АКАДЕМИЯ УПРАВЛЕНИЯ ПРИ ПРЕЗИДЕНТЕ РЕСПУБЛИКИ БЕЛАРУСЬ

ИНСТИТУТ УПРАВЛЕНЧЕСКИХ КАДРОВ

Кафедра управления информационными ресурсами

Специальность «Управление информационными ресурсами»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине «Web-дизайн»

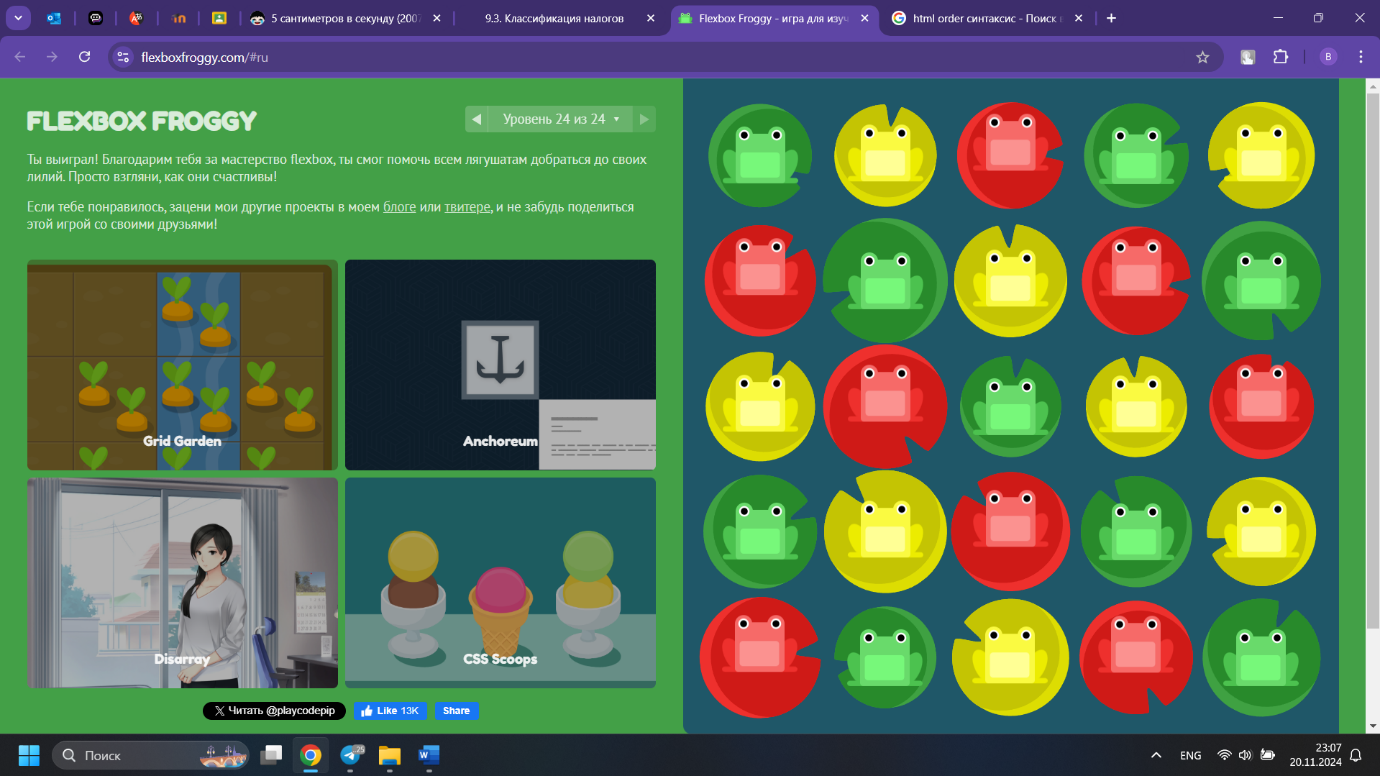
на тему «Позиционирование. Верстка при помощи Flexbox»

|  |  |  |
| --- | --- | --- |
| Студентка  3 курса, УИР-2 |  | Ю.В. Мозолева |
| Руководитель Преподаватель |  | И.В. Виноградова |

Минск 2024

**Основное задание**

***1. Потренируйтесь*** [***https://flexboxfroggy.com/#ru***](https://flexboxfroggy.com/#ru)

Скриншот выполненного задания:

***2. Используя компоненты блочной модели и позиционирование создайте страницу по образцу:***



Код страницы:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Positioning</title>

    <style>

        body {

            margin: 0;

            font-family: Arial, sans-serif;

        }

        .box-static {

            width: 100px;

            height: 100px;

            background-color: lightblue;

            margin: 10px;

            flex-direction: column;

        }

        .box-relative-1 {

            width: 100px;

            height: 100px;

            background-color: lightgreen;

            position: relative;

            top: 20px;

            left: 30px;

        }

        .box-relative-2 {

            width: 115px;

            height: 115px;

            background-color: pink;

            position: relative;

            top: -7px;

            left: 37px;

        }

        .container {

            width: 300px;

            height: 300px;

            background-color: lightgray;

            position: relative;

            margin: 20px;

        }

        .box-absolute-1 {

            width: 100px;

            height: 100px;

            background-color: rgb(255, 123, 0);

            position: absolute;

            top: 25px;

            left: 25px;

            z-index: 3;

        }

        .box-absolute-2 {

            width: 130px;

            height: 130px;

            background-color: lightblue;

            position: absolute;

            top: 60px;

            left: 50px;

            z-index: 2;

        }

        .box-absolute-3 {

            width: 160px;

            height: 160px;

            background-color: purple;

            position: absolute;

            top: 100px;

            left: 80px;

            z-index: 1;

        }

        .box-zindex-1 {

            width: 170px;

            height: 170px;

            background-color: rgb(223, 102, 102);

            position: absolute;

            top: 75px;

            right: 80px;

            z-index: 5;

        }

        .box-zindex-2 {

            width: 170px;

            height: 170px;

            background-color: rgb(2, 183, 183);

            position: absolute;

            top: 120px;

            right: 35px;

            z-index: 4;

        }

        .box-fixed-1 {

            width: 100px;

            height: 100px;

            background-color: yellow;

            position: fixed;

            top: 10px;

            right: 10px;

        }

        .box-fixed-2 {

            width: 100px;

            height: 100px;

            background-color: orange;

            position: fixed;

            bottom: 10px;

            left: 10px;

            z-index: 4;

        }

    </style>

</head>

<body>

    <!-- Static positioned boxes -->

    <div class="box-static">Static</div>

    <div class="box-static">Static</div>

    <!-- Relative positioned boxes -->

    <div class="box-relative-1">Relative 1</div>

    <div class="box-relative-2">Relative 2</div>

    <!-- Container with absolute positioned boxes inside -->

    <div class="container">

        <div class="box-absolute-1">Absolute 1</div>

        <div class="box-absolute-2">Absolute 2</div>

        <div class="box-absolute-3">Absolute 3</div>

    </div>

    <!-- Overlapping boxes with z-index outside container -->

    <div class="box-zindex-1">Z-Index 5</div>

    <div class="box-zindex-2">Z-Index 4</div>

    <!-- Fixed positioned boxes -->

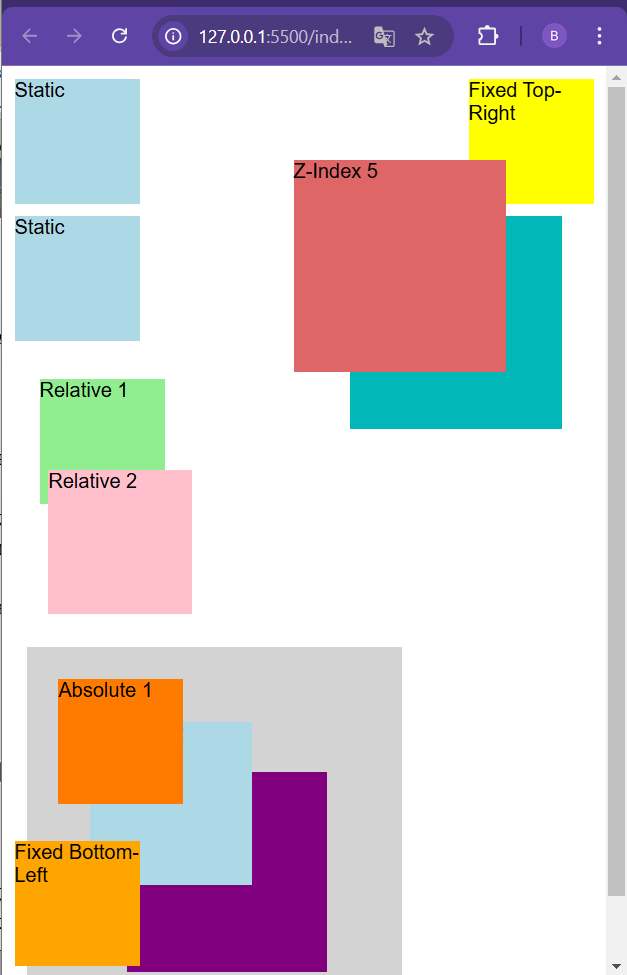
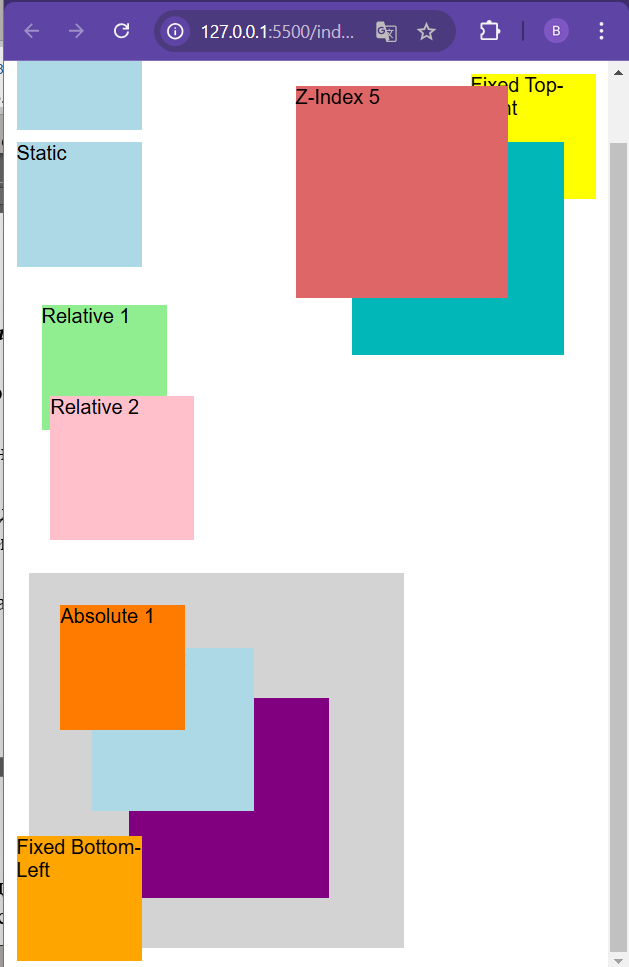
    <div class="box-fixed-1">Fixed Top-Right</div>

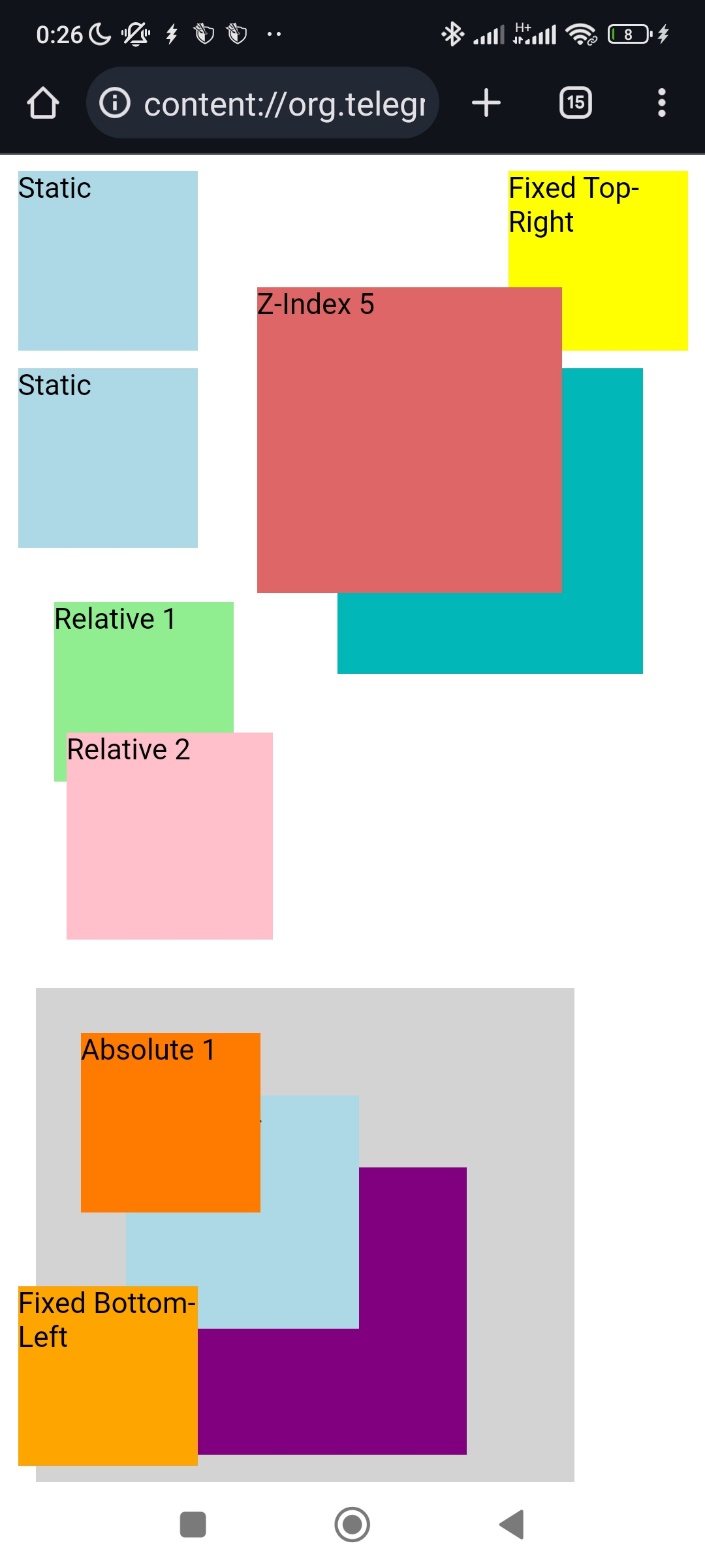
    <div class="box-fixed-2">Fixed Bottom-Left</div>

</body>

</html>

Скриншоты отображения страницы в веб-браузере ПК и мобильной версии:



****

**Контрольные вопросы**

***1. Какие значения может принимать свойство position и как они влияют на положение элемента на странице?***

* static: Значение по умолчанию. Элемент располагается в обычном потоке документа, и его положение не зависит от других элементов. Нельзя использовать свойства top, right, bottom, left.
* relative: Элемент располагается относительно своего исходного положения в потоке документа. Свойства top, right, bottom, left смещают элемент от его исходной позиции.
* absolute: Элемент позиционируется относительно ближайшего предка с установленным позиционированием (не static). Если таких предков нет, элемент будет позиционирован относительно окна браузера.
* fixed: Элемент фиксируется относительно окна браузера, что означает, что он остается на одном месте при прокрутке страницы.
* sticky: Элемент ведет себя как relative, пока не достигнет определенного положения в прокрутке, после чего становится фиксированным.

***2. В чем разница между значениями display: block, display: inline, display: inline-block и display: none? Приведите примеры использования.***

* display: block: Элемент занимает всю доступную ширину и начинается с новой строки. Пример: <div>, <h1>, <p>.

<div>Блочный элемент</div>

<p>Еще один блочный элемент</p>

* display: inline: Элемент занимает только необходимую ширину и не начинает новую строку. Пример: <span>, <a>.

<span>Внутренний элемент</span>

<a href="#">Ссылка</a>

* display: inline-block: Элемент ведет себя как inline, но позволяет задавать высоту и ширину, а также поддерживает отступы и границы.

<div style="display: inline-block; width: 100px; height: 100px; background-color: red;">Inline-block элемент</div>

* display: none: Элемент не отображается на странице и не занимает места в потоке документа.

<div style="display: none;">Этот элемент не виден</div>

***3. Опишите блочную модель CSS. Какие основные свойства влияют на размер и расположение блочных элементов?***

Блочная модель описывает структуру элементов на странице и включает следующие компоненты:

* Контент: Содержимое элемента (текст, изображения и т.д.).
* Паддинг (padding): Внутренние отступы вокруг контента.
* Граница (border): Линия вокруг элемента, которая может иметь толщину и стиль.
* Марджин (margin): Внешние отступы, которые отделяют элемент от других элементов.

Основные свойства, влияющие на размер и расположение блочных элементов:

* width;
* height;
* padding;
* border;
* margin.

***4. Как работает свойство flex-direction в контейнере с display: flex, и какие значения оно может принимать?***

Свойство flex-direction определяет направление главной оси в flex-контейнере. Оно может принимать следующие значения:

* row (по умолчанию): Элементы располагаются в строку слева направо.
* row-reverse: Элементы располагаются в строку справа налево.
* column: Элементы располагаются в столбик сверху вниз.
* column-reverse: Элементы располагаются в столбик снизу вверх.

***5. Как с помощью свойств align-items и justify-content можно управлять расположением элементов в flex-контейнере? Приведите примеры для каждого свойства.***

Align-items управляет выравниванием элементов по поперечной оси (перпендикулярно главной оси). Пример:

.container {

display: flex;

align-items: center; /\* Центрирует элементы по вертикали \*/

}

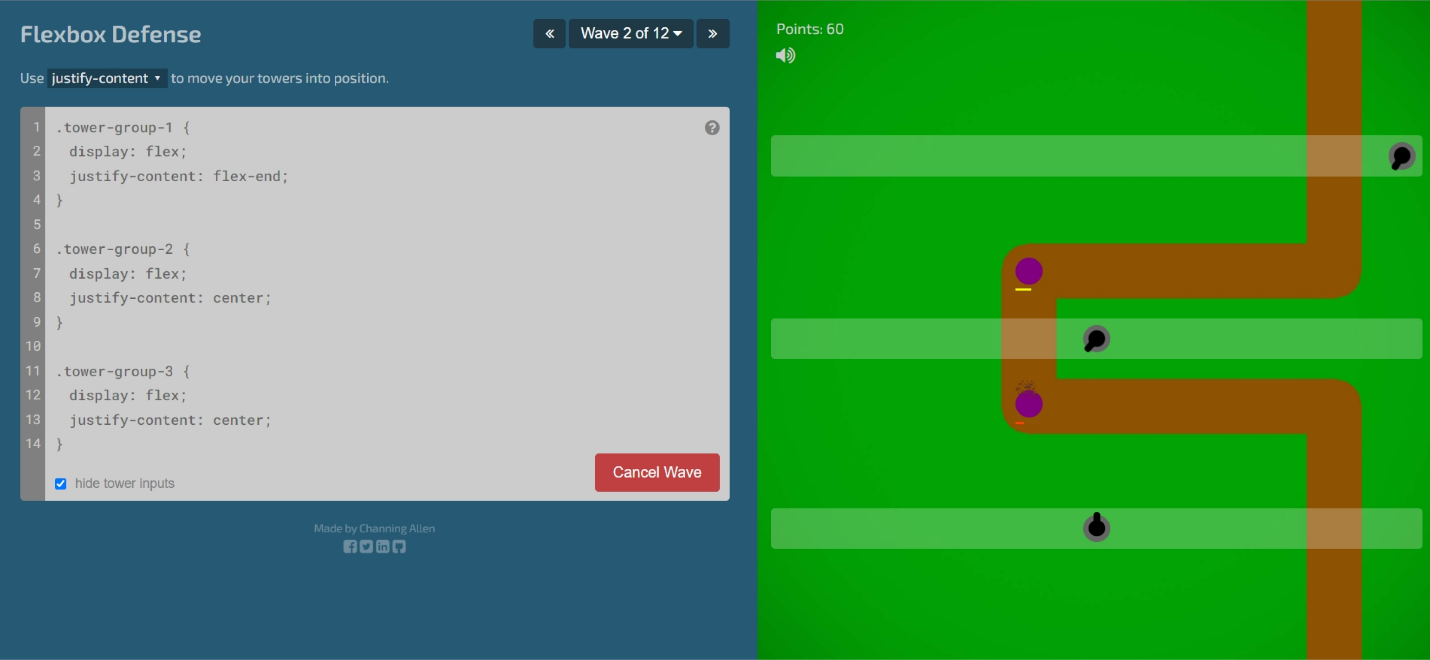
Justify-content управляет распределением пространства между элементами вдоль главной оси. Пример:

.container {

display: flex;

justify-content: space-between; /\* Распределяет элементы с равным расстоянием между ними \*/

}

***6. Найдете в сети красивое применение flexbox.***