

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №1

Выполнила:

Злотникова Карина

Группа К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

ЗАДАЧА

Пройти три игры на изучение базы работы с git, Flexbox и Grid.

ХОД РАБОТЫ

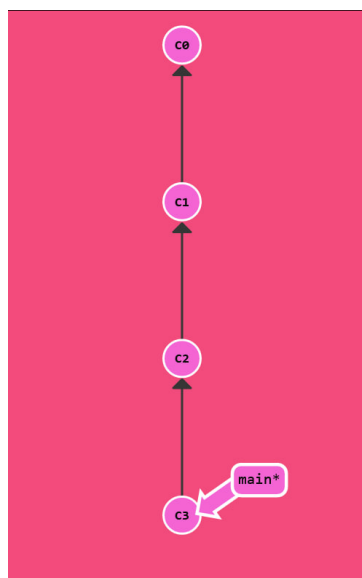
Первый сайт-игра по изучению базы работы с Git

Задание 1:

Текст задания:

Пора попробовать! После того, как это окно закроется, сделай два коммита, чтобы пройти этот уровень.

Схема уровня:



Решение:

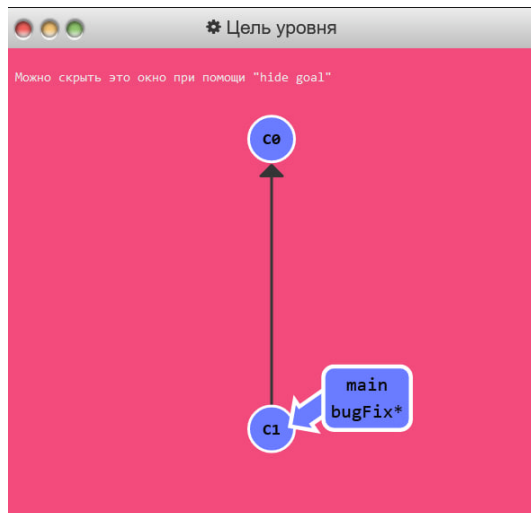
```
$ level intro1
$ hint
Попробуй дважды выполнить команду 'git commit' ;)
$ delay 2000
$ show goal
$ git commit
$ git commit
```

Задание 2:

Текст задания:

Ну что ж, теперь ты готов к работе с ветками. Как только это окно закроется, создай ветку с именем bugFix и переключись на неё.

Схема уровня:



Решение:

```
$ level intro2
$ hint
Создай новую ветку при помощи "git branch [name]" и перейди на неё при помощи "git checkout [name]"
$ delay 2000
$ show goal
$ objective
$ objective
$ git checkout -b bugFix
```

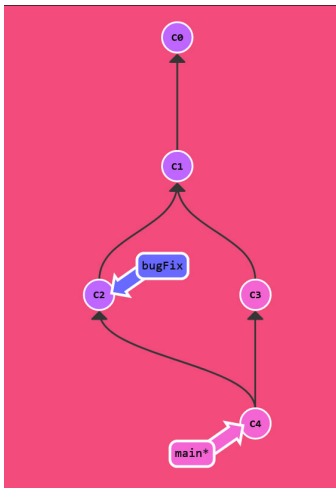
Задание 3:

Текст задания:

Чтобы пройти этот уровень, сделай следующее:

- Создай новую ветку под названием `bugFix`
- Переключись на новую ветку `bugFix` командой `git checkout bugFix`
- Сделай один коммит
- Вернись на ветку `main` при помощи `git checkout`
- Сделай ещё один коммит
- Слей ветку `bugFix` с веткой `main` при помощи `git merge`

Схема уровня:



Решение:

```
Исучаем ветвление в git
Цель уровня ⚡ Level Слияния веток в Git Задача
$ level intro3 [✓]
$ hint [✓]
Не забудь делать коммиты в правильном
порядке (сначала bugFix, потом main)
$ delay 2000 [✓]
$ show goal [✓]
$ objective [✓]
$ git checkout -b bugFix [✓]
$ git commit [✓]
$ git checkout main [✓]
$ git commit [✓]
$ git merge bugFix [✓]
```

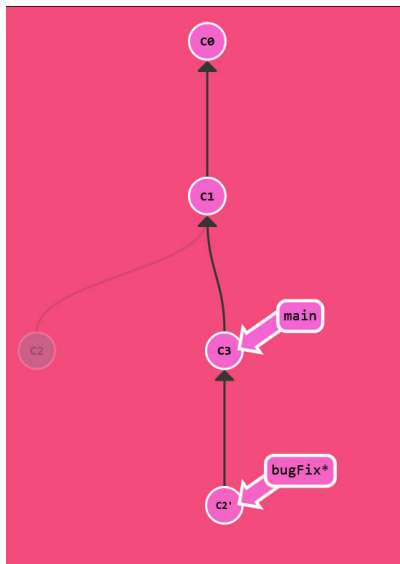
Задание 4:

Текст задания:

Чтобы пройти этот уровень, сделай следующее:

- Переключись на ветку `bugFix`
- Сделай коммит
- Вернись на `main` и сделай коммит ещё раз
- Переключись на `bugFix` и сделай rebase на `main`

Схема уровня:



Решение:

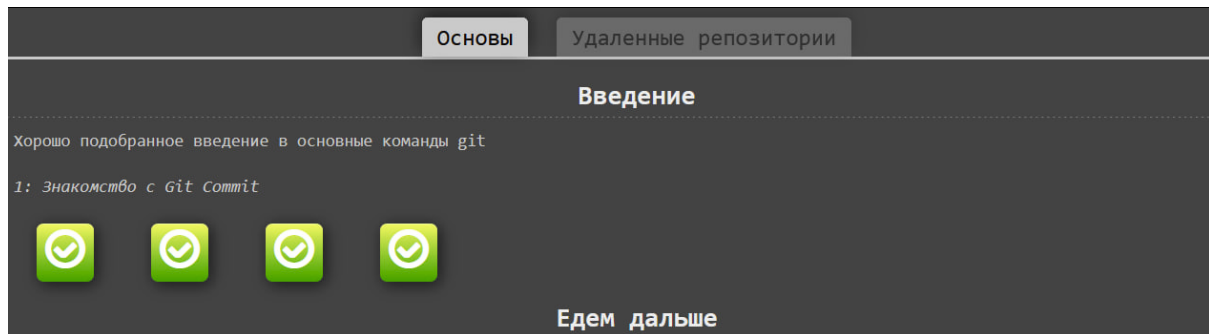
```
Изучаем ветвление в git

Цель уровня ⚡ Level Введение в rebase Задача

$ level intro4
$ hint
Убедись, что сделал коммит в ветке bugFix
$ delay 2000
$ show goal
$ git checkout -b bugFix
$ git commit
$ git checkout main
$ git commit
$ git checkout bugFix
$ git rebase main
```

Итог:

Пройдено введение в git.



[Вторая сайт-игра по изучению FlexBox](#) (Ниже приведены примеры самого простого и самого сложного задания этой игры)

Задание 1:

Текст задания:

FLEXBOX FROGGY ◀ Уровень 1 из 24 ▶

Добро пожаловать в Flexbox Froggy. Игра, в которой тебе нужно помочь лягушонку Фроги и его друзьям, напиконтейнер CSS код! Направь этого лягушонка на лилию справа, используя свойство `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.

Например, `justify-content: flex-end;` сдвинет лягушонка вправо.

Решение:

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

Следующий

Задание 24:

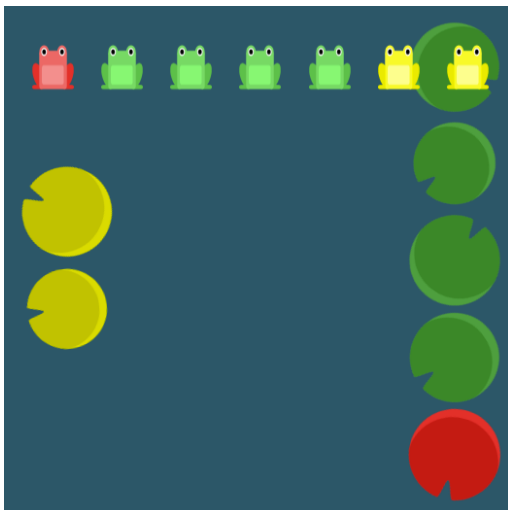
Текст задания:

FLEXBOX FROGGY ◀ Уровень 24 из 24 ▶

Доставь лягушат домой в последний раз, используя свойства CSS, которые ты выучил:

- `justify-content`
- `align-items`
- `flex-direction`
- `order`
- `align-self`
- `flex-wrap`
- `flex-flow`
- `align-content`

Схема уровня:



Решение:

```
1 #pond {
2   display: flex;
3   flex-wrap: wrap-reverse;
4   flex-direction: column-reverse;
5   justify-content: center;
6   align-content: space-between;
7 }
8
9
10
```

Следующий

[Третья сайт-игра по изучению Grid](#) (Ниже приведены примеры самого простого и самого сложного задания этой игры)

Задание 1:

Текст задания:

GRID GARDEN

Уровень 1 из 28

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".

Схема уровня:



Решение:

```
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 }
10
```


Задание 28:

Текст задания:

GRID GARDEN

Уровень 28 из 28

Сад выглядит прекрасно. Здесь вы оставили 50-пиксельную дорожку снизу и заполнили всё оставшееся пространство морковью.

К сожалению, левые 20% сада заполнили сорняки. Используйте CSS Grid Layout в последний раз, чтобы очистить сад.

Схема уровня:



Решение:

```
1 #garden {  
2   display: grid;  
3   grid-template: 1fr 50px /20% 80%;  
4 }
```

ВЫВОД

В ходе прохождения данных игр пользователь учится работать с git и знакомится со свойствами CSS:

- justify-content:
 - flex-start: Выравнивание элементов по началу контейнера.
 - flex-end: Выравнивание элементов по концу контейнера.
 - center: Выравнивание элементов по центру контейнера.
 - space-between: Равномерное распределение элементов вдоль оси с промежутками между ними.
 - space-around: Равномерное распределение элементов вдоль оси с равными промежутками до и после элементов.
 - space-evenly: Равномерное распределение элементов вдоль оси с равными промежутками между ними и вокруг них.
- align-items:
 - flex-start: Выравнивание элементов по началу перпендикулярной оси.
 - flex-end: Выравнивание элементов по концу перпендикулярной оси.
 - center: Выравнивание элементов по центру перпендикулярной оси.
 - baseline: Выравнивание элементов по базовой линии.
 - stretch: Растягивание элементов по высоте контейнера.
- flex-direction:
 - row: Расположение элементов в строку.
 - row-reverse: Расположение элементов в обратную строку.
 - column: Расположение элементов в столбец.
 - column-reverse: Расположение элементов в обратный столбец.
- order:
 - Целое число, определяющее порядок элемента в контейнере. Меньшие значения будут располагаться раньше.
- align-self:
 - auto: Использование значения align-items из родительского контейнера.
 - flex-start: Выравнивание элемента по началу перпендикулярной оси.
 - flex-end: Выравнивание элемента по концу перпендикулярной оси.
 - center: Выравнивание элемента по центру перпендикулярной оси.
 - baseline: Выравнивание элемента по базовой линии.
 - stretch: Растягивание элемента по высоте контейнера.
- flex-wrap:

- nowrap: Элементы не переносятся на следующую строку или столбец.
 - wrap: Элементы переносятся на следующую строку или столбец, если не помещаются.
 - wrap-reverse: Элементы переносятся на следующую строку или столбец в обратном порядке.
- flex-flow:
 - Комбинация flex-direction и flex-wrap в одном свойстве. Например, row nowrap.
- align-content:
 - flex-start: Выравнивание контента по началу перпендикулярной оси.
 - flex-end: Выравнивание контента по концу перпендикулярной оси.
 - center: Выравнивание контента по центру перпендикулярной оси.
 - space-between: Равномерное распределение контента вдоль перпендикулярной оси с промежутками между ними.
 - space-around: Равномерное распределение контента вдоль перпендикулярной оси с равными промежутками до и после контента.
 - stretch: Растягивание контента по высоте контейнера.
- grid-template-columns: Определяет ширину и количество колонок в сетке.
- grid-template-rows: Определяет высоту и количество строк в сетке.
- grid-template-areas: Определяет области сетки с использованием именованных областей и grid-area свойств элементов.
- grid-template: Сокращенное свойство для задания широты, высоты и областей сетки.
- grid-column-start и grid-column-end: Определяют начальную и конечную колонку элемента.
- grid-row-start и grid-row-end: Определяют начальную и конечную строку элемента.
- grid-column и grid-row: Сокращенные свойства для задания начальных и конечных позиций элемента в сетке.
- grid-area: Сокращенное свойство для задания области элемента в сетке с использованием именованных областей.