**Домашнее задание недели 6**

**Вопросы 💎**

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

justify-content: center;

justify-items: center;

align-items: center;

если я неправильно ответила, пожалуйста, поправьте, я на самом деле не очень хорошо разобралась с выравниванием у грядов и для меня эти свойства у грядов работают не так предсказуемо как у флексов

1. **Что за единица измерения - fr?**

Это гибкая единица измерения, которая условно делить область грид-контейнера на части. Если, допустим мы пропишем для какого-либо контейнера значения: 1fr 2fr 1fr – это будет означать, что экран поделен на 4 части, первая и последняя при этом будут занимать лишь по ¼, в то время как средняя область 2/4.

1. **Как можно задать грид с 5 колонками шириной по 20%? Минимум 2 способа.**

grid-template-columns: repeat(5, 20%) или grid-template-columns: 20% 20% 20% 20% 20%

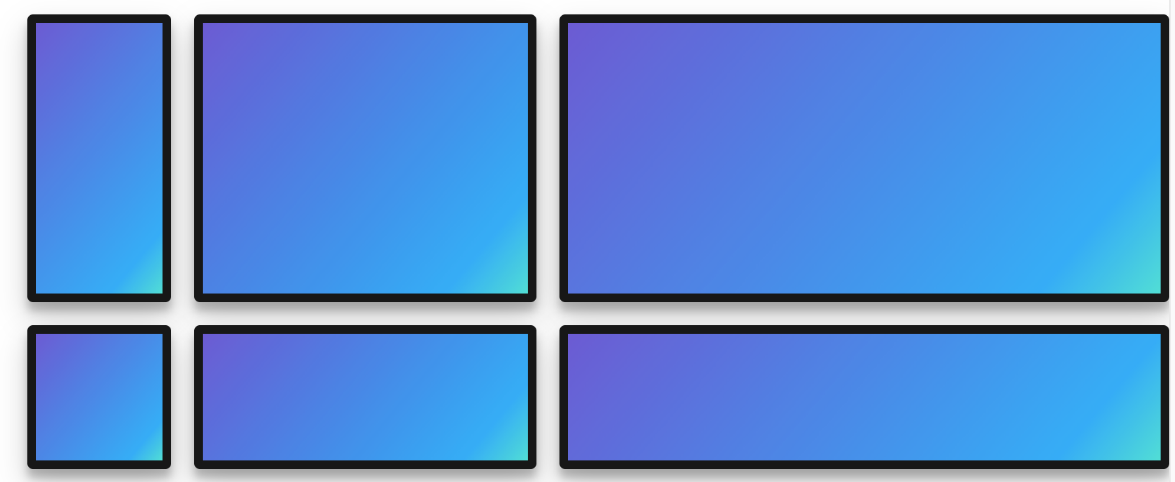
grid-template-columns: 1fr 1fr 1fr 1fr 1fr или grid-template-columns: repeat( 5, 1fr)

1. **Самостоятельно разберитесь, что такое auto-fill и auto-fit ?**

Это инструмент, который позволяет обозначить, как поведут себя элементы внутри контейнера, в зависимости от выбранного auto-fill и auto-fit. Разница между ними заключается в том, что при применении первого, элементы заполняют контейнер максимально возможным количеством колонок. Когда места перестает хватать – следующий элемент переносится на следующую строку, а место, которое не было заполнено в первой строке – остается пустым. И т.д.

По очень похожему принципу работает и auto-fit, разница лишь в том, что после переноса не уместившегося элемента, элементы первой (и последующих строк) растягиваются для заполнения пространства, которое не было заполнено.

1. **Как сделать такую табличку? Параметры: первая колонка шириной 100 пикселей, вторая 30%. Первая строчка высотой 200 пикселей, вторая строчка 100 пикселей**.



**HTML:**

  <div class="parent">

        <div></div>

        <div></div>

        <div></div>

        <div></div>

        <div></div>

        <div></div>

    </div>

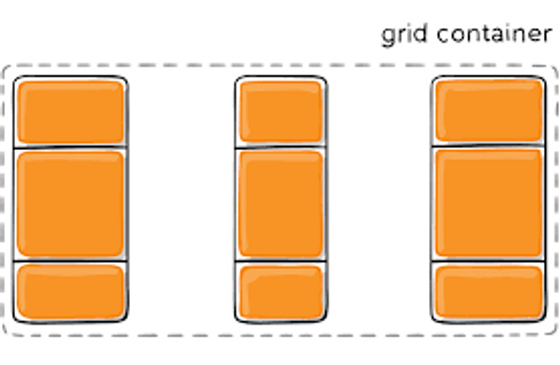
**CSS:**

  display: grid;

  grid-template-columns: 100px 30% 1fr;

  grid-template-rows: 200px 100px;

1. **Как сделать такое выравнивание в грид-контейнере?**



Необходимо задать грид контейнеру свойство justify-items: center;

1. **Что такое и как задается *grid area*?**

Grid-area – это область, ограниченная четырьмя грид-линиями. Может состоять из любого количества ячеек как по горизонтали, так и по вертикали.

1. Приведите пример использования grid-template-areas (не копированием из этого урока 😉)
2. .container {
3. display: grid;
4. grid-template-columns: 3fr 1fr;
5. grid-template-rows: 1fr 1fr 1fr;
6. grid-template-areas:
7. "header header header header"
8. "maim main aside aside "
9. "footer footer footer footer";
10. border: 1px solid red;
11. width: 500px;
12. }
13. header {
14. display: grid;
15. grid-area: header;
16. border: 1px solid black;
17. }
18. main {
19. display: grid;
20. grid-area: main;
21. border: 1px solid black;
22. }
23. aside {
24. display: grid;
25. grid-area: aside;
26. border: 1px solid black;
27. }
28. footer {
29. display: grid;
30. grid-area: footer;
31. border: 1px solid black;
32. }

<body>

    <div class='container'>

        <header>[шапка сайта]</header>

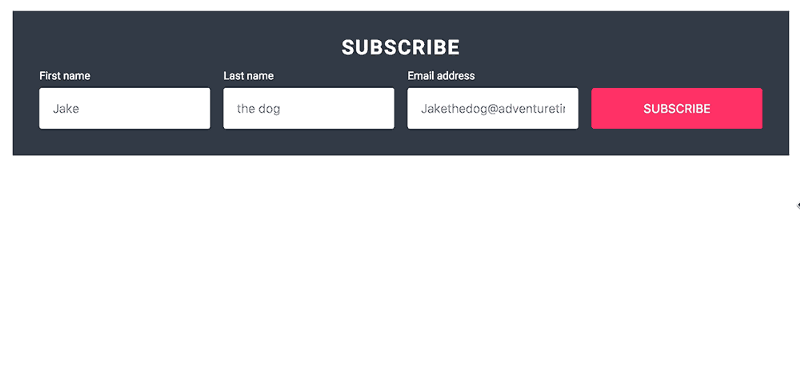
        <main>[основной контент]</main>

        <aside>[боковая область]</aside>

        <footer>[footer]</footer>

    </div>

</body>

8. Каким свойством можно задать такое поведение элементов?

Можно задать контейнеру Auto-fit, указав каким размером должны быть ячейки.

9. Самостоятельно разберитесь, как работают именованные линии? Есть ли какие-то рекомендованные правила наименований? Если да, то какие?

Да, имена линиям задаются в квадратных скобках, и хорошей практикой считается использовать суффиксы –start и end для начальных линий и конечных соответственно. У линии может быть несколько имен – это нормально. По именам можно обращаться к линиям точно также, как и по номерам строк и столбцов. Имя может быть любым, за исключением ключевого слова span.

10. **Как проще всего задать 12 одинаковых по ширине колонок?**

Проще всего это сделать используя repeat: grid-template-columns: repeat (12, 1fr)

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

position: fixed наибольшим образом подходит для ситуаций, когда нам необходимо, чтобы при прокрутке страницы, элемент оставался на одном и том же месте. Это может быть логотип или кнопка»наверх» или любая другая информация, которая не должна теряться, когда пользователь скроллит страницу.

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

Лучше всего это делать через display: grid.