**Вопросы 💎**

1. Что за единица измерения - fr?

Это гибкая единица измерения, которая рассчитывается исходя из количества доступного пространства в grid-контейнере.

Это значит, что задав список треков `1fr 1fr 1fr`, доступное пространство делится на 3 и распределяется поровну между тремя колонками. Если же указать список треков `2fr 1fr 1fr`, доступное пространство делится на 4, из которых 2 части отходят первой колонке, и по одной части для второй и третьей колонки.

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

grid-template-columns: 20% 20% 20% 20% 20%;

grid-template-columns: 1fr 1fr 1fr 1fr 1fr;

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

Помогают в создании адаптивного шаблона. Эти два слова могут использоваться как значения при указании ширины колонок в CSS Grid.

auto-fill

Вместо того, чтобы указывать количество колонок и сколько раз им повторяться, мы просто можем сказать браузеру, чтобы он уместил как можно больше колонок с учетом указанной длины.

auto-fill как бы говорит “я автоматически заполню строку таким количеством колонок, как это возможно с учетом заданной ширины”. auto-fill используется в связке с repeat() таким образом:

grid-template-columns: repeat(auto-fill, 100px);

# minmax

minxmax() функция в CSS, которую очень удобно использовать в связке с auto-fill. Она позволяет вам указывать минимальное и максимальное значение одновременно. Если нам надо, чтобы колонки были минимум в 100px и максимум в 1fr, то колонки автоматически получат размер, чтобы заполнить строку:

grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));

Браузер уместит в строку столько колонок с минимальной шириной в 100px, сколько он сможет. А если останется ещё место, то оно будет поровну распределено между всеми автозаполненными колонками.

Явное различие появится тогда, когда вы укажете браузеру, что делать с оставшимся местом в строке.

# minmax

grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));

auto-fit будет вести себя почти как auto-fill, но если есть только одна строка и есть место для дополнительных колонок, но нет элементов, которые можно вставить эти колонки, то тогда он сделает ширину пустых колонок равную нулю. А если выставить максимальную ширину колонок на 1fr, то браузер разделит оставшееся место поровну между ними.

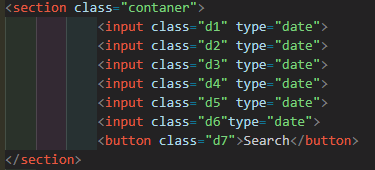
1. Как сделать такую табличку? Параметры: первая колонка шириной 100 пикселей, вторая 30%. Первая строчка высотой 200 пикселей, вторая строчка 100 пикселей.
2. display: grid;
3. grid-template-columns: 100px 30% 1fr;
4. grid-template-rows: 200px 100px;
5. Как сделать такое выравнивание в грид-контейнере?
6. justify-content: space-between;
7. Что такое и как задается *grid area*?

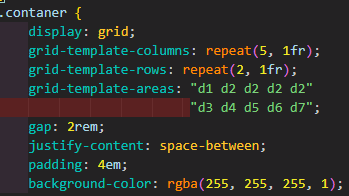
Свойство **grid-area** даёт название элементу чтобы можно было ссылаться на него с помощью шаблона созданного через [grid-template-areas](https://hcdev.ru/css/grid-template-areas/) свойство.

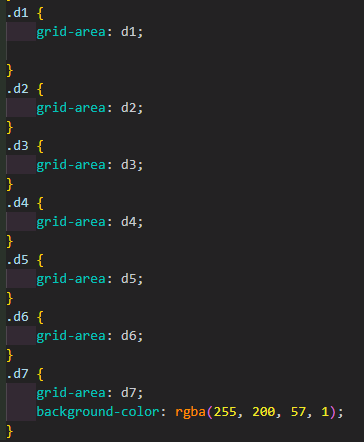
Свойство **grid-area** является сокращенным свойством для [grid-row-start](https://hcdev.ru/css/grid-row-start/), [grid-column-start](https://hcdev.ru/css/grid-column-start/), [grid-row-end](https://hcdev.ru/css/grid-row-end/) и [grid-column-end](https://hcdev.ru/css/grid-column-end/), определяя размер и расположение элемента сетки.

С помощью свойства [grid-area] мы можем назначить каждой из этих областей свое собственное имя. Именование областей еще не создает никакого макета, однако теперь у нас есть именнованные области, которые мы можем в нем использовать.

1. Приведите пример использования grid-template-areas (не копированием из этого урока 😉)







Результат



1. Каким свойством можно задать такое поведение элементов?

grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));

Так как любая пустая область схлопывается и растягивает элементы в строке

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

 Имена линиям можно задать внутри квадратных скобок в списке треков. Здесь важно помнить, что вы именуете линию, а не трек, который следует после неё. Назвав линии, вы можете использовать их имена вместо номеров при позиционировании элементов на сетке.

Имя может быть любым, кроме ключевого слова span. Хорошей практикой является именование с использованием суффикса -start для начальных линий (не важно, это линия строки или колонки), и -end для конечных. В итоге должно получиться main-start и main-end или sidebar-start и sidebar-end.  
  
Довольно часто конечная линия одной части сетки и начальная линия другой части совпадают, но это не проблема, так как линии могут иметь несколько имён. Задать линии несколько имён можно, добавив их через пробел внутри квадратных скобок.

 Как правило, линии могут быть полезны в ситуациях, когда вы хотите изменить структуру сетки в рамках медиа-запроса. Вместо того, чтобы отслеживать, на какой линии вы размещаете элементы на разных контрольных точках, можно будет оперировать последовательно именованными линиями.

1. Как проще всего задать 12 одинаковых по ширине колонок?

grid-template-columns: repeat(12, 1fr);