1. Данное свойство поддерживается всеми современными браузерами, можно выровнять элементы по центру (как горизонтали, так и вертикали), можно просто изменять параметры, встроенных в контейнер элементов, можно гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров.

2. Подходит для больших контейнеров. Например, боковая панель статической ширины рядом с "гибким" контейнером основного контента. Также таблицы можно использовать, когда нужно получить ряд одинаковых по высоте элементов. Тем не менее, есть много ограничений

3. text-align:center, margin: auto, flexbox (justify-content: center),

4. Главная и поперечная оси.

Направление главной оси можно изменить, осуществляется это с помощью CSS-свойства flex-direction. Выравнивание flex-элементов во flex-контейнере по направлению поперечной оси осуществляется с помощью CSS-свойства align-items

5. margin:auto нужно для того, чтобы отцентрировать блочный элемент по горизонтали, размер отступов будет автоматически рассчитан браузером. Для того чтобы свойство margin: auto позволило бы нам соответствующим образом выровнять элемент, нужно, чтобы была задана ширина и высота элемента.

6. Это свойство позволяет избавиться от костылей в виде лишних div-контейнеров, минимализировать стилизацию, точно вычислять размеры элемента, можно легко смешивать разные значения длины/ширины.

7. flex-grow определяет, может ли начальная ширина flex-элемента увеличиваться. Увеличение ширины flex-элемента осуществляется за счёт свободного пространства линии. flex-shrink определяет определяет, может ли ширина flex-элемента уменьшиться при определенных условиях.

8. justify-content:" space-between”

9. 148.523x19.796

10. CSS свойство order определяет порядок, используемый для размещения flex элементов в их flex контейнере. Элементы располагаются в восходящем порядке по значению order. Элементы с одинаковым значением order располагаются в том порядке, в каком они находятся в исходном коде.

Авторы должны использовать order только для визуального, а не логического, переупорядочивания контента.

Иногда логический порядок макета нельзя описать одним из вариантов обычного потока элементов во флексбоксах: слева направо, справа налево, сверху вниз или снизу вверх. Свойство order — как раз для таких случаев: создать визуальный порядок, соответствующий логическому именно для конкретного дизайна.

11.

  <table border="1" style="width: 55%;">

        <tr>

            <td>1 </td>

            <td>2</td>

            <td>3</td>

        </tr>

        <tr>

            <td rowspan="2">4</td>

            <td>5</td>

            <td>6</td>

        </tr>

        <tr>

            <td>7</td>

            <td>8</td>

        </tr>

        <tr>

            <td colspan="3">9</td>

        </tr>

    </table>

12. Фиксированное позиционирование элемента задаётся значением fixed свойства position и по своему действию похоже на абсолютное позиционирование. Но в отличие от него привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы.

Применяется такой тип позиционирования для создания меню, вкладок, заголовков, в общем, любых элементов, которые должны быть закреплены на веб-странице и всегда видны посетителю.

Чаще всего применяется для создания элемента, который «возвращает» пользователя на верх страницы.

13. Блочная модель по умолчанию

14. Размеры флекс-элементов рассчитываются как в обычной блочной модели: рамки + отступы + размеры содержания