

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 «сломанных» 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 /

Многострочный флексбокс

По умолчанию внутри флекс-контейнера перенос флекс-элементов на новую строку запрещён. Это хорошо тем, что раскладки на флексах очень «крепкие». Ситуаций, когда сетка разваливается, а колонки съезжают друг под друга, во флексе быть не может, в отличие от флоатов и блочно-строчных. Если флекс-элементы не помещаются в одну строку флекс-контейнера, то сначала браузер попытается их сжать с помощью `flex-shrink`. Если и после сжатия места будет не хватать, то элементы начнут выпадать за границу контейнера или обрезаться (всё зависит от того, какой режим переполнения установлен для контейнера).

```
.flex-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-wrap: nowrap; /* по умолчанию */
}

.flex-container > div {
  width: 200px;
  height: 50px;
  margin: 0 10px;
  padding: 10px;
}
```

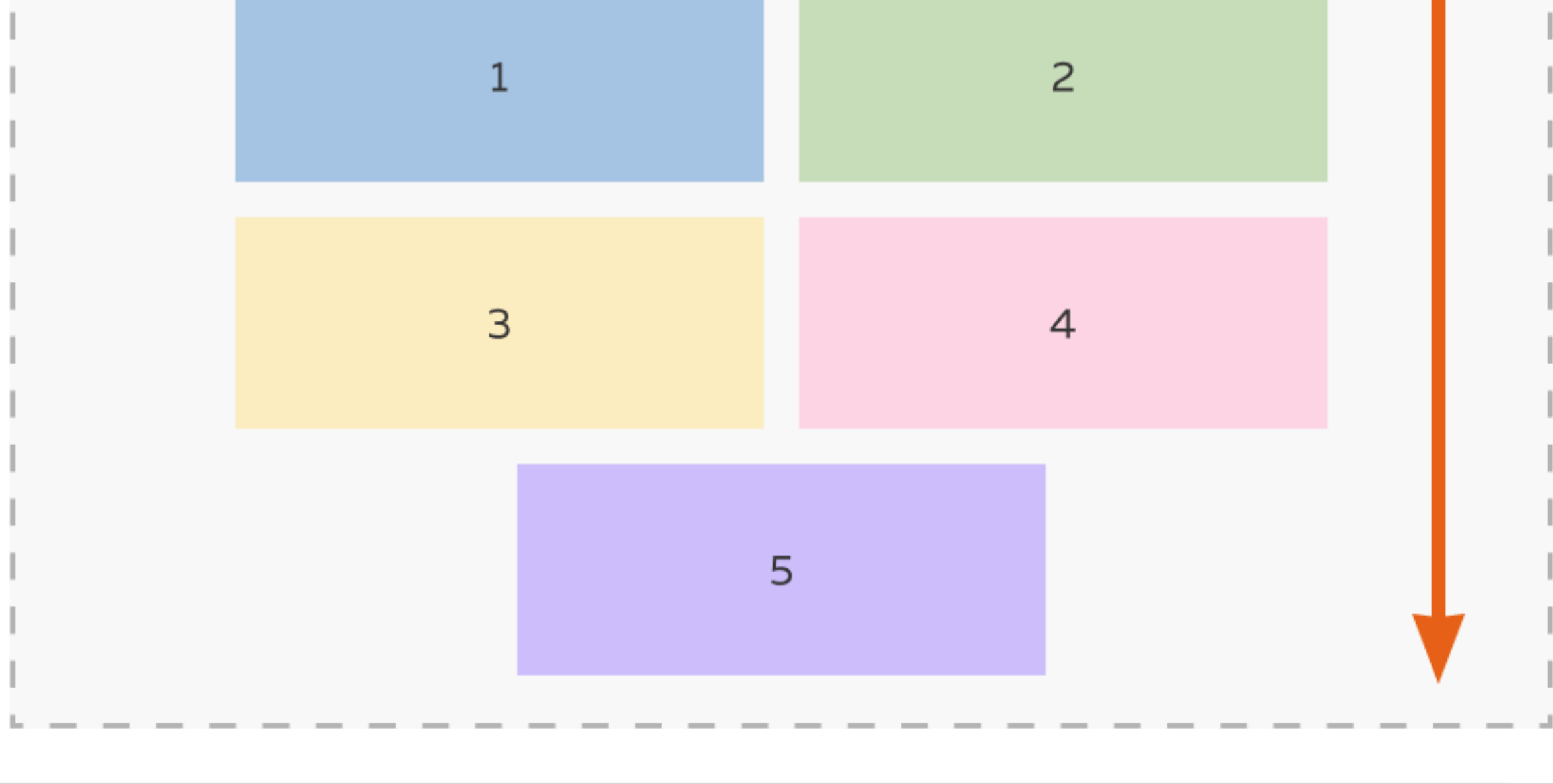


Флекс-элементы на одной строке — значение по умолчанию `flex-wrap: nowrap`

Конечно, разработчики флексбокса предусмотрели и ситуацию, когда нужно переносить флекс-элементы на новые строки. За это отвечает свойство `flex-wrap`. Его значение по умолчанию, `nowrap`, запрещает перенос флекс-элементов на новую строку. Другие значения — `wrap` и `wrap-reverse`, разрешают перенос флекс-элементов на новую строку. Если заданы значения `wrap` или `wrap-reverse`, флекс-контейнер превращается из однострочного в многострочный. В нём может появиться виртуальная сущность, которая называется *флекс-строка*. К сожалению у нас нет инструментов для управления флекс-строками. Мы не можем сказать: «А сделай так, чтобы элементы во второй строке были красные». Мы даже не можем понять, что в итоге попадёт во вторую флекс-строку.

```
.flex-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}

.flex-container > div {
  width: 200px;
  height: 50px;
  margin: 0 10px;
  padding: 10px;
}
```

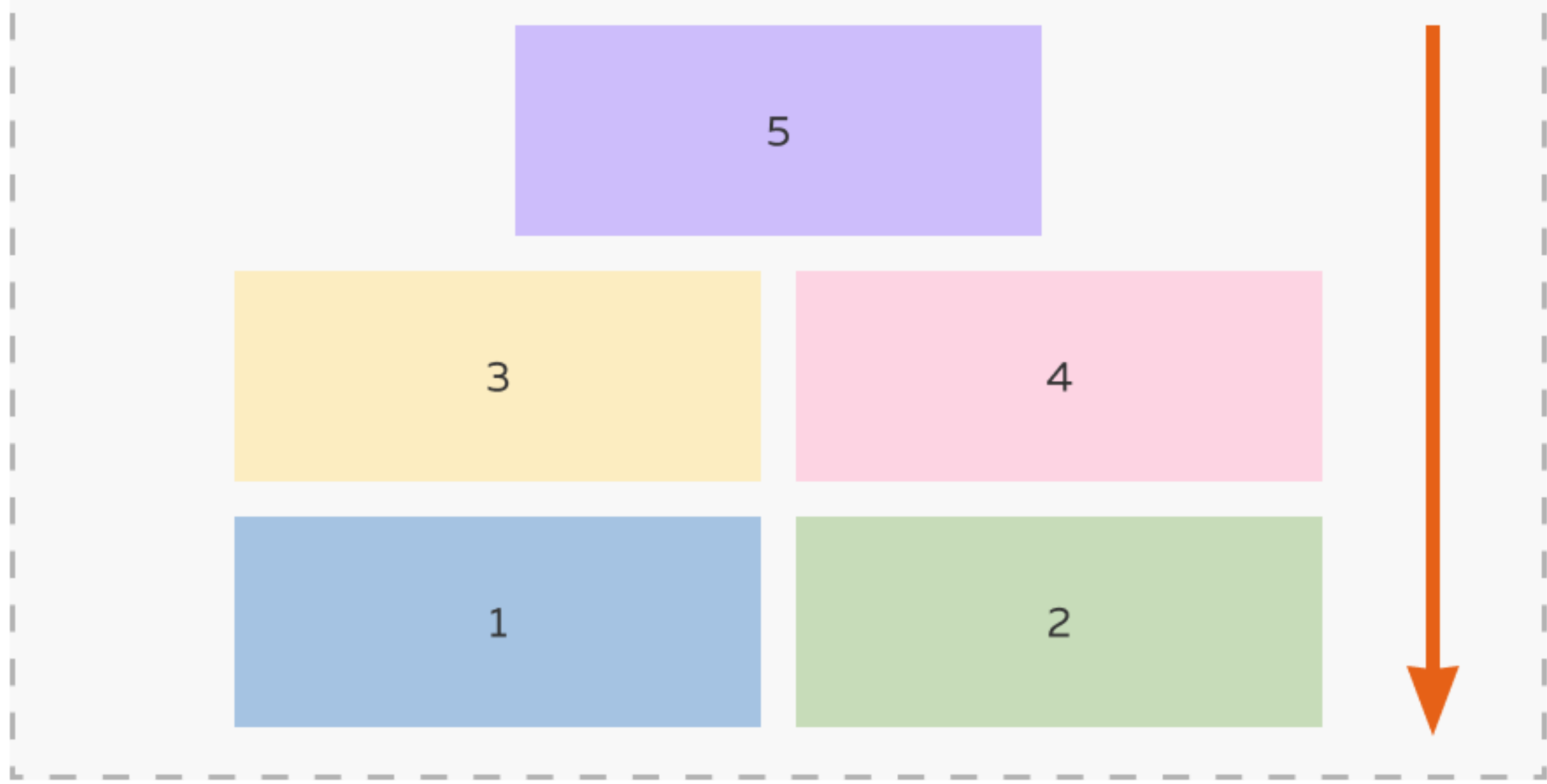


Перенос флекс-элементов на другую строку, значение `flex-wrap: wrap`

Если в свойстве `flex-wrap` задано значение `wrap`, флекс-строки выстраиваются от начала к концу поперечной оси. Если задано значение `wrap-reverse`, то порядок флекс-строк меняется — от конца поперечной оси к началу.

```
.flex-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap-reverse;
}

.flex-container > div {
  width: 200px;
  height: 50px;
  margin: 0 10px;
  padding: 10px;
}
```

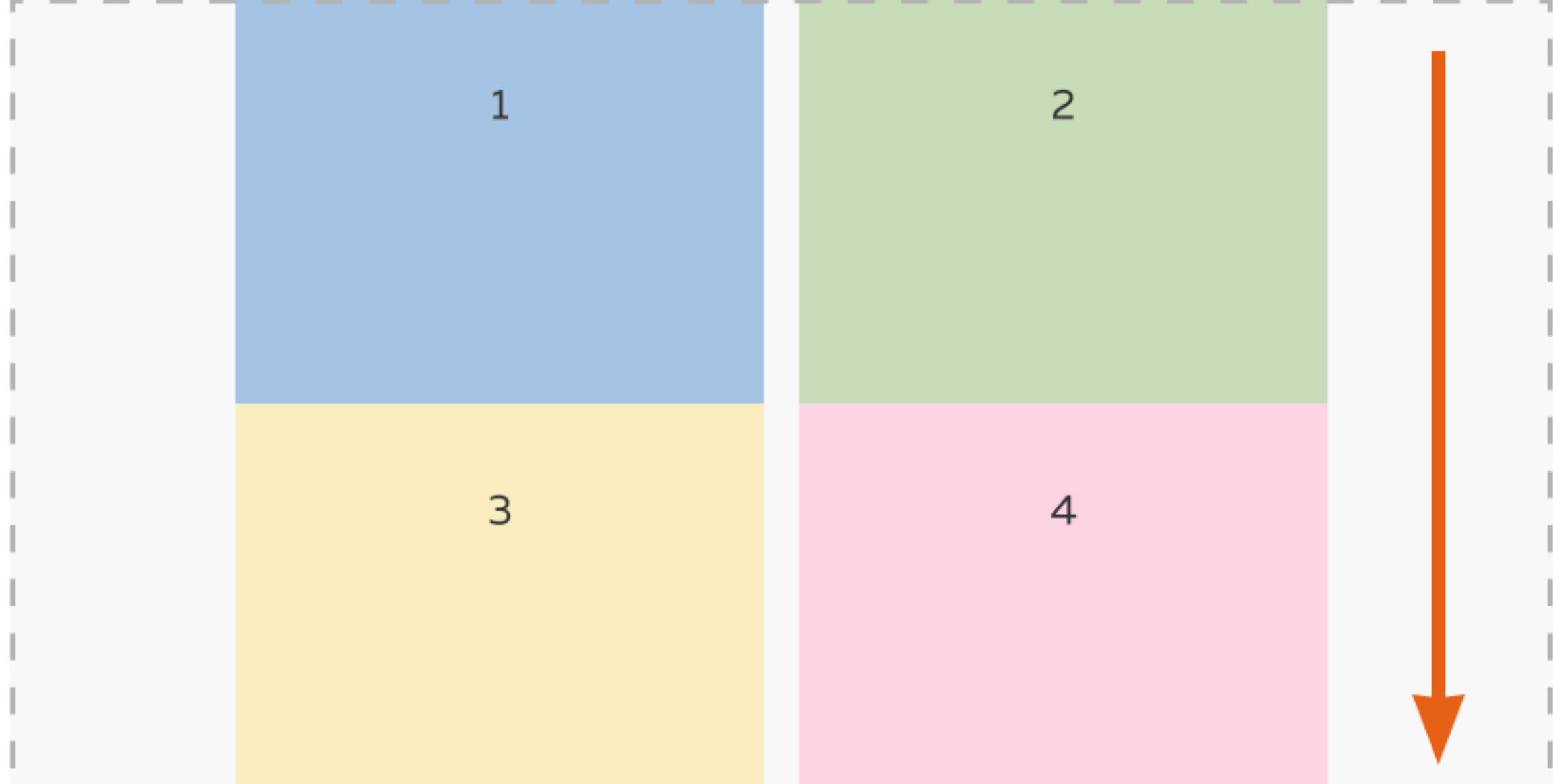


Изменяет порядок флекс-строк значение `wrap-reverse` свойства `flex-wrap`

Флексбокс позволяет управлять не только распределением флекс-элементов во флекс-строке, но и распределением флекс-строк. За это отвечает свойство `align-content`. Значения этого свойства чем-то схожи с `justify-content`, потому что у флекс-строк есть одна общая ось — поперечная ось флекс-контейнера. Свойство `align-content` используется достаточно редко. Дело в том, что обычно флекс-контейнерам не задают фиксированную высоту, а верстают так, чтобы высота флекс-контейнера зависела от его содержимого, то есть от количества флекс-строк внутри. В этом случае внутри флекс-контейнера, вдоль его поперечной оси, не остаётся свободного пространства, внутри которого можно было бы распределять флекс-строки.

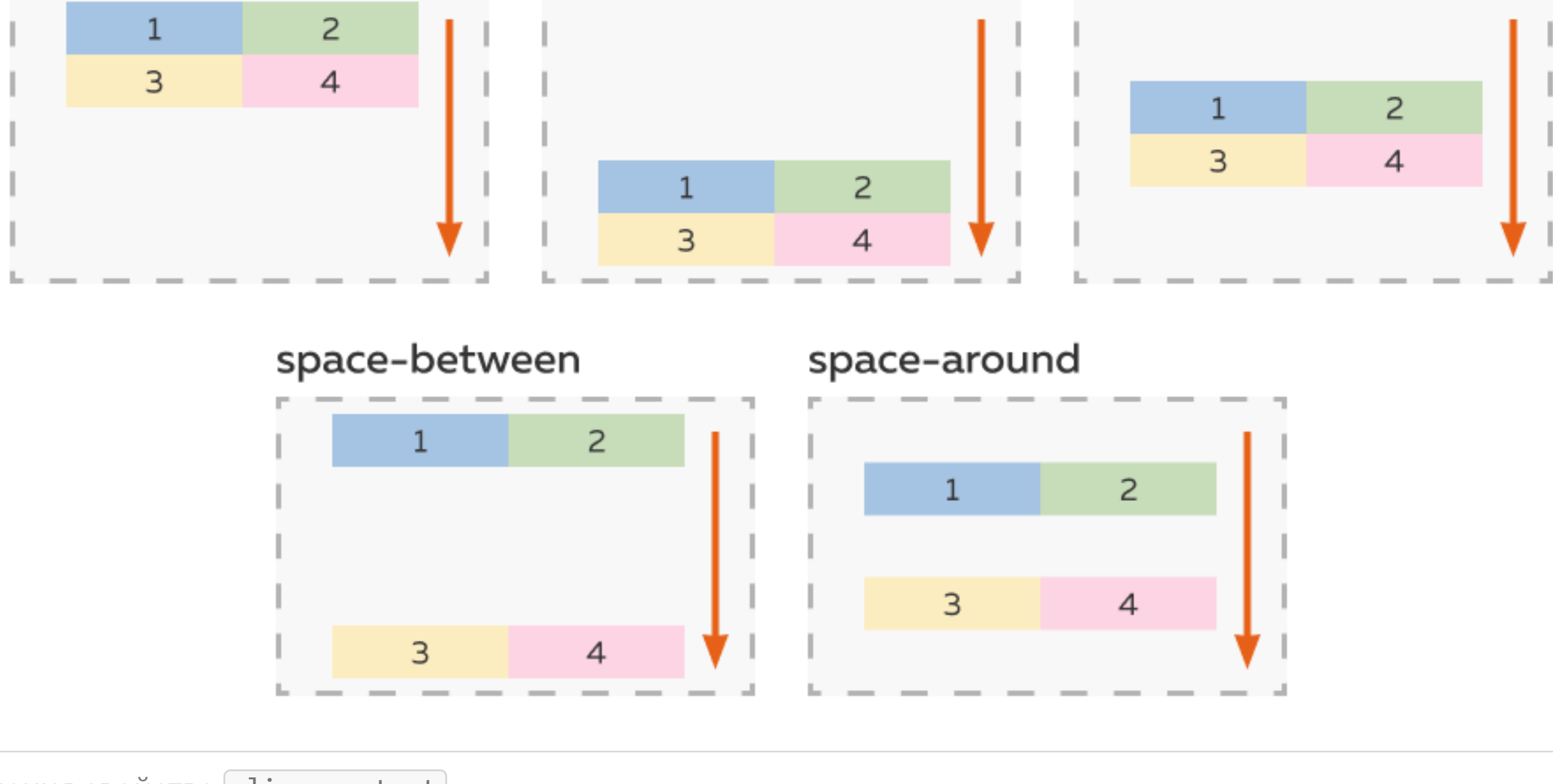
```
.flex-container {
  display: flex;
  justify-content: center;
  align-items: stretch;
  flex-wrap: wrap;
  align-content: stretch; /* по умолчанию */

  /*
  align-content: flex-start;
  align-content: flex-end;
  align-content: center;
  align-content: space-between;
  align-content: space-around;
  */
}
```



Распределение флекс-элементов во флекс-строке по умолчанию, свойство `align-content: stretch`

Как именно браузер рассчитывает, сколько флекс-элементов будет на каждой флекс-строке? Для этого он использует те самые исходные базовые размеры флекс-элементов, о которых мы говорили в материале об алгоритме раскладки.



Значения свойства `align-content`

Приведём пример:

Ширина флекс-контейнера — `500px`. Внутри него есть три флекс-элемента по `200px` каждый. В этом случае на первую строку флекс-контейнера помещаются два флекс-элемента (два элемента займут `400px`), а третий уже не поместится (три элемента займут `600px`). Поэтому внутри флекс-контейнера сформируется две флекс-строки. В первой будет два флекс-элемента, а во второй — один флекс-элемент.



Сочетание фиксированного базового размера контейнера и значения `wrap` свойства `flex-wrap`

Если контейнер увеличить до `650px`. Тогда третий флекс-элемент начнёт помещаться на первую флекс-строку и второй флекс-строка исчезнет.



Поведение флекс-элементов при увеличении базового размера флекс-контейнера

Описанная в примере механика является стандартной. Флоаты и блочно-строчные боксы работали так же: пока элементы помещаются на одну строку, они располагаются на одной строке, как только перестают помещаться — элемент из конца строки переносится на следующую строку.

Вот только флекс умеет ещё кое-что. Он умеет делать красивое заполнение строк флекс-элементами так, чтобы в каждой строке края флекс-элементов вплотную прилегли к краям контейнера. Подробнее как этого можно добиться читайте в материале [Свойство flex-grow и «адаптивность от контейнера»](#).

Ознакомились со статьёй?

Сохранить прогресс

Методика построения сеток

Вложенные флексы