1. Горизонтальное выравнивание по центру:
2. Если элемент строчный или блочно-строчный, то через свойство text-align: center;
3. Через флексы и свойство justify-content: center;
4. Через гриды и свойство justify-self: center;
5. Через margin: auto; которые делает одинаковые боковые отступы, рассчитывая из доступного пространства.
6. Fr (fraction, дробная часть) позволяет разбить сетку на пропорциональные столбцы и строки. Это гибкая единица измерения, которая рассчитывается из количество доступного пространства в контейнере.
7. .container {

display: grid;

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

}

.container {

display: grid;

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

}

.container {

display: grid;

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

}

1. Свойства auto-fill и auto-fit отвечают за автоперенос контента. auto-fill стремится поместить в строку как можно больше ячеек. При наличии свободного пространства auto-fill использует минимальную ширину колонки, а если ячейка не вмещается, то остается свободное место.  
   auto-fit стремится задать все свободное пространство и подстраивает ширину ячеек под свободное место.
2. .container {

display: grid;

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

grid-template-rows: 200px 100px;

}

1. .container {

display: grid;

grid-template-columns: 1fr 25% 1fr;

grid-template-rows: 25% 1fr 25%;

justify-content: space-between;

}

1. Grid area – пространство внутри grid-контейнера, в котором может быть помещен один или несколько grid элементов. Может состоять из одной или нескольких ячеек. Это сокращенная форма записи свойств grid-row-start, grid-column-start, grid-row-end, grid-column-end.
2. .comtainer {

display: grid;

grid-template-columns: repeat(4, 1fr);

grid-template-rows: repeat(4, 1fr);

grid-template-areas:

“article article article article”

“section1 section1 section2 . “

“aside aside section2 section2”

“aside aside section2 section2”

}

<div class=” comtainer”>

< article class=”article”> article</article>

< section class=”section1”> section1</section>

< section class=”section2”> section2>/section>

<aside class=”aside”> aside</aside>

</div>

1. auto-fit
2. Имена линиям строк и столбцов можно задать с помощью квадратных скобок при создании сетки контейнера. При этом стандартные номера линий никуда не деваются, их тоже можно использовать. Имена линий могут быть любыми, рекомендуется подбирать под описываемую область. Одной линии можно присвоить несколько имен, через пробел в одних квадратных скобках.

В том случае, когда мы создаем grid-area и даем ей название, то автоматически присваиваются названия линиям, которые ограничивают эту область.

Например,

grid-template-areas:

“article article article article”

Соответственно линиям автоматически присвоены имена article-start и article-end. И так же работает в обратном направлении: если задать имена линиям в формате article-start и article-end , то автоматически будет создана область, которая будет ограничена этими линиями.

1. grid-template-columns: repeat(12, 1fr);
2. position:fixed – элемент позиционируется относительно окна браузера, независимо от вложенности. Элемент убирается из потока страница и перестает влиять на положение окружающих элементов и размер родителя. При прокручивании страницы фиксированный элемент остается на своем месте и не прокручивается вместе со страницей.

Удобно таким образом закрепить footer или header.

1. Мне кажется, с помощью значений fit-content. Чтобы при изменении ширины области просмотра, текст не выходи за рамки своего контейнера.

Кот: Я бы сказал, что стоит использовать любые гибкие способы позиционирования, однако иногда в больших блоках текста вместо гибкости и переносов нужно сокращение текста - в этом случае часто используют троеточие, обрезая текст. В этом случае уже grid и flex не то что не помощники, а совсем не подходят, так как свойство text-overflow: ellipsis работает только с блочными элементами :disappointed: Поэтому подходит и flex, и grid, и блочная верстка