

Построение сеток на гридах по макету

Выполнен на 0%

Введение

Теория

Базовая теория

Описание сетки

Явные и неявные координаты грид-элементов

Грид-области

Именованные грид-линии

Грид-интервал

Размеры в гриде

Выравнивание в гриде

Углублённая теория

Многослойность в гриде

Повторы в гриде

Сокращённая запись

Продвинутая теория

Поведение автоматически размещаемых элементов

Определение диапазона размеров

Автоматическое заполнение грида

Дополнительный материал

Инструменты для работы с гридами

Баги и ограничения

Методика построения сеток на гридах

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, средний уровень

Кейс 8, средний уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Моё обучение / Построение сеток на гридах по макету / Теория / Базовая теория /

Размеры в гриде

Размеры строк и колонок в гриде и интервалы между ними определяются почти всеми возможными единицами измерения, относительными и абсолютными, то есть это могут быть значения в px, % и auto, а также в остальных единицах размеров CSS. Также у гридов есть своя собственная относительная единица fr.

Рассмотрим значение тех единиц, которые имеют своё особенное значение в гридах.

fr (fraction) — доля доступного (свободного от другого контента) пространства в грид-контейнере;

auto — пространство, достаточное для контента;

% — процент от доступной ширины или высоты грид-контейнера.

Если сделать 5 колонок по 1fr с одинаковым контентом, это отобразится точно так же, как если сделать их auto, и так же, как если сделать их по 20%.

```
<div class="grid">
  <div class="g1">Одинаковый контент</div>
  <div class="g2">Одинаковый контент</div>
  <div class="g3">Одинаковый контент</div>
  <div class="g4">Одинаковый контент</div>
  <div class="g5">Одинаковый контент</div>
</div>
```

```
.grid {
  display: grid;

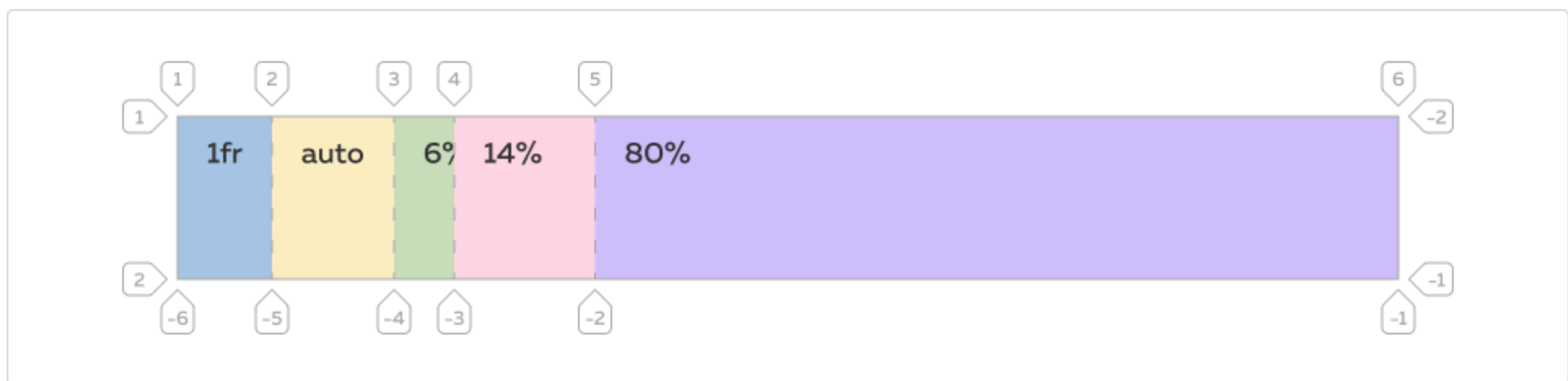
  grid-template-columns: auto auto auto auto auto;

  /* при минимальном или одинаковом контенте отобразились бы так же
  grid-template-columns: 20% 20% 20% 20% 20%;
  grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
  */
}
```



Когда контент одинаков, все возможные единицы рассчитывают размеры грид-ячеек одинаково

Фракции и auto не сжимаются меньше наименьшего размера контента, и если экран становится меньше, чем возможно, чтобы контент уместить, появляется полоса прокрутки. Значения в процентах не учитывают контент, и столбцы могут сжиматься бесконечно, полоса прокрутки не появляется.



Отображение в браузере грида с колонками разного размера

Если один из пяти столбцов с одинаковым контентом сделать 1fr, а остальные auto, то произойдёт следующее:

- Колонка с 1fr займёт минимально возможное место.
- Остальное оставшееся место поделят 4 колонки размером auto.
- Когда контент в колонках с автоматическим размером займёт свой максимум (например, текст вытянется в одну строку), всё свободное место займёт колонка размером 1fr, потому что это будет свободным местом, из которого фракции и рассчитываются.

```
.grid {
  display: grid;
  grid-template-columns: auto auto auto auto 1fr;

  /* все колонки займут минимальное необходимое для контента место, потом первые четыре колонки займут
  столько места, сколько нужно их контенту, а если что-то останется — всё займёт пятая колонка */
}
```



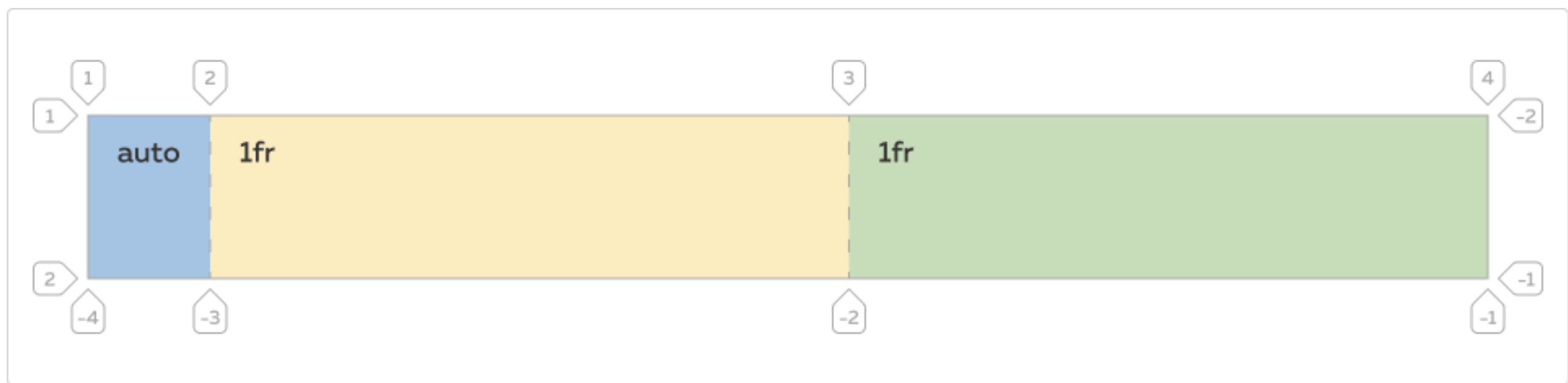
Как это отображается в браузере при минимальном контенте

Рассмотрим ещё один пример.

```
.grid {
  display: grid;
  grid-template-columns: auto 1fr 1fr;
}
```

Такая трёхколоночная сетка будет рассчитываться следующим образом:

- Сначала будет высчитан минимально возможный размер, необходимый для контента во всех трёх колонках.
- Потом колонка размером auto займёт пространство по размеру своего контента.
- Остальное пространство (если оно будет) поделят поровну колонки размером 1fr.



Как это отображается в браузере при минимальном контенте

И ещё один пример сразу со всеми тремя единицами измерения: fr, auto и %.

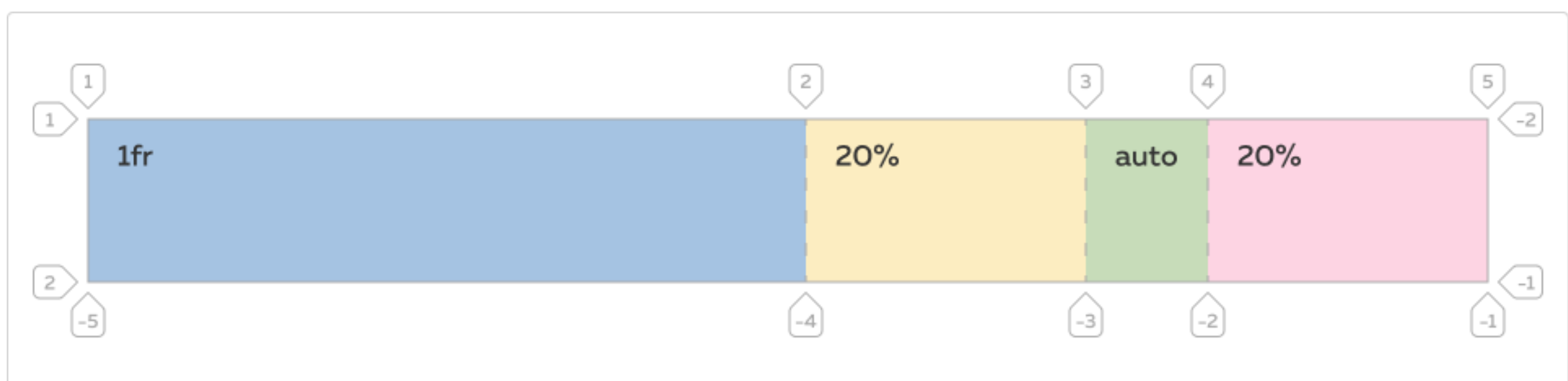
```
.grid {
  display: grid;
  grid-template-columns: 1fr 20% auto 20%;
}
```

Эта четырёхколоночная сетка будет рассчитываться так:

- Второй и четвёртый столбцы, по 20%, займут по 20% размера грид-контейнера.
- Столбцы размером auto и 1fr займут не меньше места, чем нужно для их контента в минимальном размере (например, по ширине самого длинного слова).
- Потом столбец auto займёт достаточно места для своего контента (возможно, это будет всё место).
- Если что-то останется — это достанется столбцу размером 1fr.

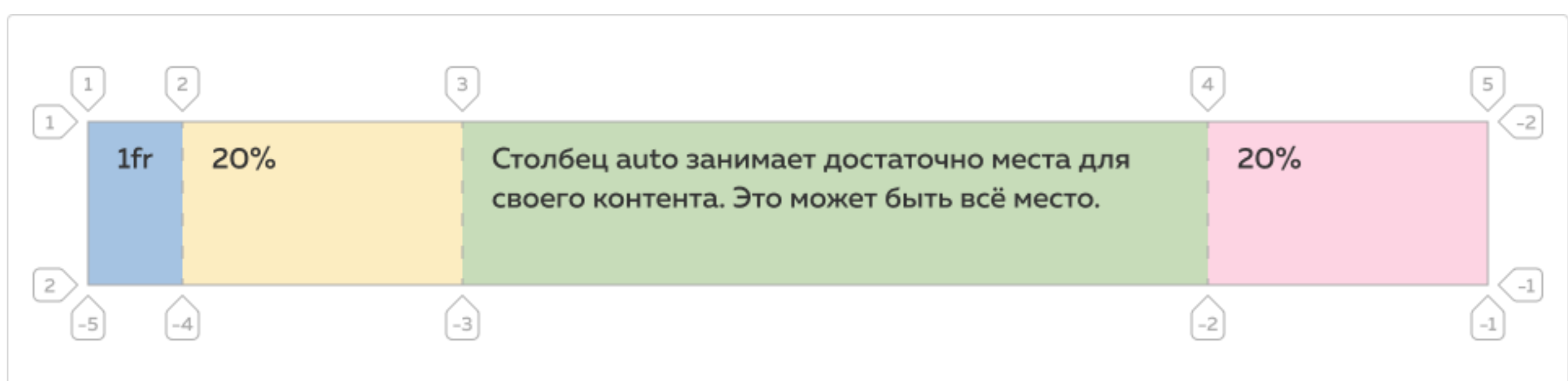
Рассмотрим два варианта содержимого столбца, ширина которого равна значению auto.

Небольшое количество текста в содержимом:



Грид с четырьмя колонками, две по 20%, одна авто и одна фракция

Длинный текст в содержимом колонки:



Грид с четырьмя колонками, две по 20%, одна авто и одна фракция

В фракциях (fr) можно задавать только размеры столбцов и строк в гриде. Размеры грид-интервалов (gap) во фракциях не указываются (даже если вы это сделаете, это не работает). Теоретически можно задавать значения грид-интервалов в процентах (проценты в данной ситуации считаются от размера контейнера или доступного пространства), но спецификация не фокусируется на этой возможности, и в разных браузерах проценты рассчитываются по-разному.

Ознакомьтесь со статьёй?

Сохранить прогресс

Грид-интервал

Выравнивание в гриде

Практикум

Тренажёры  
Подписки  
Для команд и компаний  
Учебник по PHP

Профессии

Фронтенд-разработчик  
React-разработчик  
Фулстек-разработчик  
Бэкенд-разработчик

Услуги

Работа наставником  
Для учителей  
Стать автором

Курсы

HTML и CSS. Профессиональная вёрстка сайтов  
HTML и CSS. Адаптивная вёрстка и автоматизация  
JavaScript. Профессиональная разработка веб-интерфейсов  
JavaScript. Архитектура клиентских приложений  
React. Разработка сложных клиентских приложений  
PHP. Профессиональная веб-разработка  
PHP и Yii. Архитектура сложных веб-сервисов  
Node.js. Разработка серверов приложений и API  
Анимация для фронтендеров  
Вёрстка email-рассылок  
Vue.js для опытных разработчиков  
Регулярные выражения для фронтендеров  
Шаблонизаторы HTML  
Алгоритмы и структуры данных  
Анатомия CSS-каскада

Блог

С чего начать  
Шпаргалки для разработчиков  
Отчеты о курсах

Информация

Об Академии  
О центре карьеры

Остальное

Написать нам  
Мероприятия  
Форум