**Домашнее задание недели 5**

**Вопросы 💎**

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

flexbox-позиционирование является самым простым и удобным вариантом в современном мире и именно оно является официально рекомендованным вариантом от World Wide Web Consortium (W3C). Flexbox позволяет создать адаптивный дизайн намного проще, чем с использованием других видов позиционирования.

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

Если разобраться, добиться табличных значений можно двумя способами: через HTML код и через CSS-свойства. Разница состоит в том, что в первом случае, изменение данных с табличных на обычные невозможно без переписания HTML-кода, в то время как во-втором, достаточно изменить CSS-свойство. Благодаря ему, можно:

* реализовать динамическое вертикальное выравнивание по центру:

- реализовать динамическое горизонтальное выравнивание по центру;

- реализовать динамический прилипающий подвал.

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

Основная ось (слева – направо)

Поперечная ось (сверху-вниз)

Направление задается через flex-direction

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

Значение margin-left:auto/margin-right:auto заставляет браузер выделять под margin всё доступное сбоку пространство. А если и то и другое auto, то слева и справа будет одинаковый отступ, таким образом элемент окажется в середине.

Однако, это работает только для горизонтальных отступов. Но не будет работать для плавающих и строчных элементов. А также для абсолютно и фиксировано позиционированных элементов.

Важно, чтобы ширина колонки была ограничена каким-то размером, поскольку если она будет 100% или 100vw, то отступать будет некуда, т.к. не будет свободного места.

Пример:

**HTML:**

<body>

    <h1 class='h1'>Hello, world!</h1>

</body>

**CSS:**

.h1 {

  width: 80vw;

  margin: 0 auto;

  text-align: center;

  background-color: rgb(236, 195, 233);

}

**5. В чем преимущества box-sizing?**

Он дает возможность задавать исключения и поддерживается почти всеми браузерами. Свойство box-sizing позволяет нам понять, каким образом будут вычисляться размеры элемента. Если присвоено значение content-box, то размер вычисляется width + height + padding + border/ margin. Если значение не присвоено, то оно по умолчанию является content-box. В противном случае, присваивается значение border-box, которое существенно упрощает нам математику при вычислении размеров, поскольку общий размер элемента уже включает width, height, border и padding.

Примечание: какое бы свойство не было выбрано, любые значения margin не будет включен в общий размер вычисления

**6. Чем отличается flex-grow от flex-shrink?**

Flex-grow определяет, займет ли элемент свободное пространство внутри контейнера (расшириться), может принимать любые целочисленные значения. Если значение = 0, то элемент не сможет «захватывать» свободное пространство. Если 1 и более – сможет.

Flex-shrink действует по аналогичному принципу с flex-grow, только в сторону уменьшения. При значении 0, элемент не будет уменьшаться, при значении 1 и более – будет.

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

Display: flex;

Justify-content: space-between;

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

Ширина:149, высота: 18

**9. Самостоятельно разберитесь, зачем нужно свойство order?**

При помощи свойства order можно менять порядок отображения флекс-элементов внутри флекс-контейнера.

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

**HTML**

<body>

    <table class="table">

        <tr>

            <td>Columu1</td>

            <td>Columu2</td>

            <td>Columu3</td>

        </tr>

        <tr>

            <td rowspan="2">Row 1 Cel l1</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>

**CSS**

.table, td {

  border: 1px solid black

}

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

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