### Вопросы 💎

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

**text-align, margin: auto, position: absolute, flexbox, grid**

1. Что за единица измерения - `fr`?

**Fr — дробная единица гибкости. 1fr равна 1 части доступного пространства. Благодаря fr вам больше не нужно пересчитывать ширину разделов. Единица гибкости fr берет 100% экрана и равномерно распределяет его между элементами сетки**

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

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

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

**auto-fill стремится поместить в строку как можно больше ячеек. Это свойство предпочитает использовать минимальную ширину для ячейки, даже если есть свободное место. auto-fit стремится занять всё доступное пространство в строке и подстраивает ширину ячеек под это пространство**

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

 grid-template-columns: 100px 30%;

  grid-template-rows: 200px 100px;

1. Как сделать такое выравнивание в грид-контейнере?

**justify-content и align-content space-between**

![https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9530286c-633a-44dd-a8e0-543d51d669d6/Untitled.png](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9530286c-633a-44dd-a8e0-543d51d669d6/Untitled.png)

7. Что такое и как задается \*grid area\*?

**CSS-свойство grid-area - это**[**сокращённая форма записи**](https://developer.mozilla.org/ru/docs/Web/CSS/Shorthand_properties)**для свойств**[**grid-row-start**](https://developer.mozilla.org/ru/docs/Web/CSS/grid-row-start)**,**[**grid-column-start (en-US)**](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-column-start)**,**[**grid-row-end (en-US)**](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-row-end)**и**[**grid-column-end (en-US)**](https://developer.mozilla.org/en-US/docs/Web/CSS/grid-column-end)**. Определяет размер и местоположение грид-элемента в пределах [grid row (en-US)](https://developer.mozilla.org/en-US/docs/Glossary/Grid_Row" \o "Currently only available in English (US)). Задаёт края [грид-области (en-US)](https://developer.mozilla.org/en-US/docs/Glossary/Grid_Areas" \o "Currently only available in English (US)) грид-элемента.**

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

**Свойство grid-template-areas определяет шаблон сетки ссылаясь на имена областей, которые заданы с помощью свойства grid-area.**

**Повторение названия области приводит к тому, что содержимое охватывает эти ячейки. Точка означает пустую ячейку. Сам синтаксис предоставляет визуализацию структуры сетки.**

**grid-template-areas:**

**"a a ."**

**"a a ."**

**". b c";**

9. Каким свойством можно задать такое поведение элементов?

**Auto-fit**

![http://css-live.ru/Primer/grid1/1-MyO6enrNZEL9HSzYmdA8VA.gif](http://css-live.ru/Primer/grid1/1-MyO6enrNZEL9HSzYmdA8VA.gif)

Ссылка на анимацию: [http://css-live.ru/Primer/grid1/1-MyO6enrNZEL9HSzYmdA8VA.gif](http://css-live.ru/Primer/grid1/1-MyO6enrNZEL9HSzYmdA8VA.gif)

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

**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 получаться main-start и main-end или sidebar-start и sidebar-end. Линии могут иметь несколько имён. Задать линии несколько имён можно, добавив их через пробел внутри квадратных скобок.**

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

**grid-template-columns: repeat(12, 20px)**

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

**для создания меню, вкладок, заголовков, в общем, любых элементов, которые должны быть закреплены на веб-странице и всегда видны посетителю.**

13. Каким способом лучше всего верстать большие блоки текста?

**Свойство overflow-wrap позволяет сообщить браузеру о том, что он должен разорвать слово, перенеся его на новую строку, в том случае, если слово не помещается в контейнер.**

**Значение auto CSS-свойства hyphens позволяет сообщить браузеру о том, что он должен самостоятельно принять решение о разделении длинных слов с использованием дефиса и о переносе их на новые строки. Это свойство может принимать и значение manual, что позволяет, используя особые символы, предусмотреть возможность и порядок переноса слова на новую строку в том случае, если в этом возникнет необходимость. При использовании свойства hyphens: auto браузер способен разрывать даже такие слова, которые не выходят за пределы контейнеров.**

**Обрезка однострочного и многострочного текста.**

**В некоторых ситуациях непрактично организовывать разрыв слов или их перенос на новые строки с использованием дефиса. Например, если имеется поле, в котором должен выводиться фрагмент JavaScript-кода, такой код будет тяжело читать в том случае, если слова будут выводиться с переносом на новые строки. В подобном случае облегчить работу с текстом можно, предусмотрев возможность горизонтальной прокрутки содержимого поля.**

**Поле для вывода содержимого статей.**