1. в Flexbox позиционирование элементов проще, а следовательно, можно создавать более сложные раскладки меньшим кодом, что облегчает разработку. Алгоритм flexbox-раскладки основан на направлении, в отличие от блочной или строчной раскладки, которые основаны на вертикали и горизонтали.

2. CSS-свойство display: table и другие, делают вывод группы элементов подобно таблице <table>, но с ограничением – объединения ячеек colspan и rowspan не поддерживаются.Таким образом использовать верстку на основе таблиц не рационально.
Таблицы лучше использовать по их прямому назначению - для отображения таблиц. Табличную верстку можно применять, например, при печати странички.

3. У flex-контейнера есть две оси: главная и перпендикулярная ей. По умолчанию все предметы располагаются вдоль главной оси — слева направо. flex-direction позволяет вращать главную ось.

4. Свойство margin: auto позволяет центрировать элементы внутри контейнера.Когда ставится свойство margin auto к элементу во флексбоксе, оно автоматически устанавливает значение margin для заданной стороны элемента на авто, что приводит к распределение свободного пространства по обеим сторонам элемента и центрирует его в соответствии с направлением оси флексбокса. Например, если направление оси установлена row, то при margin: auto, элемент будет центрирован по горизонтали. Если направление оси - column, то - по вертикали.Когда ставится свойство margin auto к элементу во флексбоксе.

5. При помощи свойства box-sizing можно изменить то, как браузер будет рассчитывать размеры элемента. По умолчанию размером элемента считается размер контентной области. Если кроме width и height указать ещё и padding с border то браузер посчитает размер элемента как width + padding * 2 + border * 2 и height + padding * 2 + border * 2. Если задать значение border-box для свойства box-sizing, то браузер изменит принцип расчёта и padding с border уже будут включены в width и height.

6.Это свойство определяет, может ли начальная ширина flex-элемента расти. Увеличение ширины flex-элемента осуществляется за счёт свободного пространства линии. В качестве значения CSS-свойства flex-grow указывается целое число. Контролировать возможность элемента ужиматься в зависимости от размеров flex-контейнера помогает свойство flex-shrink. Если свойство flex-grow помогало нам распределить свободное пространство между элементами, то flex-shrink помогает распределить пространство, если места в контейнере недостаточно. Необходимая ширина рассчитывается на основании начальной ширины,

которую имеет каждый flex-элемент в ней.

7. С помощью создания контейнера а в нем поместить 4 блока например div. И в css для контейнера применить свойство display: flex.

8. ~141*26

9. Свойство order отвечает за порядок расположения элементов внутри контейнера. Это удобная функция, позволяющая, например, перемещать элементы внутри контейнера при открытии страницы на разных устройствах. В качестве значения order может принимать любое число, которое укажет номер расположения элемента по главной оси. Также возможно использование отрицательного значения. В таком случае никакой магии не произойдет, просто элементы с отрицательным значением order будут находиться раньше элементов с положительным значением. Если свойство order указано не у всех элементов, то элементы без этого свойства будут выведены в месте, согласно своему расположению внутри документа. Причем первыми всегда будут выведены те элементы, у которых свойство order отсутствует.

10.
```
<table>

    <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
    </tr>
    <tr>
        <td rowspan="2">Row 2 &amp;amp; 3</td>
        <td>Row 2, Column 2</td>
        <td>Row 2, Column 3</td>
    </tr>
    <tr>
        <td>Row 4, Column 2</td>
        <td>Row 4, Column 3</td>
    </tr>
    <tr>
        <td colspan="3">Row 5</td>
    </tr>
</table>
```

11. Размер флексконтейнера рассчитывается на основе заданных размеров элементов и свойств flex-grow, flex-shrink, flex-basis.