

## Построение сеток на флексах по макету

Выполнен на 0%

- Введение
- Углублённая теория
- Методики работы с флексами

Методика построения сеток

Многострочный флексбокс

Вложенные флексы

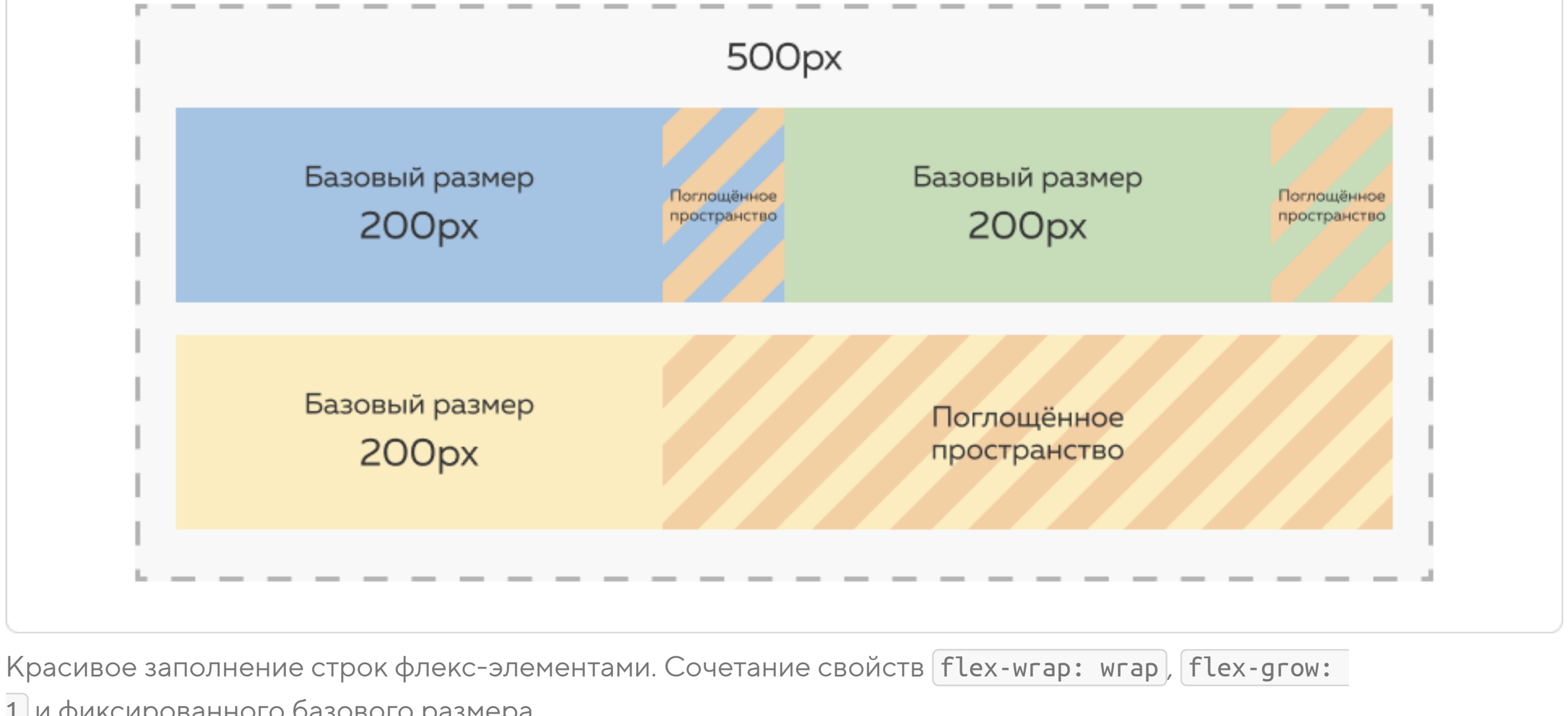
Свойство flex-grow и «адаптивность от контейнера»

Применяем флексы для «слоанных» и нестандартных сеток
- 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, сложный уровень
- Кейс 9, сложный уровень
- Кейс 10, сложный уровень

Главная / Мое обучение / Построение сеток на флексах по макету / Методики работы с флексами /

# Свойство flex-grow и «адаптивность от контейнера»

Понимо всего прочего, флекс умеет делать красивое заполнение строк флекс-элементами так, чтобы в каждой строке край флекс-элементов вплотную прилегли к краям контейнера. И работает этот механизм как для фиксированных так и для резиновых контейнеров. Такое поведение возможно благодаря сочетанию `flex-wrap: wrap` и `flex-grow: 1` (подробнее о свойстве читайте в материале «Многострочный флексбокс») и положительного значения `flex-grow`.



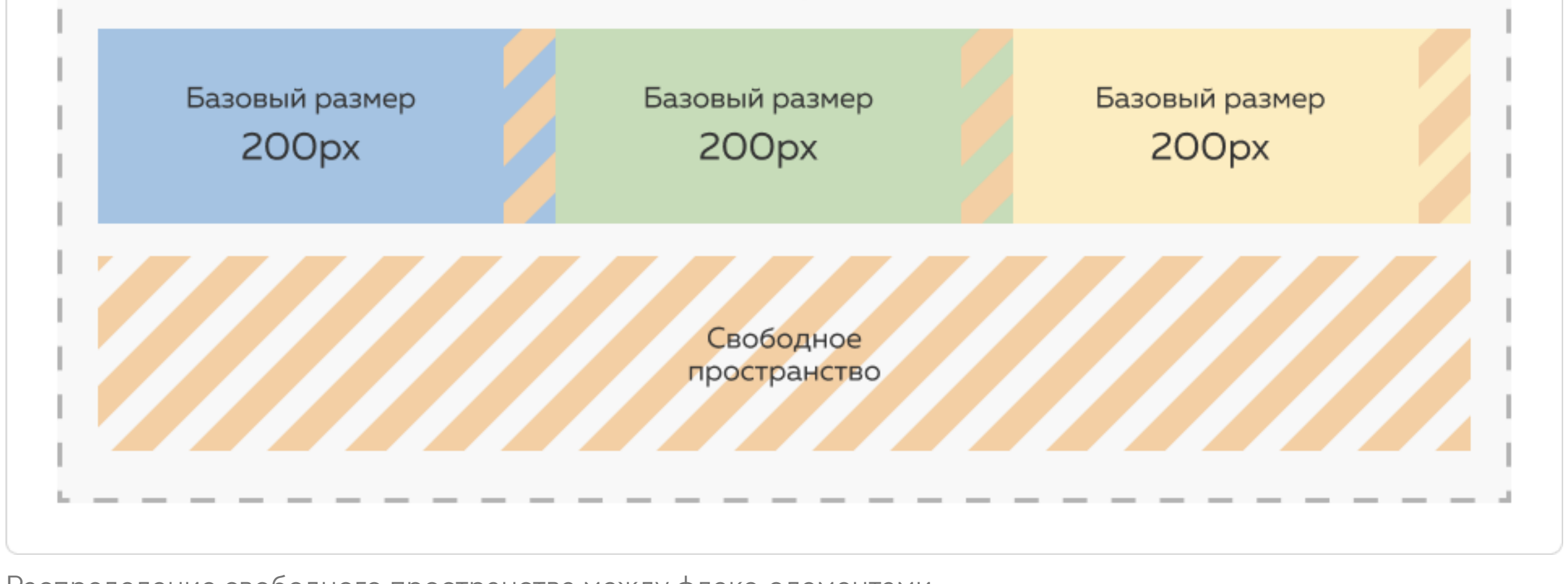
Красивое заполнение строк флекс-элементами. Сочетание свойств `flex-wrap: wrap`, `flex-grow: 1` и фиксированного базового размера

Алгоритм заполнения строк состоит из следующих шагов:

- по исходным базовым размерам флекс-элементов браузер определяет, сколько будет флекс-элементов в каждой флекс-строке;
- после распределения флекс-элементов по флекс-строкам для элементов с положительным значением `flex-grow` включаются гибкие размеры и такие элементы «сдвигают» свободное пространство в своей строке пропорционально значению свойства `flex-grow`.

Принем этот алгоритм работает как для однострочных, так и для многострочных флекс-контейнеров.

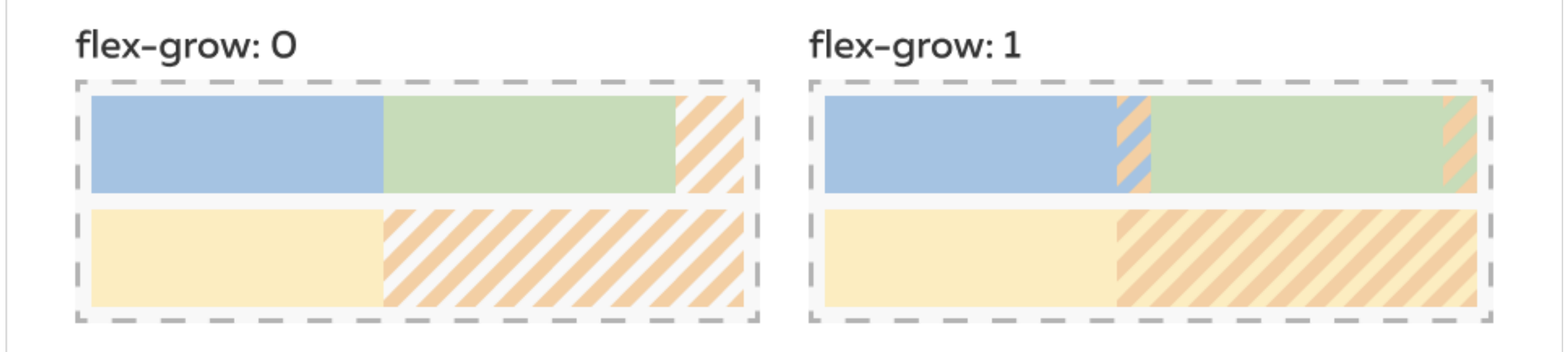
Разберём всё на схемах. И нанём с простого примера, все элементы имеют значение `flex-grow: 1`.



Распределение свободного пространства между флекс-элементами

Как видите, все элементы получили равное дополнение к базовому размеру и заполнили всю ширину флекс-контейнера.

В следующем примере мы имеем дело с многострочным флекс-контейнером, мы разрешили перенос флекс-элементов на следующую строку `flex-wrap: wrap`. В первой строке находятся два элемента, третий элемент переходит на вторую строку. Установим для всех элементов значение свойства `flex-grow: 1`. После этого элементы первой строки делают свободное пространство только первой строки, несильно увеличиваясь в размерах. А элементу во второй строке достаётся больше свободного пространства, поскольку он в строке всего один.



Порядок применения `flex-grow` при `flex-wrap: wrap`

Иногда сочетание `flex-wrap: wrap` с положительным `flex-grow` называют «адаптивностью от контейнера». Ведь при изменении ширины контейнера сетка может достаточно сильно перестраиваться. Но это не полноценная адаптивность, а просто один из возможных приёмов перестроения сетки в ситуации, когда не важен точный контроль размеров и пропорций.



«Адаптивность от контейнера»

Рассмотрим как можно использовать этот механизм при построении сеток для разделов сайта.

Задача построить сетку для раздела с галереей фотографий. Исходные файлы фотографий имеют два типа размеров: 570x320 и 285x320. По макету необходимо, чтобы между фотографиями не было отступов, а также чтобы они занимали всю ширину контейнера. Контейнер имеет фиксированную ширину `1200px`.

Разметки и стили для раздела могут выглядеть так:

```
<div class="container">
  <ul class="gallery-list">
    <li class="gallery-item gallery-item-big">
      
    </li>
    <li class="gallery-item">
      
    </li>
    <li class="gallery-item">
      
    </li>
    <li class="gallery-item">
      
    </li>
    <li class="gallery-item">
      
    </li>
    <li class="gallery-item gallery-item-big">
      
    </li>
  </ul>
</div>
```

```
/* стили для контейнера с фотографиями */
.container {
  width: 1200px;
}

/* стили для списка фотографий */
.gallery-list {
  padding: 0;
  margin: 0;
  list-style: none;

  display: flex;
  flex-wrap: wrap;
}

/* стили для отдельного элемента списка */
.gallery-item {
  display: flex;
}
```

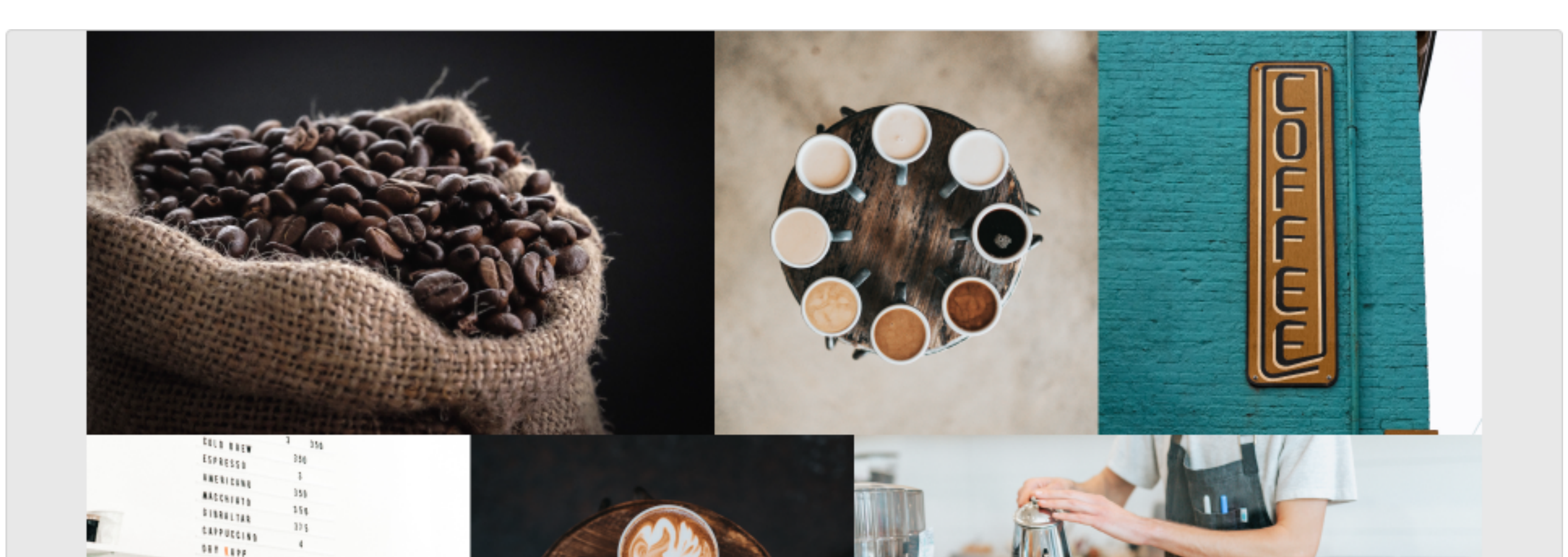


Галерея с исходным размером изображений

Как видите, справа у контейнера с фотографиями осталось свободное место.

Попробуем заполнить свободное пространство используя `flex-grow: 1`. Добавим это свойство для флекс-элементов:

```
.gallery-item {
  display: flex;
  flex-grow: 1;
}
```



Сетка для галереи с `flex-grow: 1`

Замечание

Чтобы изображения увеличивалось только по ширине и при этом сохраняло свои пропорции при заполнении блока можно использовать следующие стили:

```
.gallery-item-img {
  width: 100%;
  object-fit: cover;
}
```

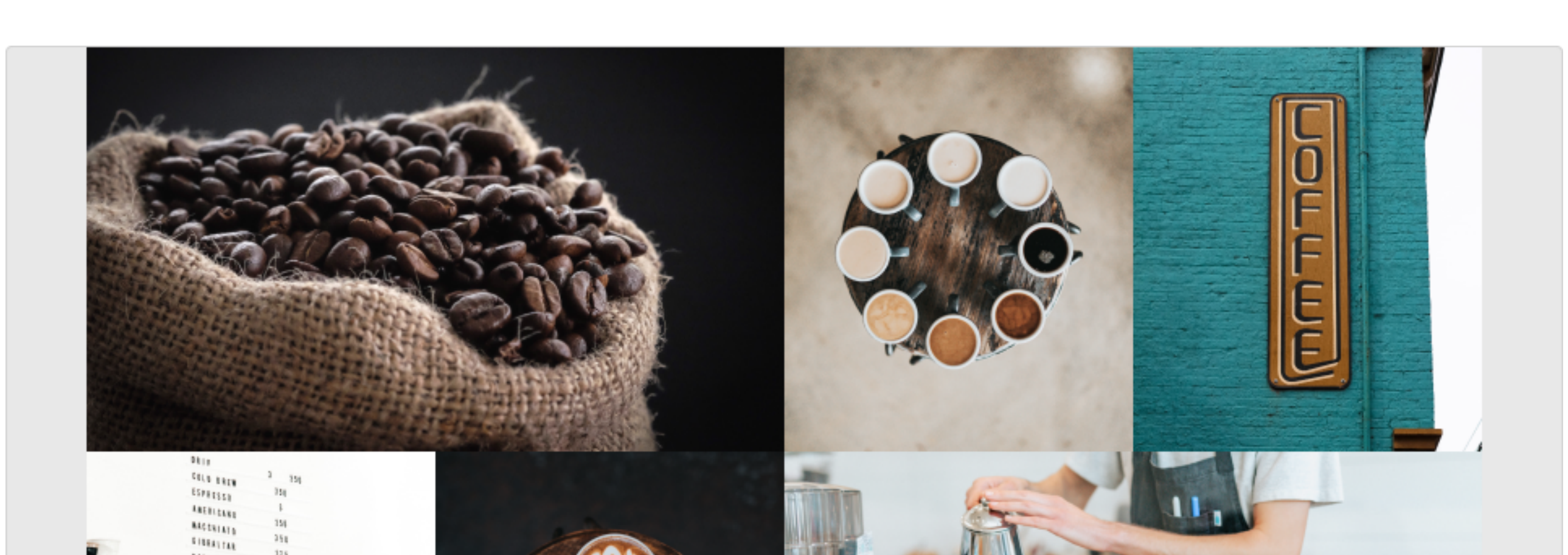
Блоки растянулись на всю ширину контейнера. Но вы, наверное, обратили внимание, что сетка немного сломалась. Нет чётких колонок, как в исходном состоянии. Это произошло из-за того, что изначальные размеры элементов отличаются. В нашем случае ширина первого типа элементов в два раза больше, чем ширина второго типа элементов. Поэтому лучше свободное пространство поделить в эти же пропорциях. То есть для элементов меньшего размера задать `flex-grow: 1`, а для большего – `flex-grow: 2`.

```
...

/* стили для элементов меньшего размера */
.gallery-item {
  display: flex;
  flex-grow: 1;
}

/* стили для элементов большего размера */
.gallery-item-big {
  flex-grow: 2;
}
```

В результате получим:



Сетка для галереи с разными значениями свойства `flex-grow`

Мы растянули элементы галереи на всю ширину контейнера и сохранили колонки для сетки.

Итоговые CSS-правила для нашей галереи могут выглядеть так:

```
/* стили для контейнера с фотографиями */
.container {
  width: 1200px;
}

/* стили для списка фотографий */
.gallery-list {
  padding: 0;
  margin: 0;
  list-style: none;

  display: flex;
  flex-wrap: wrap;
}

/* стили для элементов меньшего типа размеров */
.gallery-item {
  display: flex;
  flex-grow: 1;
}

/* стили для элементов большего типа размеров */
.gallery-item-big {
  flex-grow: 2;
}
```

Ознакомились со статьёй?

Сохранить прогресс

Вложенные флексы

Применяем флексы для «слоанных» и нестандартных сеток