**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

##### Группа 73 ТП

## Отчет

## по лабораторной работе № 5

**"Применение различных технологий размещения элементов в HTML-документе"**

учебный предмет

“Программные средства создания Интернет приложений”

**Исполнитель: А.А. Артюшевский**

**Руководитель: Е.П. Сушкевич**

**Минск, 2025**

**Цель работы:**

Формирование умений применения различных способов расположения элементов на HTML-странице средствами CSS.

Задание 1: Создайте страницу, представленную на рисунке 8. Используйте семантическую разметку HTML5.

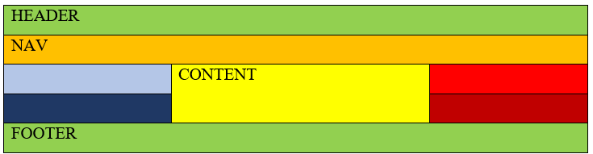
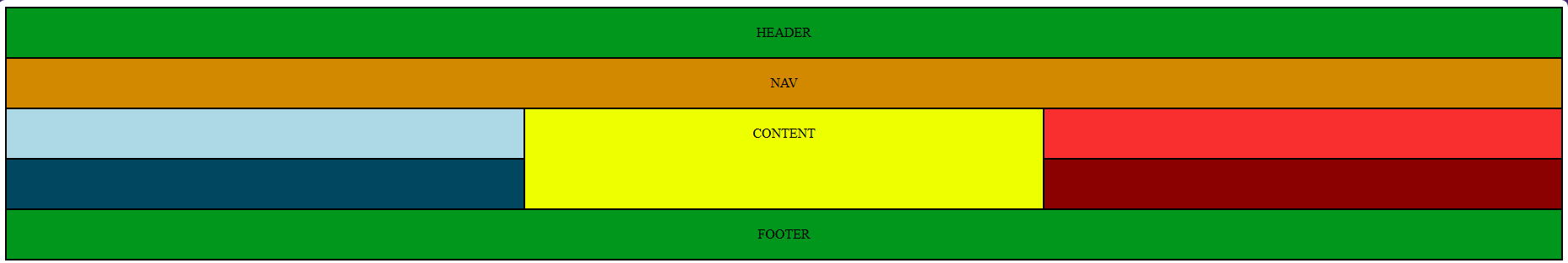


Рисунок – Пример реализации задания

Решение:



task.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link rel="stylesheet" href="/style.css">

    <title>Task 1</title>

</head>

<body>

    <div class="container">

        <div class="header">HEADER</div>

        <div class="nav">NAV</div>

        <div class="content">CONTENT</div>

        <div class="item1"></div>

        <div class="item2"></div>

        <div class="item3"></div>

        <div class="item4"></div>

        <div class="footer">FOOTER</div>

    </div>

</body>

</html>

style1.css:

.container {

    display: grid;

    border: 1px solid black;

    background-color: red;

    width: 100%;

    height: 100%;

    grid-template-columns: 1fr 1fr 1fr;

    grid-template-rows: 1fr 1fr 1fr 1fr 1fr;

}

.item1 {

    border: 1px solid black;

    grid-column: 1;

    grid-row: 3;

    background-color: lightblue;

    padding: 20px;

    text-align: center;

}

.item2 {

    grid-column: 1;

    grid-row: 4;

    background-color: rgb(0, 71, 95);

    padding: 20px;

    border: 1px solid black;

    text-align: center;

}

.item3 {

    grid-column: 3;

    grid-row: 3;

    background-color: rgb(249, 47, 47);

    padding: 20px;

    border: 1px solid black;

    text-align: center;

}

.item4 {

    grid-column: 3;

    grid-row: 4;

    background-color: rgb(139, 0, 0);

    padding: 20px;

    border: 1px solid black;

    text-align: center;

}

.header {

    grid-column: 1 / 4;

    background-color: rgb(0, 151, 28);

    border: 1px solid black;

    padding: 20px;

    text-align: center;

}

.content {

    grid-column: 2;

    grid-row: 3/5;

    background-color: rgb(238, 255, 0);

    border: 1px solid black;

    padding: 20px;

    text-align: center;

}

.nav {

    border: 1px solid black;

    grid-column: 1 / 4;

    grid-row: 2;

    background-color: rgb(210, 137, 0);

    padding: 20px;

    text-align: center;

}

.footer {

    grid-column: 1 / 4;

    grid-row: 5;

    background-color: rgb(0, 151, 28);

    border: 1px solid black;

    padding: 20px;

    text-align: center;

}

Задание 2: С помощью свойства position выполните позиционирование на html-странице трех изображений.

Решение:



task2:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link rel="stylesheet" href="/style2.css">

    <title>Task 2</title>

</head>

<body>

    <div class="container">

        <img class="img1" src="/img/image1.png" width="100" height="100" alt="Изображение 1">

        <img class="img3" src="/img/image2.png" width="100" height="100" alt="Изображение 2">

    </div>

    <img class="img2" src="/img/image3.png" width="100" height="100" alt="Изображение 3">

</body>

</html>

style2.css:

.container {

    position: relative;

    width: 500px;

    height: 300px;

    border: 2px solid black;

}

.img1 {

    position: absolute;

    top: 10px;

    left: 10px;

}

.img2 {

    position: fixed;

    bottom: 20px;

    right: 20px;

}

.img3 {

    position: relative;

    left: 50px;

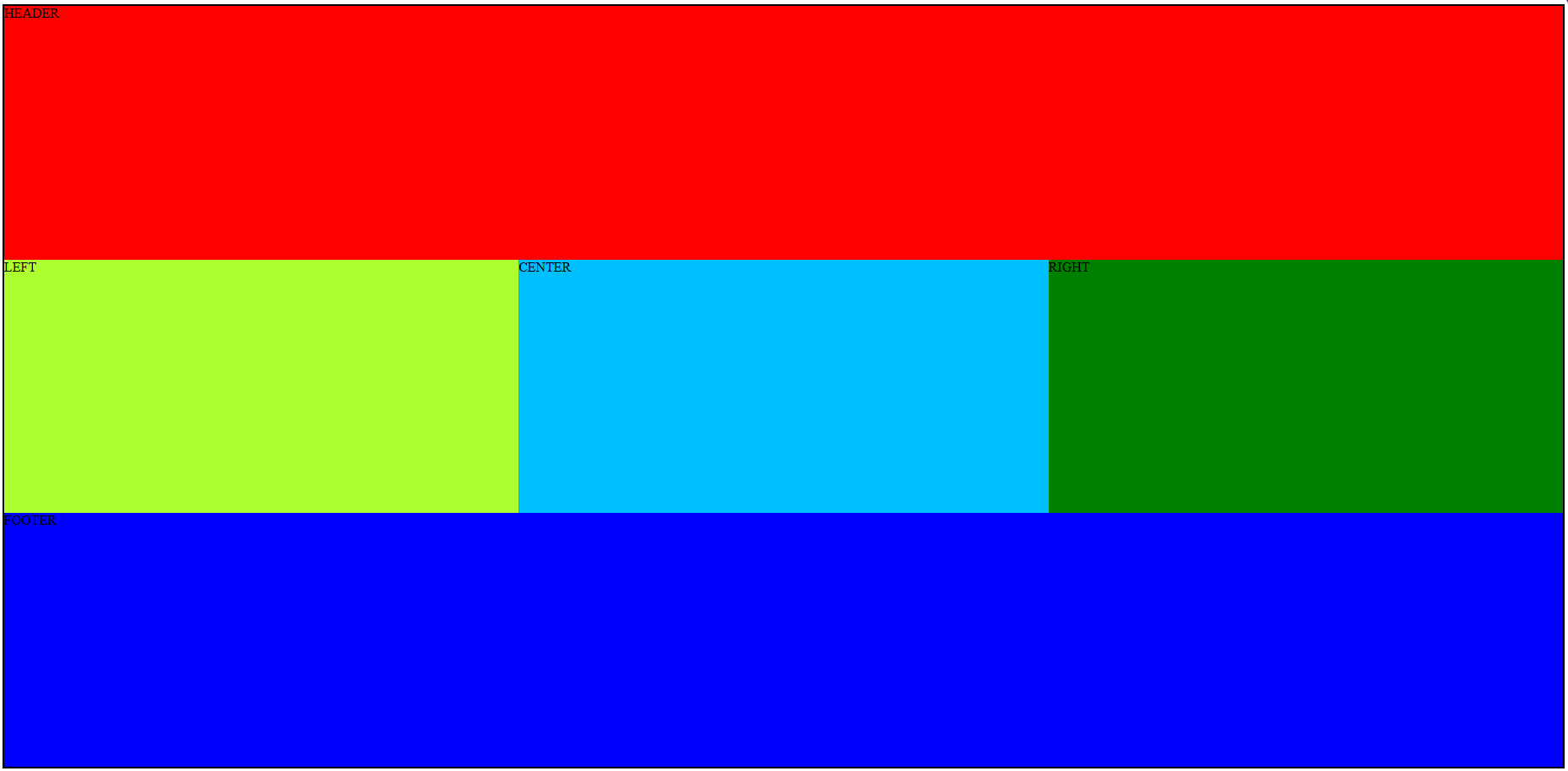
    border: 2px solid black;

}

Задание 3: Выполните задания в соответствии с вариантом, таблица 5.7

|  |  |  |
| --- | --- | --- |
| Вариант | Задание | Пример |
| 2 | 1 Оформите html-страницу с использованием свойства float | ЛР_9. Основы блочной верстки |
| 2 Оформите html-страницу с использованием технологии flex-box |  |
| 3 Оформите html-страницу с использованием технологии grid | Базовые принципы CSS Grid на простых примерах |

Решение 1:



task3.1.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link rel="stylesheet" href="/task 3/style3.1.css">

    <title>Task 3</title>

</head>

<body>

    <div class="container">

        <div class="header">HEADER</div>

        <div class="content">

            <div class="left">LEFT</div>

            <div class="center">CENTER</div>

            <div class="right">RIGHT</div>

        </div>

        <div class="footer">FOOTER</div>

    </div>

</body>

</html>

style3.1.css:

.container {

    width: 100%;

    height: 90vh;

    border: 2px solid black;

    display: flex;

    flex-direction: column;

}

.content {

    display: flex;

}

.header {

    background-color: red;

    height: 30vh;

}

.left {

    background-color: greenyellow;

    height: 30vh;

    width: 33%;

}

.center {

    background-color: deepskyblue;

    height: 30vh;

    width: 34%;

}

.right {

    background-color: green;

    height: 30vh;

    width: 33%;

}

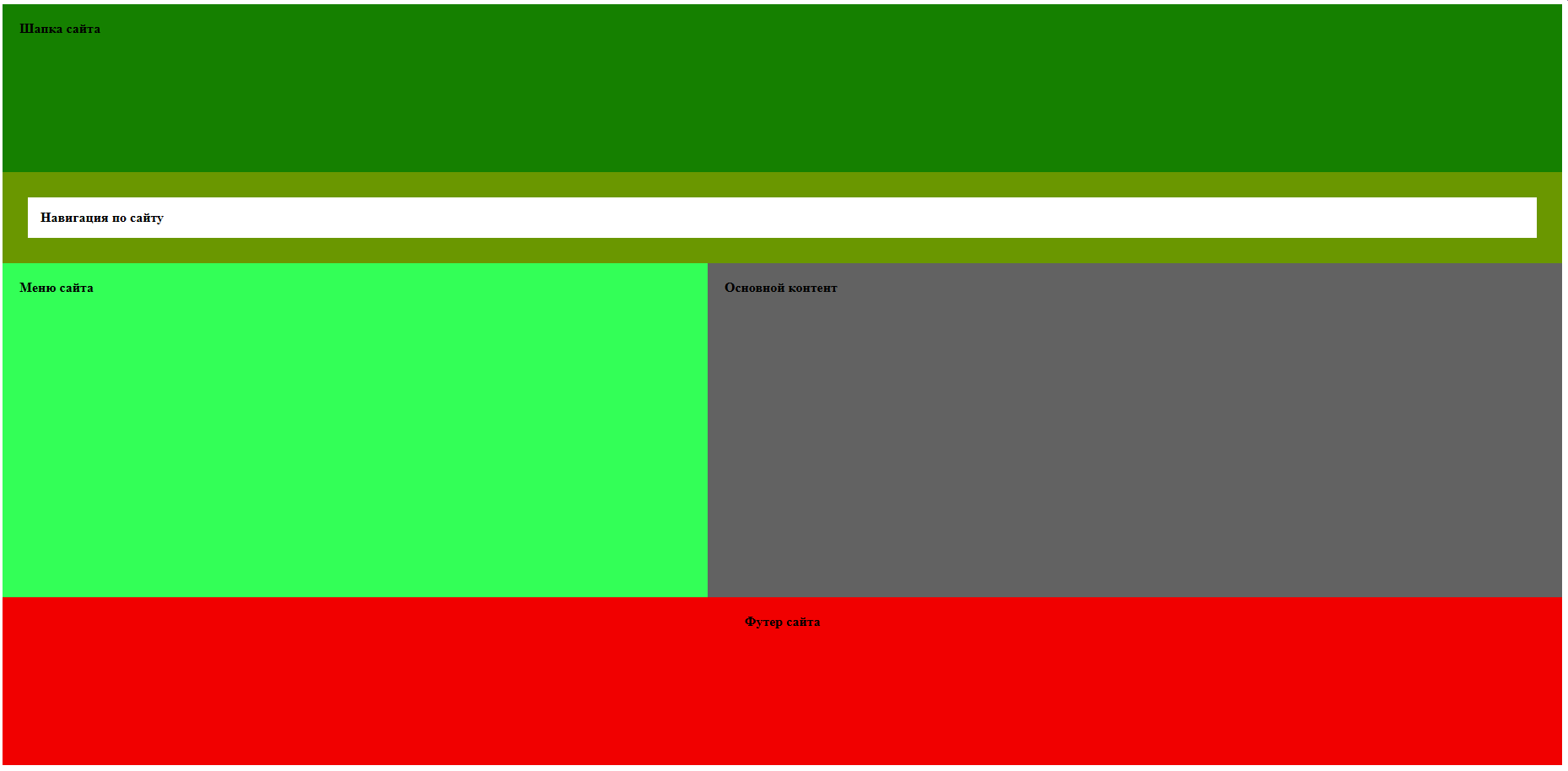
.footer {

    background-color: blue;

    height: 30vh;

}

Решение 2:



task3.1.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link rel="stylesheet" href="/task 3/style3.2.css">

    <title>Task 3</title>

</head>

<body>

    <div class="container">

        <div class="header">Шапка сайта</div>

        <div class="nav">Навигация по сайту</div>

        <div class="minicontainer">

            <div class="menu">Меню сайта</div>

            <div class="content">Основной контент</div>

        </div>

        <div class="footer">Футер сайта</div>

    </div>

</body>

</html>

style3.2:

\* {

    color: black;

    font-weight: bold;

}

.container {

    display: flex;

    flex-direction: column;

    width: 100%;

    height: 90vh;

    background-color: #6a9700;

}

.minicontainer {

    display: flex;

    height: 50vh;

}

.header {

    background-color: #158000;

    padding: 20px;

    height: 20vh;

}

.nav {

    background-color: #fff;

    padding: 15px;

    margin: 30px;

    height: auto;

}

.menu {

    background-color: #33ff57;

    width: 45%;

    padding: 20px;

}

.content {

    background-color: #626262;

    width: 55%;

    padding: 20px;

}

.footer {

    background-color: #f10000;

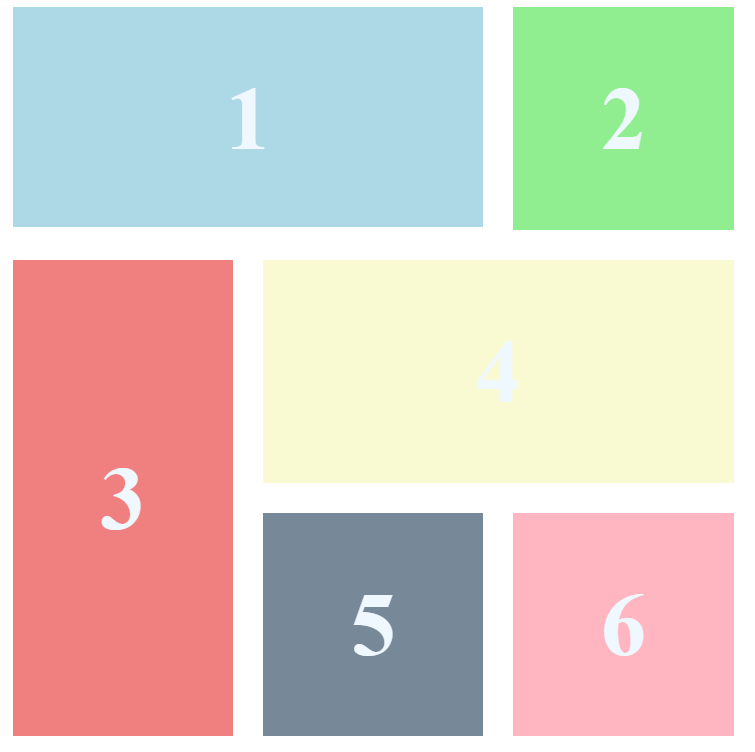
    text-align: center;

    padding: 20px;

    height: 20vh;

}

Решение 3:



task3.3.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <link rel="stylesheet" href="/task 3/style3.3.css">

    <title>Task 3</title>

</head>

<body>

    <div class="container">

        <div class="block block1">1</div>

        <div class="block block2">2</div>

        <div class="block block3">3</div>

        <div class="block block4">4</div>

        <div class="block block5">5</div>

        <div class="block block6">6</div>

    </div>

</body>

</html>

style3.3.css:

.container {

    display: grid;

    justify-content: space-between;

    grid-template-columns: repeat(3, 1fr);

    grid-template-rows: repeat(3, 1fr);

    gap: 10px;

    width: 30vh;

    height: 30vh;

}

.block {

    padding: 20px;

    text-align: center;

    font-weight: bold;

    font-size: 30px;

    color: aliceblue;

}

.block1 {

    background-color: lightblue;

    grid-row: 1;

    grid-column: span 2;

    height: 10vh;

}

.block2 {

    background-color: lightgreen;

    width: 10vh;

}

.block3 {

    background-color: lightcoral;

    grid-column: 1;

    grid-row: span 2;

    display: flex;

    align-items: center;

    justify-content: center;

    width: 10vh;

}

.block4 {

    background-color: lightgoldenrodyellow;

    grid-row: 2;

    grid-column: 2/4;

}

.block5 {

    background-color: lightslategray;

    width: 10vh;

}

.block6 {

    background-color: lightpink;

    width: 10vh;

}

Контрольные вопросы:

1. Перечислите значения свойства position. В чем их различие?

Ответ:

* static – значение по умолчанию. Элемент размещается в потоке документа и не реагирует на свойства top, right, bottom, left.
* relative – элемент остается в потоке документа, но его можно смещать относительно исходного положения с помощью top, right, bottom, left.
* absolute – элемент извлекается из потока документа и позиционируется относительно ближайшего предка с position: relative (или absolute/fixed). Если таких нет, позиционируется относительно body.
* fixed – элемент закреплен относительно окна просмотра (viewport) и не перемещается при прокрутке страницы.
* sticky – элемент ведёт себя как relative, пока не достигает определенного порога (top, right, bottom, left), после чего закрепляется как fixed.

2. Чем отличаются значения свойства position absolute и relative?

Ответ:

* relative сохраняет место в потоке документа, но позволяет смещать элемент от его первоначального положения.
* absolute удаляет элемент из потока, поэтому другие элементы ведут себя так, как будто его нет. Он позиционируется относительно ближайшего родителя с relative.

3. Для чего применяется технология flex-box?

Ответ:

Flexbox (display: flex) используется для упрощенного управления макетом, особенно когда элементы динамически изменяют размер. Позволяет легко:

* Выравнивать элементы вертикально и горизонтально.
* Управлять их порядком (order).
* Равномерно распределять пространство (justify-content).
* Автоматически адаптироваться к изменению размеров контейнера.

4. Охарактеризуйте технологию grid.

Ответ:

CSS Grid – мощная система разметки, позволяющая создавать сложные макеты с рядами и колонками. Основные возможности:

* Управление сеткой через grid-template-columns и grid-template-rows.
* Размещение элементов с помощью grid-column и grid-row.
* Гибкость при адаптации к разным экранам.