pre>
6	<pre>1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #poison { 8 grid-column-start: -3; 9 }</pre>

7

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` работает только с положительными значениями.

Для примера, полейте эту морковь, используя свойство `grid-column-end: span 2`.

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 2;
9   grid-column-end: span 2;
10 }
```

8

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 1;
9   grid-column-end: span 5;
10 }
```

9

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: span 3;
9   grid-column-end: 6;
10 }
```

10

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 4 / 6
9 }
10 }
```

11

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 2 / span 3
9 }
```

12

Одна из вещей, которая отличает CSS Grid Layout от Flexbox'a, — это то, что вы можете легко позиционировать элементы в двух направлениях: в столбцах и в строках. `grid-row-start` работает почти как `grid-column-start`, но только по вертикальной оси.

Используйте `grid-row-start`, чтобы полить эту морковь.

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-row-start: 3;
9 }
```

13

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-row: 3 / 6;
9 }
```

14

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #poison {
8   grid-column: 2 / 3;
9   grid-row: 5 / span 1;
10 }
11
```

15

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 2 / 6;
9   grid-row: 1 / 6;
10 }
11
```

16

Если вас также утомляет печатать `grid-column` и `grid-row`, есть сокращённое свойство и для этого. `grid-area` принимает 4 значения, разделённые косой чертой `/`: `grid-row-start`, `grid-column-start`, `grid-row-end` и `grid-column-end`.

Пример: `grid-area: 1 / 1 / 3 / 6;`.

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-area: 1 / 2 / 4 / 6;
9 }
10
```

17

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water-1 {
8   grid-area: 1 / 4 / 6 / 5;
9 }
10
11 #water-2 {
12   grid-area: 2 / 3 / span 3 / span 3;
13 }
14
```

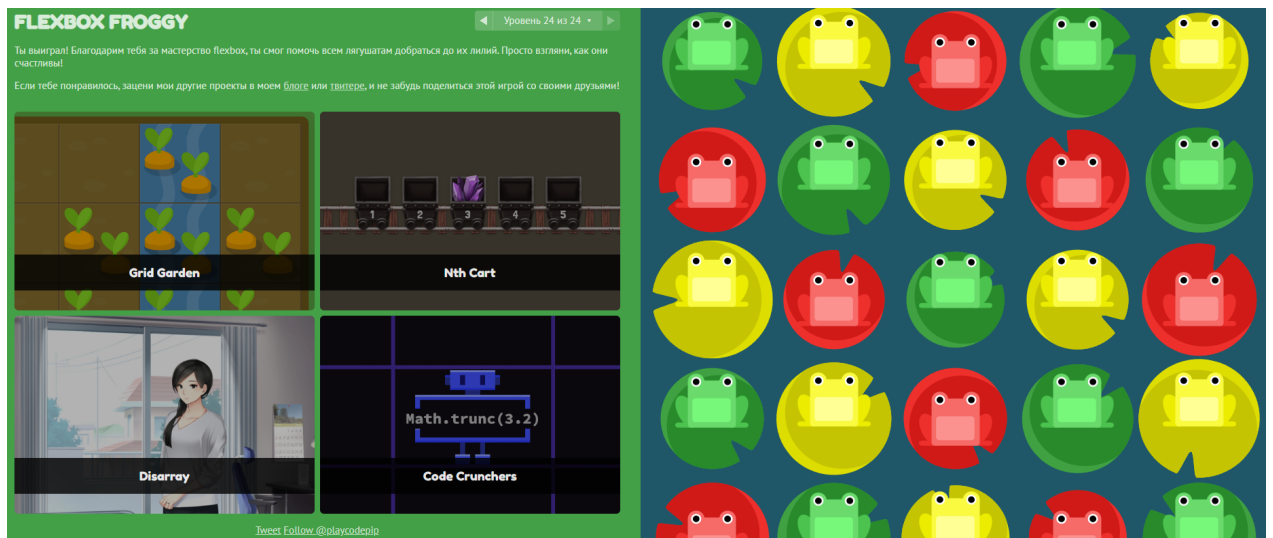
18

```
1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 .water {
8   order: 0;
9 }
10
11 #poison {
12   order: 1;
13 }
```

19	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 .water { 8 order: 0; 9 } 10 11 .poison { 12 order: -1; 13 } 14 </pre>
20	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 50%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column: 1; 9 grid-row: 1; 10 } 11 12 </pre>
21	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: repeat(8, 12.5%); 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-column: 1; 9 grid-row: 1; 10 } 11 </pre>
22	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 100px 3em 40%; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 </pre>
23	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 1fr 5fr; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } </pre>

24	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 50px 1fr 1fr 1fr 50px; 4 grid-template-rows: 20% 20% 20% 20% 20%; 5 } 6 7 #water { 8 grid-area: 1 / 1 / 6 / 2; 9 } 10 11 #poison { 12 grid-area: 1 / 5 / 6 / 6; 13 }</pre>
25	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 75px 3fr 2fr; 4 grid-template-rows: 100%; 5 }</pre>
26	<pre> 1 #garden { 2 display: grid; 3 grid-template-columns: 20% 20% 20% 20% 20%; 4 grid-template-rows: 12.5px 12.5px 12.5px 12.5px 1fr; 5 } 6 7 #water { 8 grid-column: 1 / 6; 9 grid-row: 5 / 6; 10 }</pre>
27	<pre> 1 #garden { 2 display: grid; 3 grid-template: 60% 40% / 200px; 4 } 5 6 #water { 7 grid-column: 1; 8 grid-row: 1; 9 }</pre>
28	<pre> 1 #garden { 2 display: grid; 3 grid-template: 1fr 50px / 20% 80%; 4 }</pre>

<https://flexboxfroggy.com/>



скриншот

1	<div><h2>FLEXBOX FROGGY</h2><p>Уровень 1 из 24</p><p>Добро пожаловать в Flexbox Froggy. Игра, в которой тебе нужно помочь лягушонку Фроги и его друзьям написанием CSS кода! Направь этого лягушонка на лилию справа используя свойство justify-content, которое выравнивает элементы горизонтально и принимает следующие значения:</p><ul style="list-style-type: none">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.<p>Например, justify-content: flex-end; сдвинет лягушонка вправо.</p><pre>1 #pond { 2 display: flex; 3 justify-content: flex-end; 4 }</pre></div>
2	<div><pre>1 #pond { 2 display: flex; 3 justify-content: center; 4 }</pre></div>
3	<div><pre>1 #pond { 2 display: flex; 3 justify-content: space-around; 4 }</pre></div>
4	<div><pre>1 #pond { 2 display: flex; 3 justify-content: space-between; 4 }</pre></div>

5

FLEXBOX FROGGY

Уровень 5 из 24

Теперь используй `align-items` чтоб помочь лягушатам добраться 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.

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

6

FLEXBOX FROGGY

Уровень 6 из 24

Направь лягушонка в центр пруда, используя `justify-content` и `align-items` вместе.

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

7

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

8

FLEXBOX FROGGY

Уровень 8 из 24

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;  
3   flex-direction: row-reverse;  
4 }
```

9	<div> <div>FLEXBOX FROGGY</div> <div>Уровень 9 из 24</div> </div> <p>Помоги лягушатам расположиться на своих лилиях используя <code>flex-direction</code>. Это CSS свойство задает направление, в котором будут расположены элементы в контейнере и принимает следующие значения:</p> <ul style="list-style-type: none"> • <code>row</code>: Элементы размещаются по направлению текста. • <code>row-reverse</code>: Элементы отображаются в обратном порядке к направлению текста. • <code>column</code>: Элементы располагаются сверху вниз. • <code>column-reverse</code>: Элементы располагаются снизу вверх. <pre> 1 #pond { 2 display: flex; 3 flex-direction: column; 4 }</pre>
10	<pre> 1 #pond { 2 display: flex; 3 flex-direction: row-reverse; 4 justify-content: flex-end; 5 }</pre>
11	<pre> 1 #pond { 2 display: flex; 3 flex-direction: column; 4 justify-content: flex-end; 5 }</pre>
12	<pre> 1 #pond { 2 display: flex; 3 flex-direction: column-reverse; 4 justify-content: space-between; 5 }</pre>
13	<pre> 1 #pond { 2 display: flex; 3 flex-direction: row-reverse; 4 justify-content: center; 5 align-items: flex-end; 6 }</pre>

14

FLEXBOX FROGGY

Уровень 14 из 24

Иногда изменения порядка отображения элементов в контейнере недостаточно. В таких случаях мы можем применить свойство **order** 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.

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

15

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

16

FLEXBOX FROGGY

Уровень 16 из 24

Еще одно свойство, которое ты можешь применить к определенному элементу это **align-self**. Это свойство принимает те же значения, что и **align-items**.

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

17

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

Следующий

18

FLEXBOX FROGGY

Уровень 18 из 24

О нет! Лягушат сплющило на одном ряду лилий. Раздвинь их с помощью свойства `flex-wrap`, которое принимает следующие значения:

- **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.

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

19

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

20

FLEXBOX FROGGY

Уровень 20 из 24

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

Например, ты можешь использовать `flex-flow: row wrap`, чтоб элементы располагались в ряд и автоматически переносились на новую строку.

Попробуй использовать `flex-flow`, чтоб повторить предыдущий уровень.

```
1 #pond {
2   display: flex;
3   flex-flow: column wrap;
4 }
5
```

21

FLEXBOX FROGGY

Уровень 21 из 24

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 }
```

22	<pre> 1 #pond { 2 display: flex; 3 flex-wrap: wrap; 4 align-content: flex-end; 5 }</pre>
23	<pre> 1 #pond { 2 display: flex; 3 flex-wrap: wrap; 4 flex-direction: column-reverse; 5 align-content: center; 6 }</pre>
24	<div> <div>FLEXBOX FROGGY</div> <div>Уровень 24 из 24</div> </div> <p>Доставь лягушат по домам в последний раз, используя CSS свойства, которые ты выучил:</p> <ul style="list-style-type: none"> • <code>justify-content</code> • <code>align-items</code> • <code>flex-direction</code> • <code>order</code> • <code>align-self</code> • <code>flex-wrap</code> • <code>flex-flow</code> • <code>align-content</code> <pre> 1 #pond { 2 display: flex; 3 flex-flow: column-reverse wrap-reverse; 4 justify-content: center; 5 align-content: space-between; 6 } 7</pre>

Вывод

Изучена работа

- **CSS grid:**
 - создание грида (`grid-template-columns` `grid-template-rows` `grid-template`) с использованием различных измерений (`fr` `px` `%` `em`)
 - размещение элементов (`grid-row` `grid-column` `grid-row-start` `grid-row-end` `grid-area` `order` и проч)
- **CSS flexbox:**
 - выравнивание контента горизонтально
 - вертикальное выравнивание
 - порядок расположения и перенос элементов