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

С помощью флексбокса можно легко настраивать расположение элементов в контейнере, выравнивать их по горизонтали и вертикали, создавать вложенные контейнеры и сложные элементы. И flexbox поддерживается всеми современными браузерами.

1. Самостоятельно изучи способ позиционирования через display: table и ответьте на вопрос, для каких ситуаций оно лучше всего подходит?

display: table можно использовать, когда надо создать элементы на странице, которые будут вести себя как таблица с ячейками и строками, с равномерным распределением контента по ячейкам.

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

Есть 2 оси: главная и поперечная. Главная ось определяет направление элементов в контейнере, может быть вертикальной или горизонтальной. Т.е если основная ось горизонтальная, то поперечная – вертикальная, и наоборот. Направление задается с помощью свойства flex-direction. У него есть несколько значений: column (колонка в обычном порядке), column-reverse (колонка в обратном порядке), row (обычный ряд), row-reverse (ряд в обратном порядке).

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

Если у элемента будет задано свойство margin: auto, то это приведет к центрированию элемента по горизонтали, т.к. свободное пространство будет распределяться по обеим сторонам от элемента.

1. В чем преимущества box-sizing?

С помощью box-sizing можно управлять размером блока, задавать ширину блока. Можно выравнивать и размещать элементы в пределах заданной ширины. Еще свойство box-sizing учитывает паддинги и границы, благодаря чему в блоке будет достаточно пространства для отображения контента.

1. Чем отличается flex-grow от flex-shrink?

flex-grow помогает распределить свободное пространство между элементами, а flex-shrink помогает распределить пространство, если в контейнера недостаточно места.

1. Как можно добиться следующего позиционирования элементов:

.container {

display: flex;

justify-content: flex-end;

}

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

148,5 x 15,8

1. Самостоятельно разберись, зачем нужно свойство order?

Это свойство отвечает за порядок расположения элементов внутри контейнера. Этот порядок можно изменить, задавая численное значение элементу. Значения могут быть положительными и отрицательными.

1. Каким кодом можно сделать такую таблицу?

<body>

    <table border="1" style="width: 50%">

      <tr>

        <td>Column 1</td>

        <td>Column 2</td>

        <td>Column 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>

    </table>

</body>

1. Как рассчитывается размер flex-контейнера?

Размер flex-контейнера рассчитывается на основе размеров его дочерних элементов и настроек свойств flex. Но в контейнере могут быть элементы, размеры которых не заданы явно, а определяются с помощью свойств flex-grow, flex-shrink, flex-basis. flex-grow указывает на то, какую долю свободного пространства нна главной оси должен занимать каждый дочерний элемент. flex-shrink указывает на то, какую долю свободного пространства на главной оси должен занимать каждый дочерний элемент при нехватке места. flex-basis задает размер элемента до распределения доступного пространства и может быть задано как абсолютное значение или процентное значение отн-го контейнера. Итоговый размер контейнера будет зависеть от суммарного размера его элементов и значения свойства flex-wrap. Если flex-wrap установлено Nowrap, то контейнер имеет размер, достаточный для размещения всех элементов в одной строке или столбце. Если wrap или wrap-reverse, то контейнер будет растянут в соответствии с размерами его элементов и перенесет их на доп.строки или столбцы.