1. **неделя:**
2. Самый часто используемый:

margin: auto и display: flex со свойством justify-content: center;

1. **fr** - гибкая единица измерения, которая рассчитывается исходя из количества доступного пространства в grid-контейнере.
2. Grid-template-columns: 20% 20% 20% 20% 20%

Grid-template-columns: repeat (5, 20%).

1. **Auto-fill** – автоматически заполняет строку таким количеством колонок с учетом заданной ширины. Используется вместе с repeat:

grid-template-columns: repeat (auto-fill, 100px) – т.е., каждая строка заполнена возможным количеством колонок, учитывая их длину 100px. Когда в строке нет места, элементы переходят на следующую строку.

Если нужно, чтобы колонки были минимум 100px и максимум 1fr, то колонки получат автоматически размер:

grid-template-columns: repeat (auto-fill, minmax (100px, 1fr)).

**Auto-fit** ведет себя как auto-fill, но любая пустая область схлопывается и растягивает элемент в строке.

<!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">

    <title>Document</title>

    <style>

        .container {

            display: grid;

            grid-template-columns: 100px 30% 1fr;

            grid-template-rows: 200px 100px;

            grid-gap: 20px;

        }

    </style>

</head>

<body>

    <div class="container">

        <div style="background-color: blue;"></div>

        <div style="background-color: blue;"></div>

        <div style="background-color: blue;"></div>

        <div style="background-color: blue;"></div>

        <div style="background-color: blue;"></div>

        <div style="background-color: blue;"></div>

    </div>

</body>

</html>

1. justify-content: space-between.
2. **Grid area** **-** это пространство внутри grid контейнера, в которое может быть помещен один или больше grid элементов. Свойство grid-area - сокращенн свойством для grid-row-start, grid-column-start, grid-row-end и grid-column-end.

8.

<!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">

    <title>Document</title>

    <style>

        .header {

            background-color: aqua;

            grid-area: header;

        }

        .nav {

            background-color: rgb(226, 167, 89);

            grid-area: nav;

        }

        .main {

            background-color: blanchedalmond;

            grid-area: main;

        }

        .aside {

            background-color: aquamarine;

            grid-area: aside;

        }

        .footer {

            background-color: brown;

            grid-area: footer;

        }

        .container {

            display: grid;

            grid-template-columns: 3fr 1fr;

            grid-template-rows: 1fr 1fr 2fr 1fr;

            grid-template-areas:

                "header header header header"

                "nav nav nav nav"

                "main main main aside"

              "footer footer footer footer";

            grid-gap: 20px;

        }

    </style>

</head>

<body>

    <div class="container">

        <header class="header">Header</header>

        <nav class="nav">Nav</nav>

        <main class="main">Main</main>

        <aside class="aside">Aside</aside>

        <footer class="footer">Footer</footer>

    </div>

</body>

</html>

1. auto-fit
2. При именовании линий их имена заключаются в квадратные скобки, а между именами указывается ширина столбца или высота строки, которые находятся между этими линиями (удобно использовать при именовании суффикс –start для начальной линии и –end для конечной линии):

**grid-template-columns: [column-start] 1fr [column-end] 10px**

Используя эти названия, можно позиционировать элементы между определенными линиями (используем вместо цифр):

**.container {**

**grid-column: column-start / column-end;**

**}**

1. Grid-template-columns: repeat (12, 1fr).
2. position: fixed - применяется для создания меню, вкладок, заголовков, т.е., для любых элементов, которые должны быть закреплены на веб-странице и всегда видны посетителю.
3. Все зависит еще от того, какие это блоки текста. Я бы сказал, что стоит использовать любые гибкие способы позиционирования, однако иногда в больших блоках текста вместо гибкости и переносов нужно сокращение текста - в этом случае часто используют троеточие, обрезая текст. В этом случае уже grid и flex не то что не помощники, а совсем не подходят, так как свойство text-overflow: ellipsis работает только с блочными элементами :disappointed: Поэтому тут возможны как flex.grid, так и простая блочная верстка!