**1. Почему в большинстве ситуаций стоит использовать flexbox-позиционирование?**

Он позволяет быстро и удобно выравнивать элементы относительно друг друга, распределять пространство между элементами. Создать гибкий макет.

**2. Самостоятельно изучите способ позиционирования через `display: table` и ответьте на вопрос, для каких ситуаций оно лучше всего подходит?**

А) Для вёрстки таблиц

Б) Для вёрстки некоторых функций сайта (например, прилипающего подвала), если предполагается, что основными посетителями сайта будут пользователи со старыми версиями браузеров.

**3. Какими способами можно сделать горизонтальное выравнивание по центру? Минимум 3 варианта, можно больше**

А) присвоить элементу text-align: center

Б) с помощью margin настроить отступы, чтобы текст оказался по центру

В) поставить блоку ширину на весь экран, с помощью padding настроить отступ, чтобы текст был по центру

Г) присвоить элементу position: absolute или position: relative, с помощью left/ right выставить элемент по центру

Д) присвоить родительскому блоку display: flex, с помощью justify-content: center расположить элемент по центру.

**4. Какие есть оси во флекс-верстке и как задается их направление?**

Во флекс-вёрстке есть две оси: главная и поперечная

Для свойства flex-direction: row, они расположены так → (гл) ↓ (п)

Для свойства flex-direction: row-reverse, они расположены так ← (гл) ↓ (п)

Для свойства flex-direction: column, они расположены так ↓ (гл) → (п)

Для свойства flex-direction: column-reverse, они расположены ↑ (гл) → (п)

**5. Разберитесь, как работает свойство `margin: auto` во флекс-верстке, приведите пример использования**

Margin-auto мы можем присвоить одному из flex-элементу, тогда он займёт всё доступное свободное место. Пример: разделить логотип и блоки меню по разным сторонам.

**6. В чем преимущества `box-sizing`?**

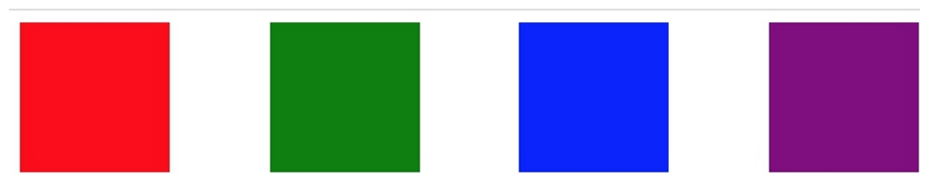
Можем присвоить ему box-sizing: border-box. Это упростит нам подсчёт размеров блока. Border и padding входят в размеры блока. Мы сразу понимаем, что общий размер блока не превысит указанный нами.

**7. Чем отличается `flex-grow` от `flex-shrink`?**

flex-grow может ли ширина flex элемента увеличиться (занять свободное пространство) и на сколько

flex-shrink может ли ширина flex элемента уменьшаться (если ширины линии недостаточно для отображения всех flex элементов)

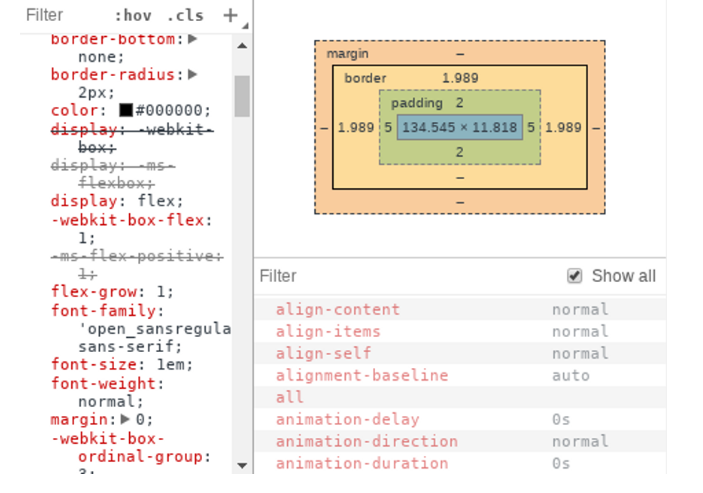
**8. Как можно добиться следующего позиционирования элементов:**



Присвоить родительскому блоку display: flex

Justify-content: space-between

**9. Какой в итоге будет размер у элемента (можно округлить)?**



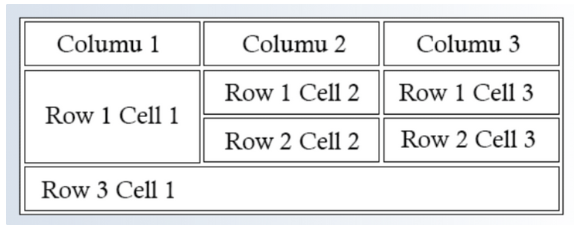
Высота: 17,807

Ширина: 148,523

**10. Самостоятельно разберитесь, зачем нужно свойство `order`?**

Помогает изменить порядок визуального отображения элементов. Только визуальное отображение, на first-child и тд не работает.

**11. Каким кодом можно сделать такую таблицу?**



    <body>

        <table border="1">

            <tbody>

                <tr>

                    <td>Columu 1</td>

                    <td>Columu 2</td>

                    <td>Columu 3</td>

                </tr>

                <tr>

                    <td rowspan="2">Row 1 Cell 1</td>

                    <td>Row 1 Cell 2</td>

                    <td>Row 1 Cell 3</td>

                </tr>

                <tr>

                    <td>Row 2 Cell 2</td>

                    <td>Row 2 Cell 3</td>

                </tr>

                <tr>

                    <td colspan="3">Row 3 Cell 1</td>

                </tr>

            </tbody>

        </table>

    </body>

**12. Изучите материалы и найдите, для каких ситуаций подходит `position: fixed`?**

Для кнопки прокрутки страницы вверх, для блока меню всегда остающегося на экране (н-р: header у ВК)

**13. Каким способом лучше всего верстать большие блоки текста?**

Использовать блочную вёрстку и добавлять элементы, обтекаемые текстом, н-р, картинки с присвоенным float

**14. Как рассчитывается размер flex-контейнера?**

Padding с двух сторон + border с двух сторон + размер содержимого