1. Используется меньше кода, проще для понимания. Можно с легкостью менять правила свойств элементов. Можно расположить элементы так как надо. Поддерживается всеми современными браузерами
2. Для демонстрации табличных данных
3. text-align: center; .flexbox {     display: flex; justify-content: center;};   
   margin-left: auto; margin-right: auto;  
   align-self: center
4. Какие есть оси во флекс-верстке и как задается их направление? Главная и поперечная. С помощью свойства flex-direction: задается направление. Дополняется row , row-reverse; column или column-reverse. Есть еще flex-wrap:, можно задать направление по поперечной оси
5. Позволяет поместить элемент в центр, позиция сохраняется при изменении размеров контейнера. Применив это свойство, можно например сделать так, чтобы один элемент из многих занял все свободное место
6. Свойство задает не размеры контента, а самого блока с помощью ширины и высоты
7. Flex-grow-определяет, можно ли будет элементу увеличиваться за счет свободного места в ряду. Flex-shrink наоборот, определяет, смогут ли элементы уменьшатся, если для всех запланированных элементов не будет хватать места в ряду
8. Вроде подходит justify-content: space-around или space-between;, склоняюсь ко второму варианту
9. Ширина-1.989х2+10+134.545=148.523=149  
   Высота-1.989+4+11.818=17.807=18
10. С его помощью можно выстраивать элементы в нужном порядке в контейнере, задав числовое значение
11. <table border="1">
12. <tr>
13. <td>Columu 1</td>
14. <td>Columu 2</td>
15. <td>Columu 3</td>
16. </tr>
17. <tr>
18. <td rowspan="2">Row 1 Cell 1</td>
19. <td>Row 1 Cell 2</td>
20. <td>Row 1 Cell 3</td>
21. </tr>
22. <tr>
23. <td>Row 2 Cell 2</td>
24. <td>Row 2 Cell 3</td>
25. </tr>
26. <tr>
27. <td colspan="3">Row 3 Cell 1</td>
29. </tr>
30. </table>
31. Используется, если необходимо, чтобы элемент оставался на месте и был виден при прокрутке страницы вниз
32. Блочная модель
33. Его размер на странице будет таким, чтобы поместились все запланированные элементы