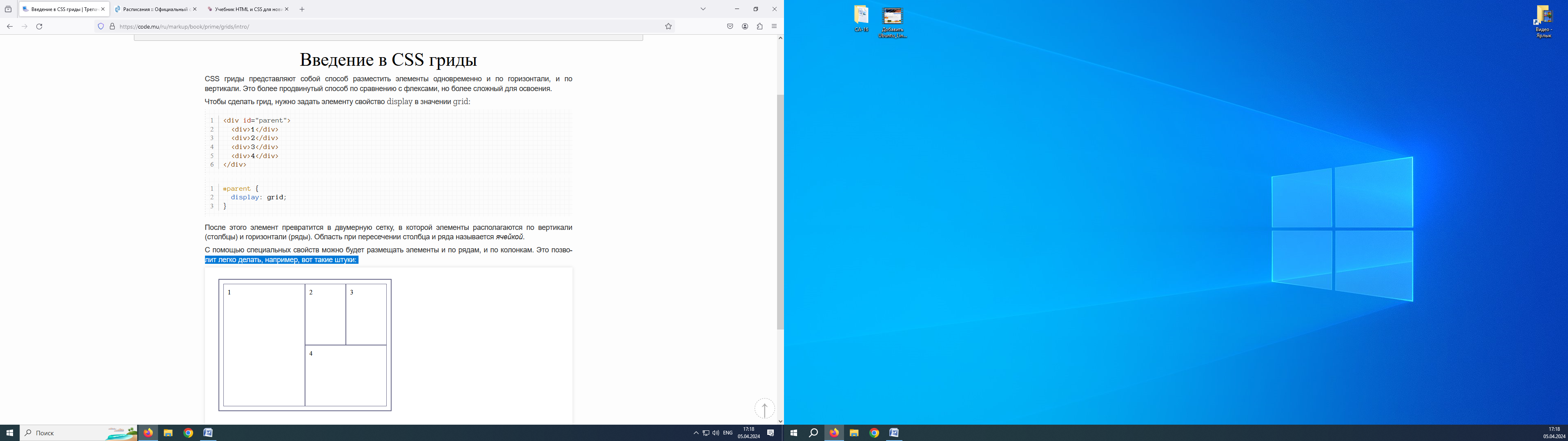
**Теория к практическому занятию**

**Введение в CSS Grid**

CSS Grid представляют собой способ размещения элементов одновременно и по горизонтали, и по вертикали. Это более продвинутый способ по сравнению с флексами, но более сложный для освоения.

Чтобы сделать грид, нужно задать элементу свойство display в значении grid:



После этого элемент превратится в двумерную сетку, в которой элементы располагаются по вертикали (столбцы) и горизонтали (ряды). Область при пересечении столбца и ряда называется ***ячейкой***.

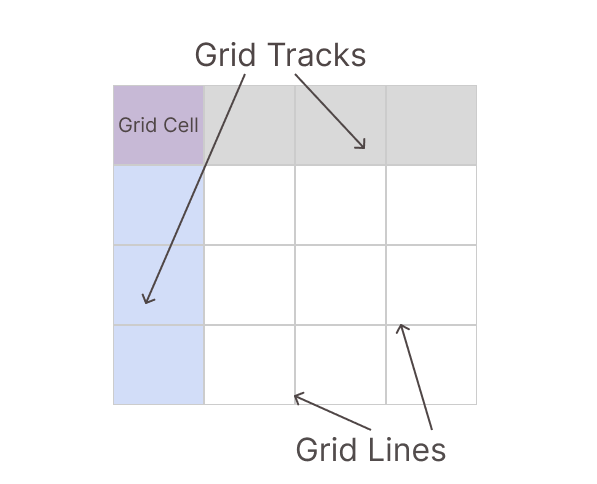
**Треки и линии в CSS гридах**

По своей сути сетка грида состоит из треков и линий.

Под ***треком*** *(gridtracks)* понимают столбец (вертикальная строка) или ряд (горизонтальная строка) грида.

***Ячейка*** *(gridcell)* - это область пересечения столбца и ряда.

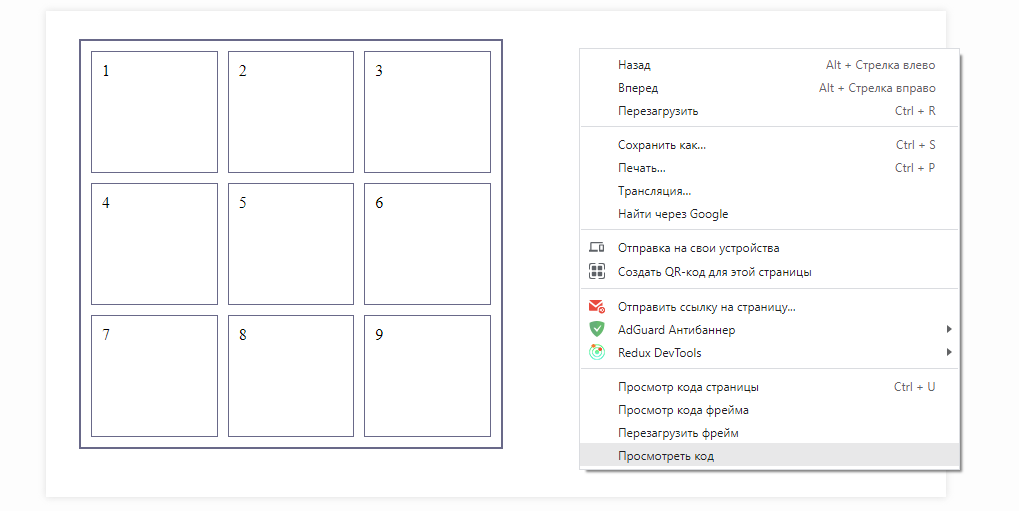
***Линии*** *(gridlines)* размещаются между треками в гриде. Смотрите на рисунок ниже:



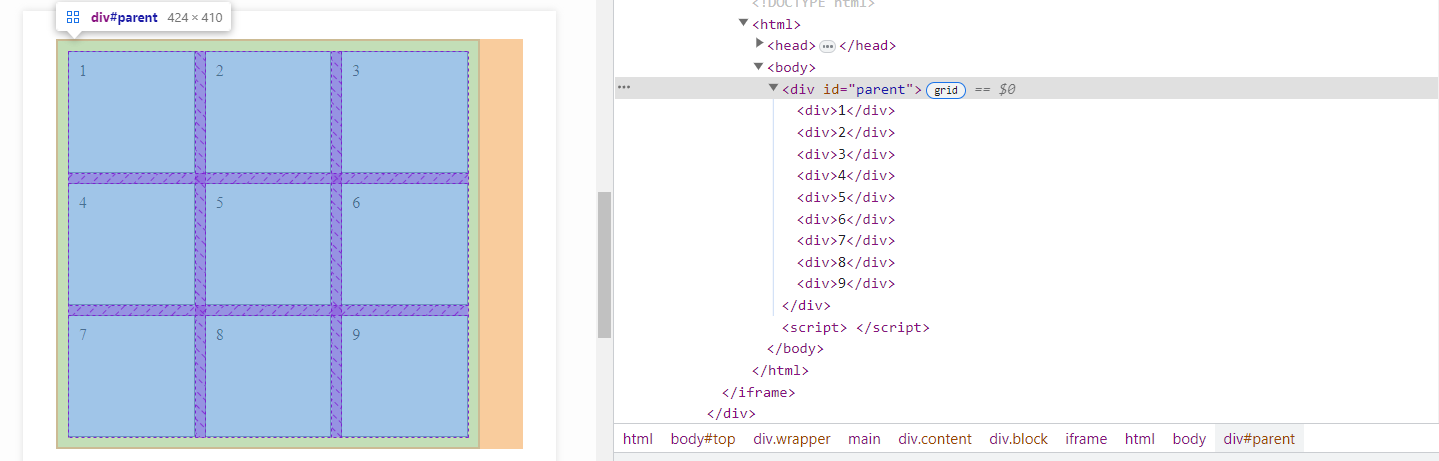
**Отображение гридов в отладчике браузера**

При работе с гридами в процессе верстки сайта бывает нужно проверить насколько правильно происходит расстановка элементов в сетке.

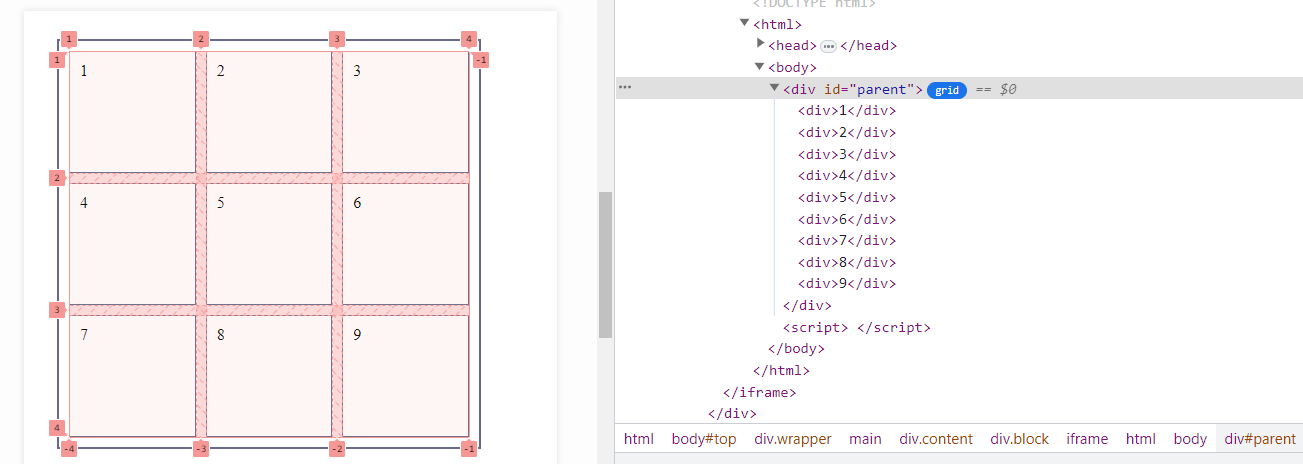
Рассмотрим грид на примере отладчика в браузере Chrome. Вначале выбираем интересующий нас объект с грид и кликнув правой кнопкой мыши выбираем из списка опцию 'Просмотреть код':



Теперь в открывшейся справа панели браузера наводим курсор на наш объект с гридом и замечаем, что в строке рядом с родительским элементом находится кнопка grid:



После того как нажали на кнопку grid, мы увидим, что наш объект отображается в визуально понятной системе грид, с указанием нумерации всех столбцов и рядов:



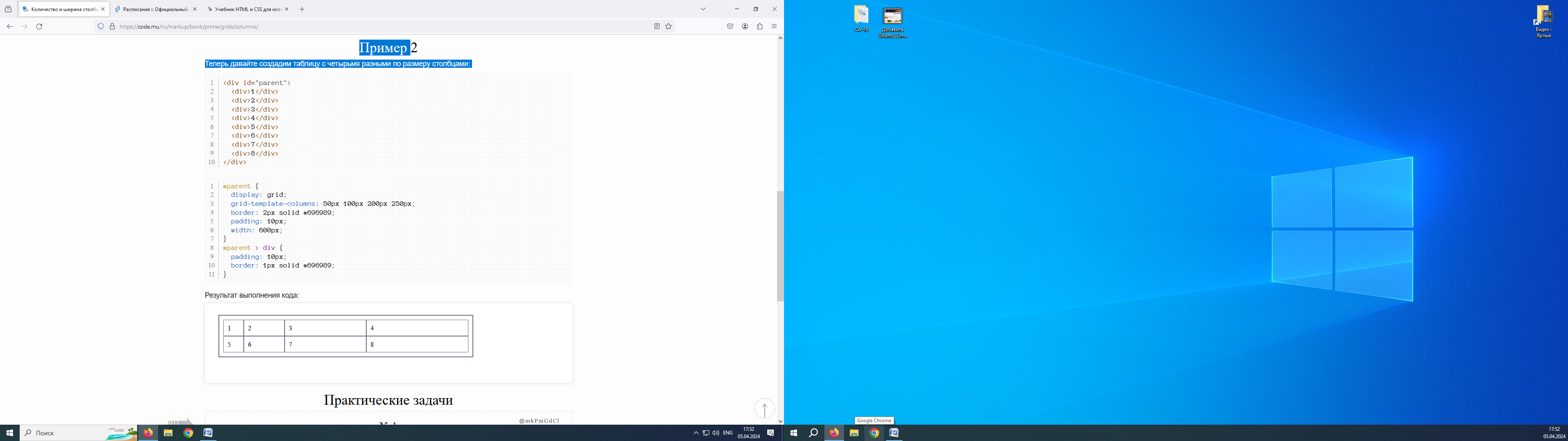
# Количество и ширина столбцов в CSS гридах

Знакомство с работой в гриде мы начнем с того, что назначим количество и ширину столбцов, по которым будут размещаться дочерние элементы. Для этой цели мы используем свойство ***grid-template-columns***, которое указывается в элементе-родителе и задает количество и ширину столбцов, которые будут занимать элементы-потомки в гриде. В значении свойства указываем ширину столбцов в пикселях.

## Пример

Создайте родительский элемент и сделайте его грид-контейнером. Пусть у нас есть div, в котором располагается четыре дочерних элемента. Зададим для него в свойстве display значение grid, а в свойстве ***grid-template-columns*** пропишем ширину для столбцов:

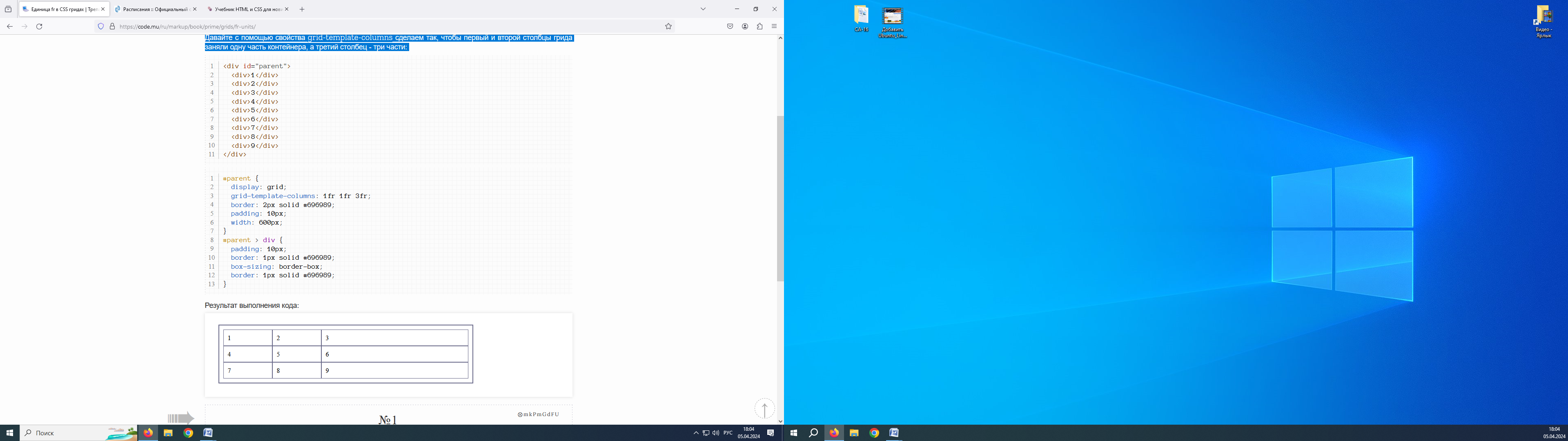
Создайте таблицу с четырьмя разными по размеру столбцами:



# Единица fr в CSS гридах

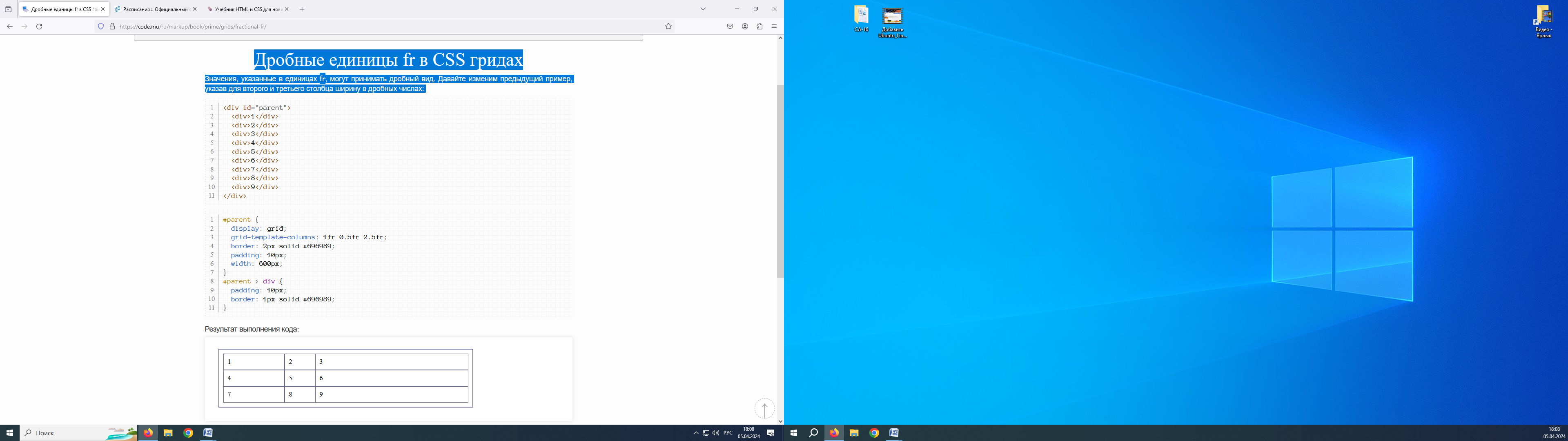
Размеры рядов и столбцов в гриде можно задавать не только в **px**, но и в единицах гибкости **fr** (фракция). Использование этих единиц означает, что все пространство для размещения элементов будет разбито на одинаковые доли или фракции. Каждый элемент сможет взять определенную часть этого разбиения.

Давайте с помощью свойства ***grid-template-columns*** сделаем так, чтобы первый и второй столбцы грида заняли одну часть контейнера, а третий столбец - три части:



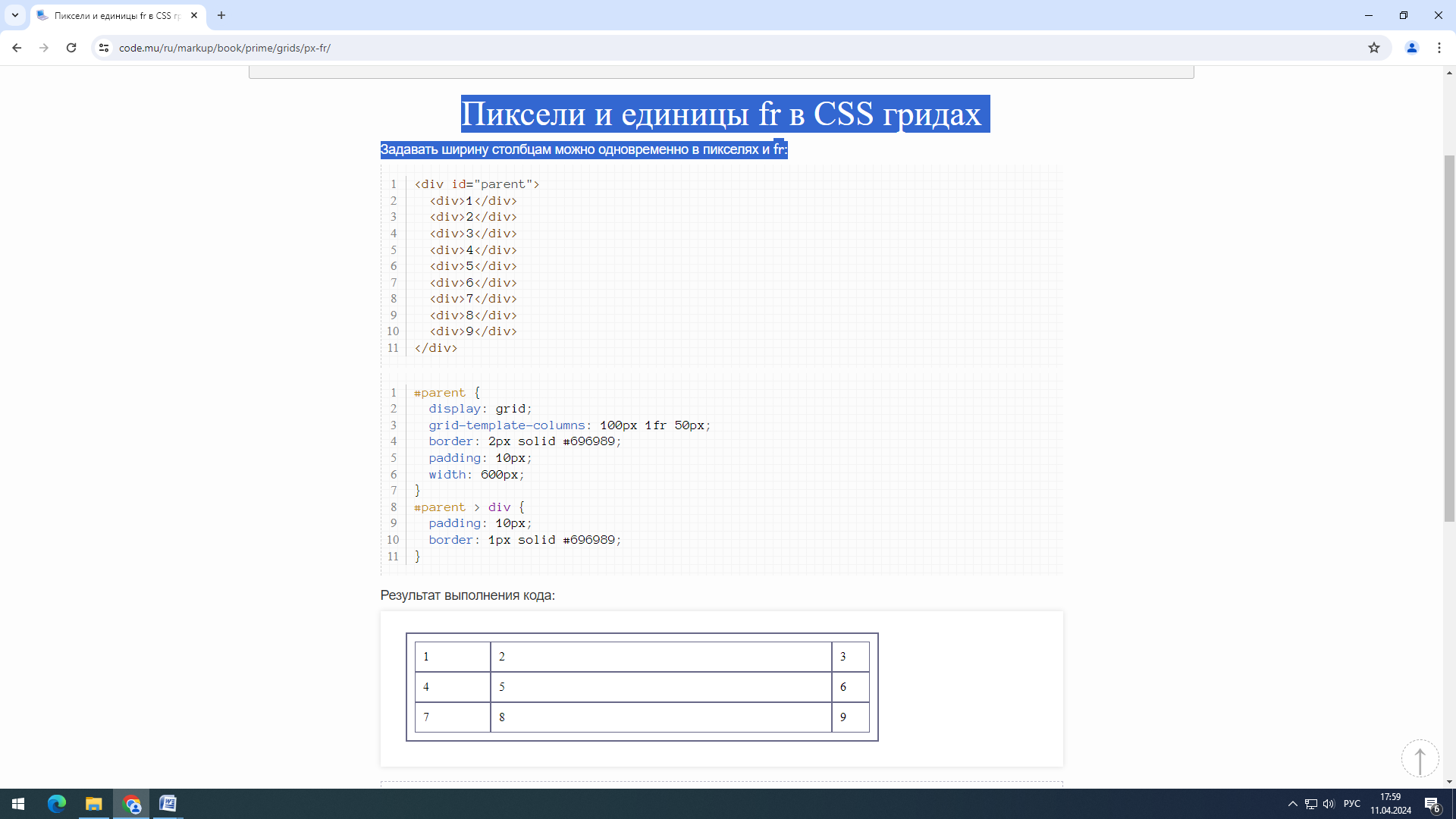
# Дробные единицы fr в CSS гридах

Значения, указанные в единицах fr, могут принимать дробный вид. Давайте изменим предыдущий пример, указав для второго и третьего столбца ширину в дробных числах:



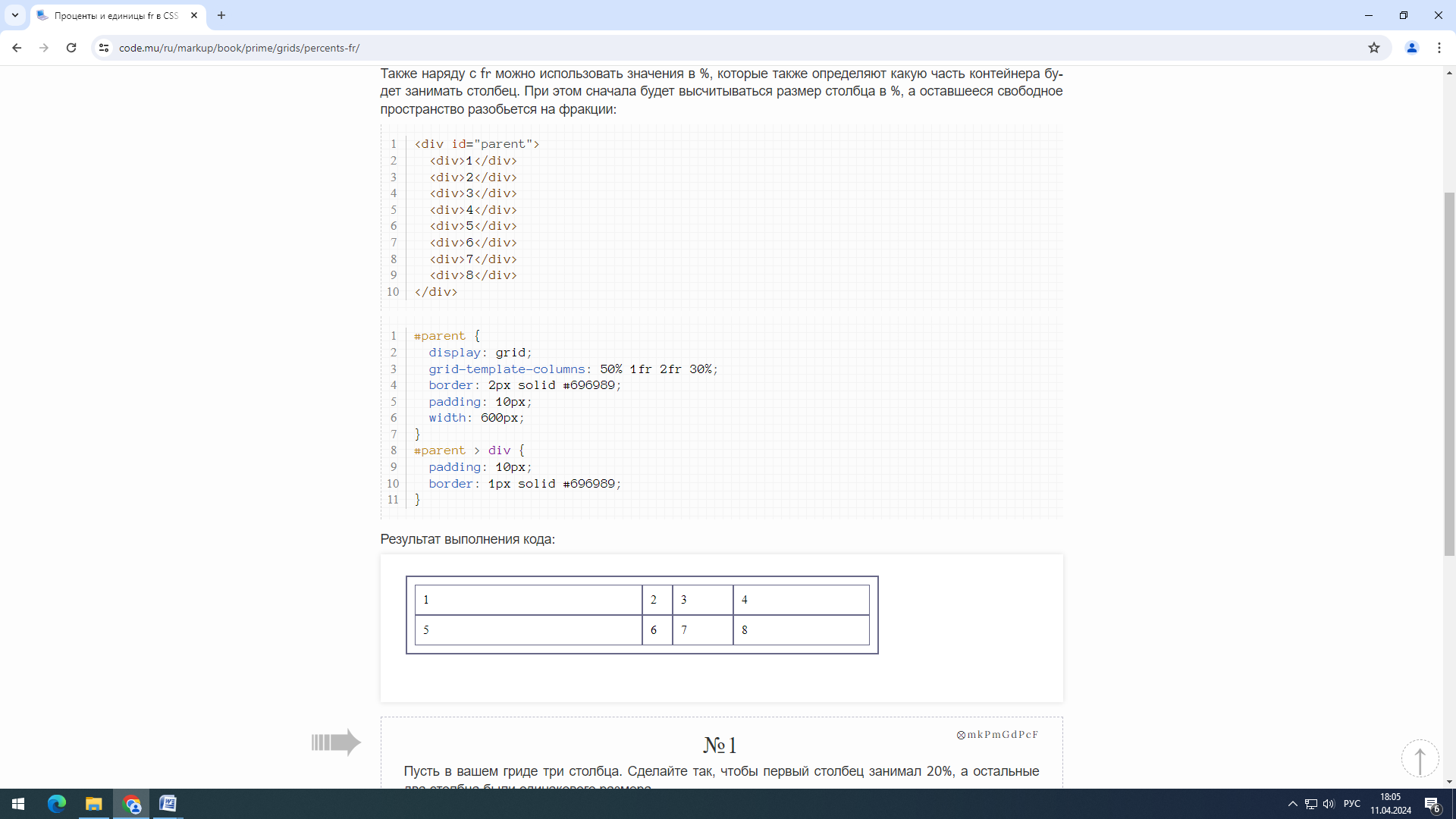
**Пиксели и единицы fr в CSS гридах**

Задавать ширину столбцам можно одновременно в пикселях и fr:



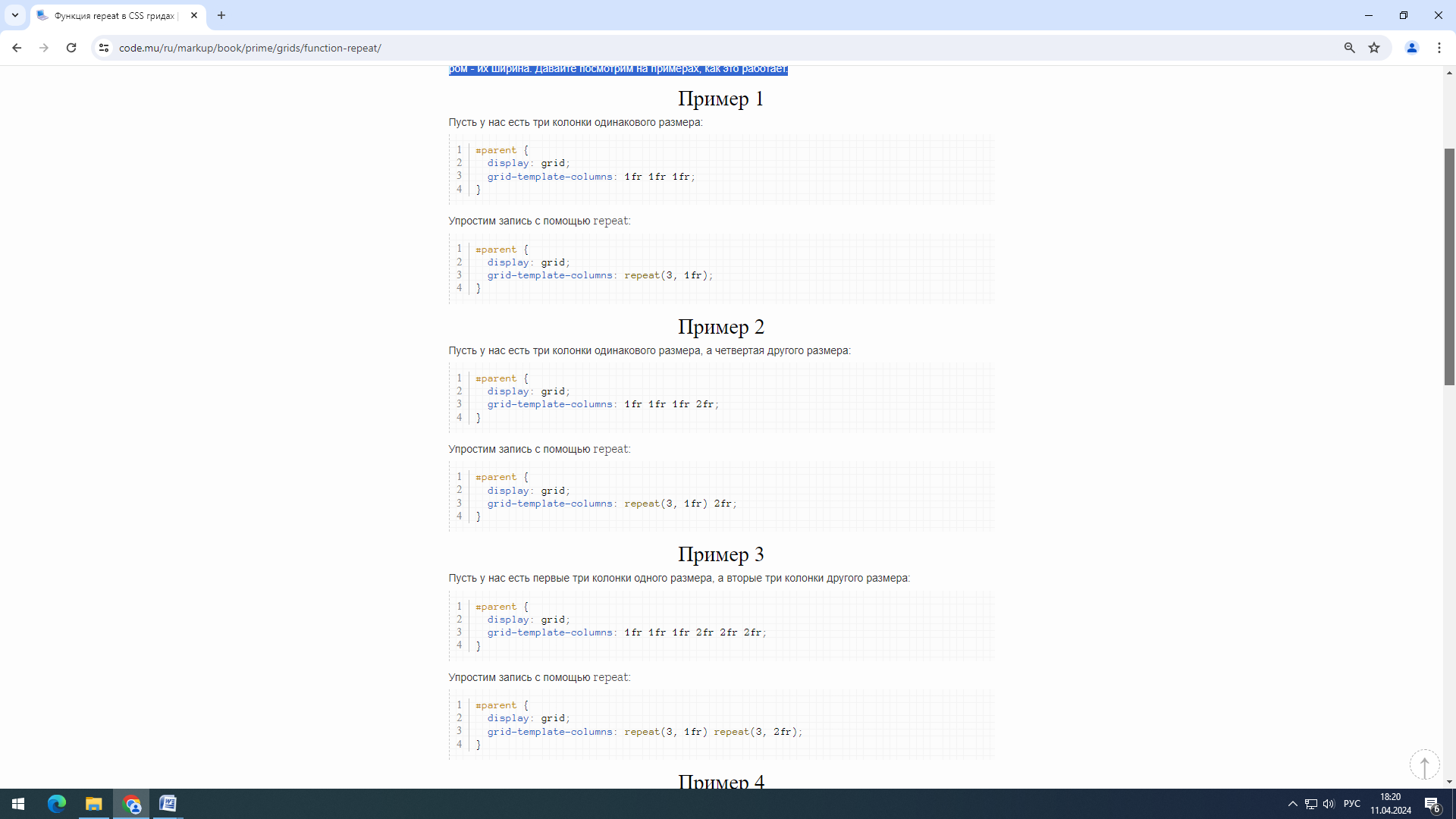
**Проценты и единицы fr в CSS гридах**

Также наряду с **fr** можно использовать значения в %, которые также определяют какую часть контейнера будет занимать столбец. При этом сначала будет высчитываться размер столбца в %, а оставшееся свободное пространство разобьется на фракции:

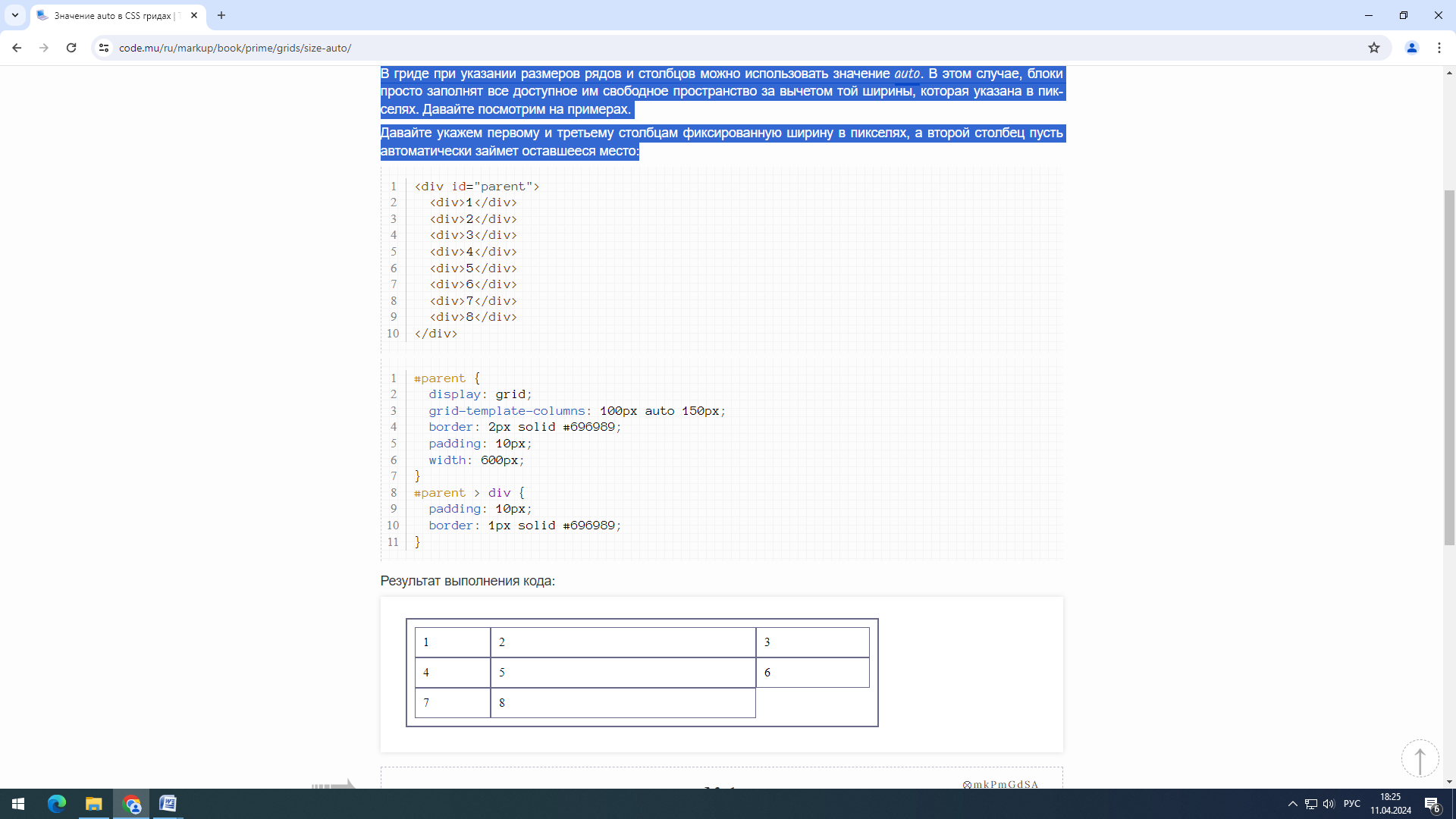


**Функция repeat в CSS гридах**

Если у вас несколько столбцов, для которых задаются одинаковые размеры, то можно упростить запись, использовав функцию ***repeat***. В первом параметре этой функции указывается количество столбцов, а во втором - их ширина. Давайте посмотрим на примерах, как это работает.



**Значение auto в CSS гридах**

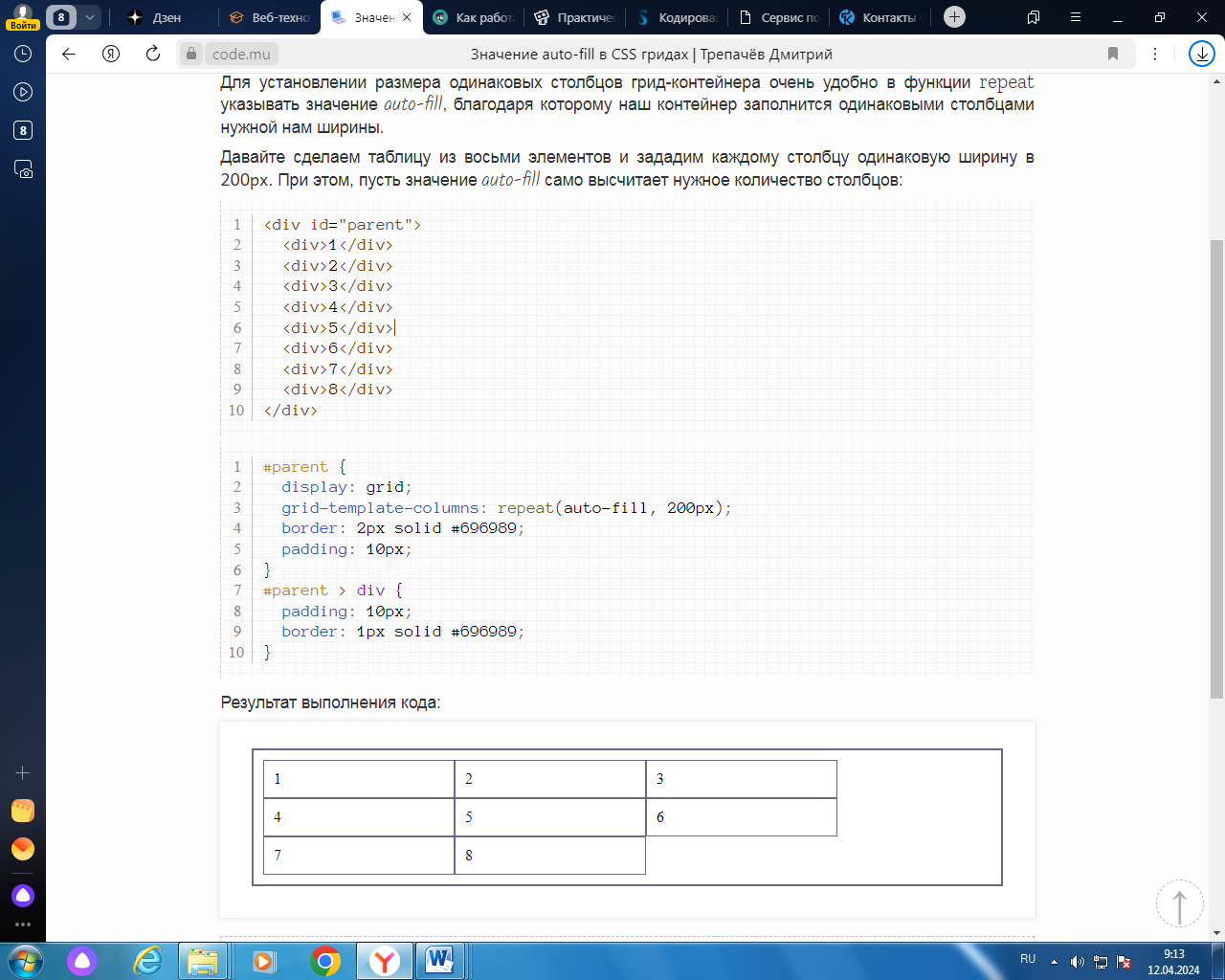
В гриде при указании размеров рядов и столбцов можно использовать значение ***auto***. В этом случае, блоки просто заполнят все доступное им свободное пространство за вычетом той ширины, которая указана в пикселях. Давайте посмотрим на примерах.

Давайте укажем первому и третьему столбцам фиксированную ширину в пикселях, а второй столбец пусть автоматически займет оставшееся место:

# Значение auto-fill в CSS гридах

Для установлении размера одинаковых столбцов грид-контейнера очень удобно в функции repeat указывать значение auto-fill, благодаря которому наш контейнер заполнится одинаковыми столбцами нужной нам ширины.

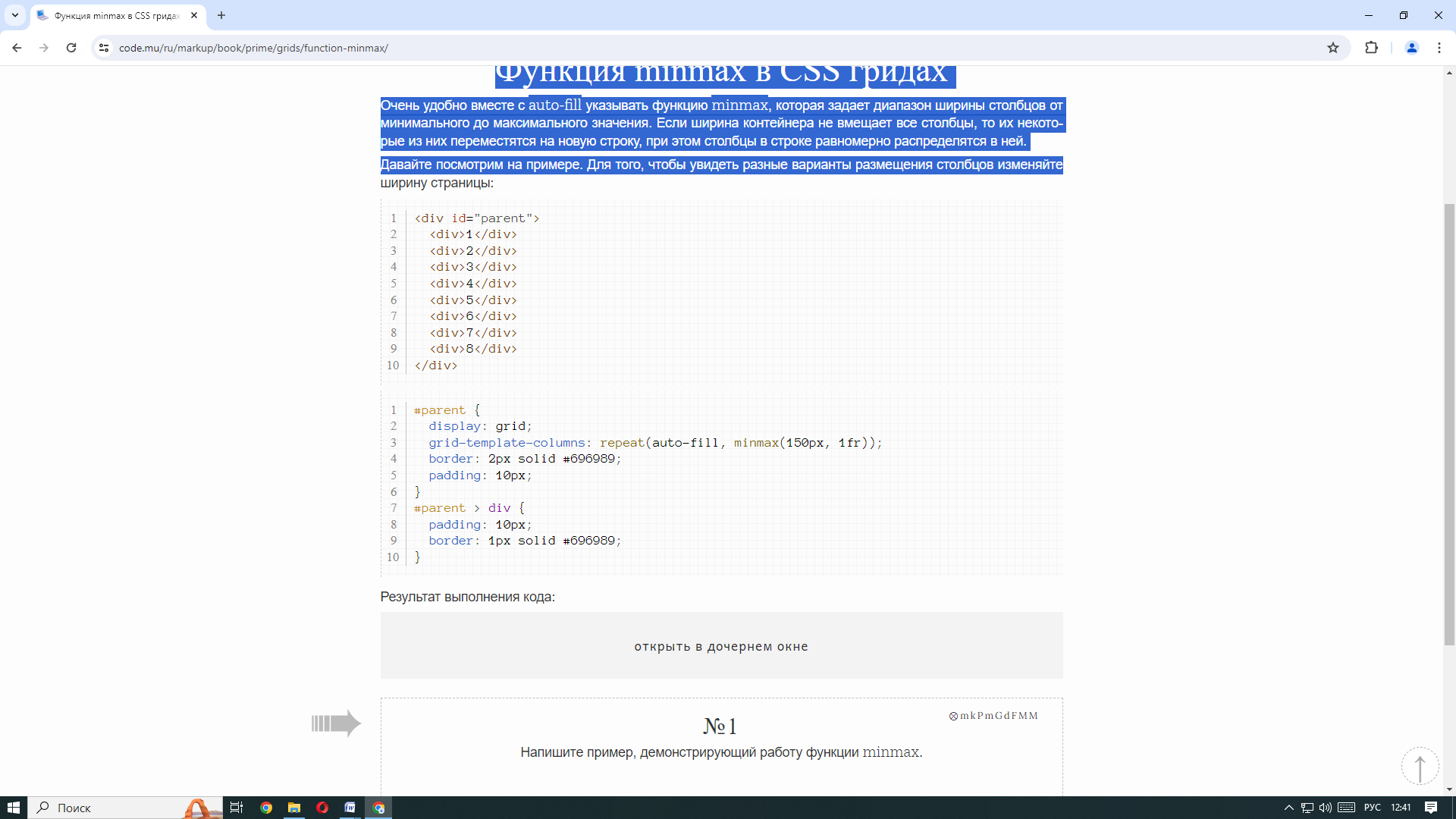
Давайте сделаем таблицу из восьми элементов и зададим каждому столбцу одинаковую ширину в 200px. При этом, пусть значение auto-fill само высчитает нужное количество столбцов:

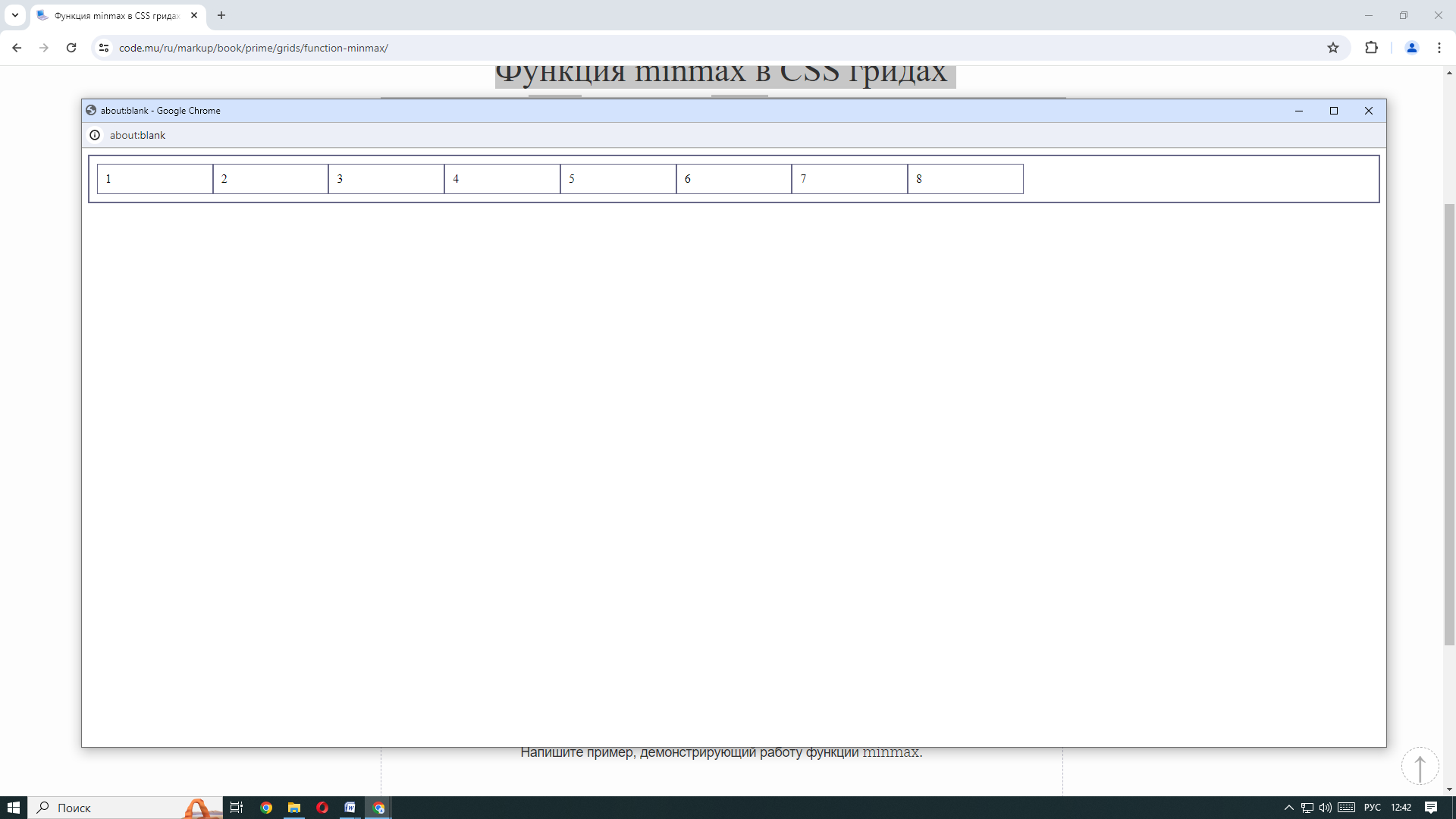


**Функция minmax в CSS гридах**

Очень удобно вместе с ***auto-fill***указывать функцию ***minmax***, которая задает диапазон ширины столбцов от минимального до максимального значения. Если ширина контейнера не вмещает все столбцы, то их некоторые из них переместятся на новую строку, при этом столбцы в строке равномерно распределятся в ней.

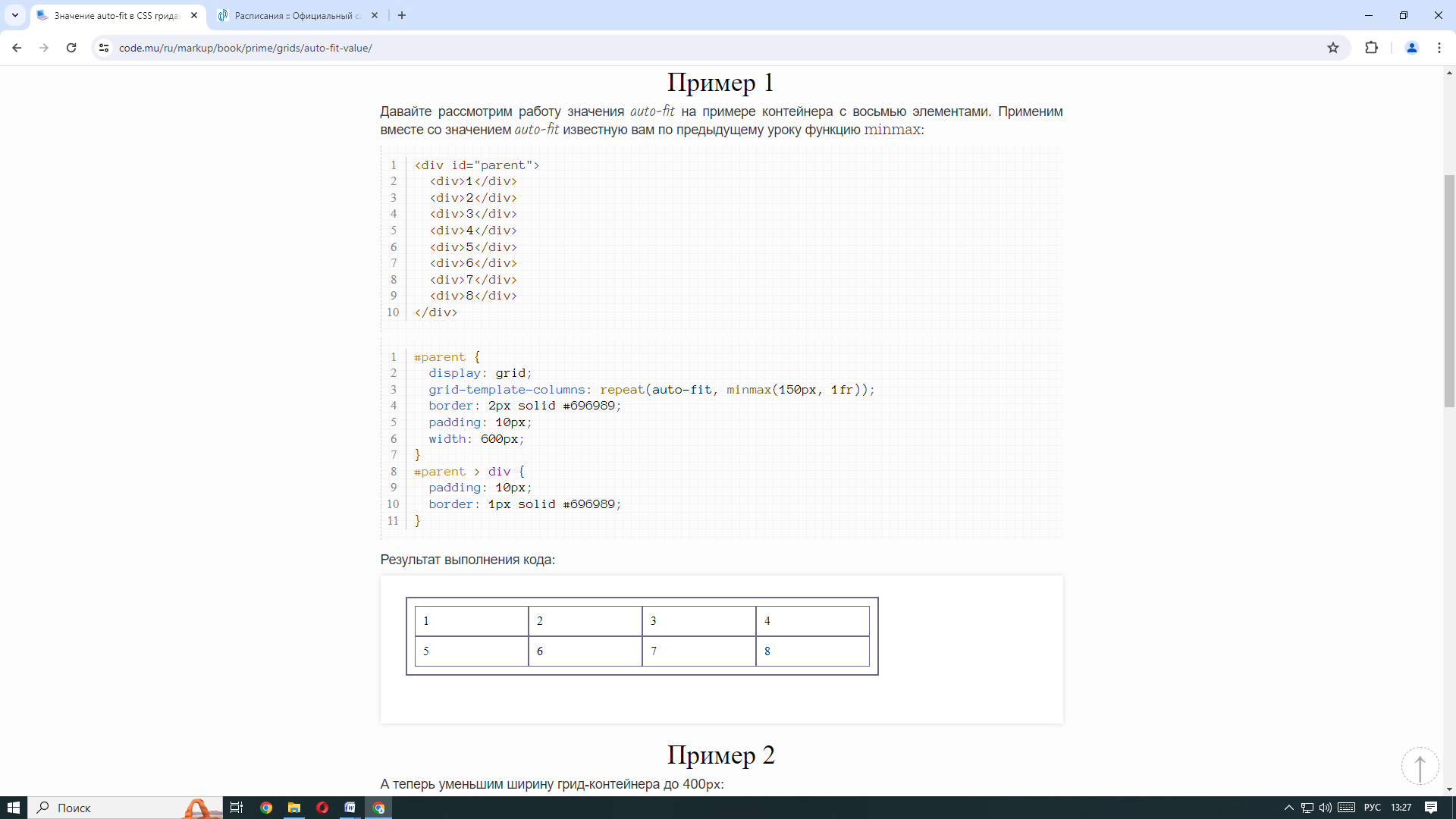
Давайте посмотрим на примере. Для того, чтобы увидеть разные варианты размещения столбцов изменяйте ширину страницы:





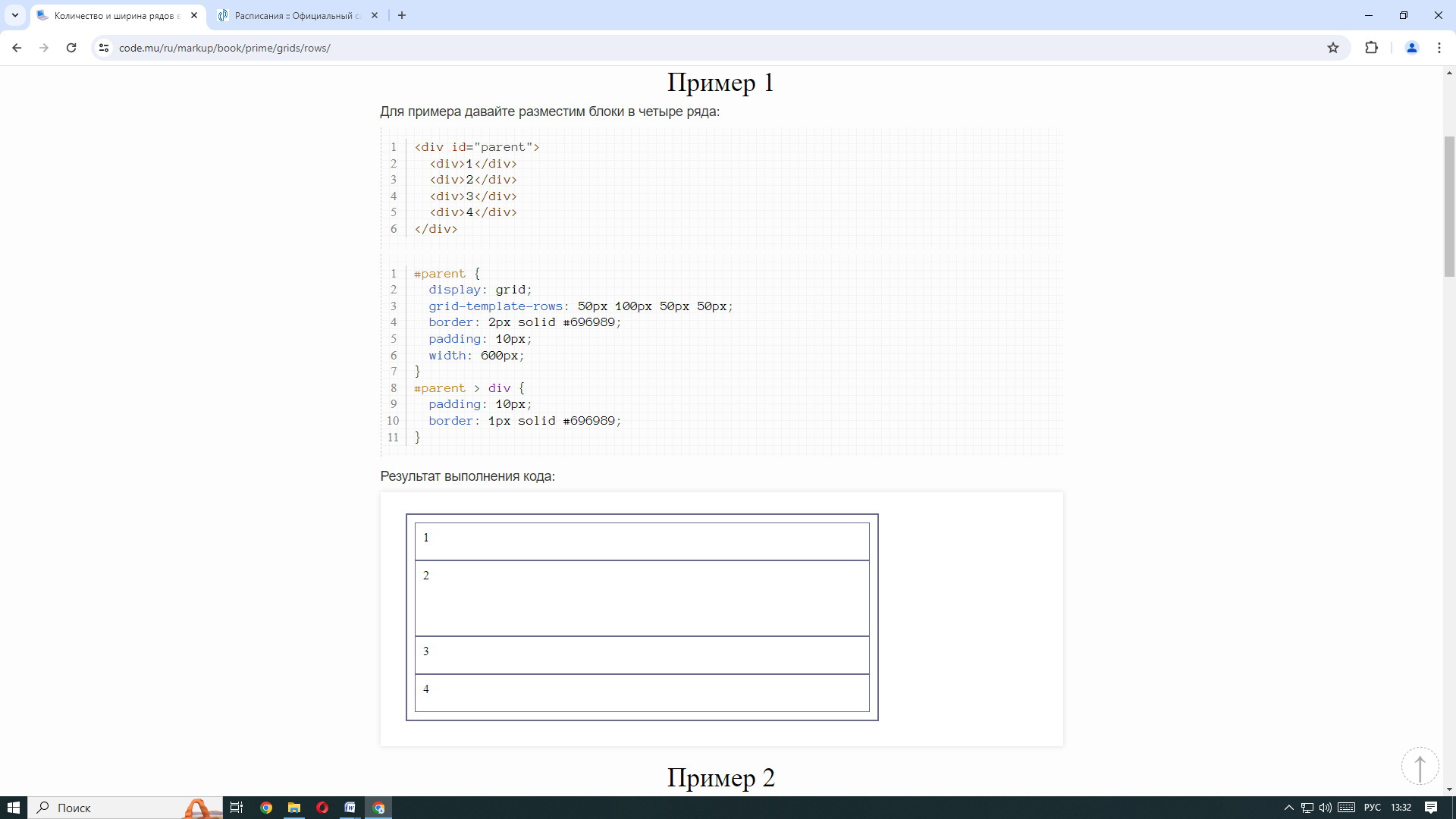
**Значение auto-fit в CSS гридах**

Рассмотрим значение ***auto-fit***, которое также применяется при указании столбцов одинакового размера совместно с функцией repeat. Его отличие от значения auto-fill заключается в том, что auto-fit подгоняет количество столбцов под доступную ширину контейнера, расширяя или сжимая их.



**Количество и ширина рядов в CSS гридах**

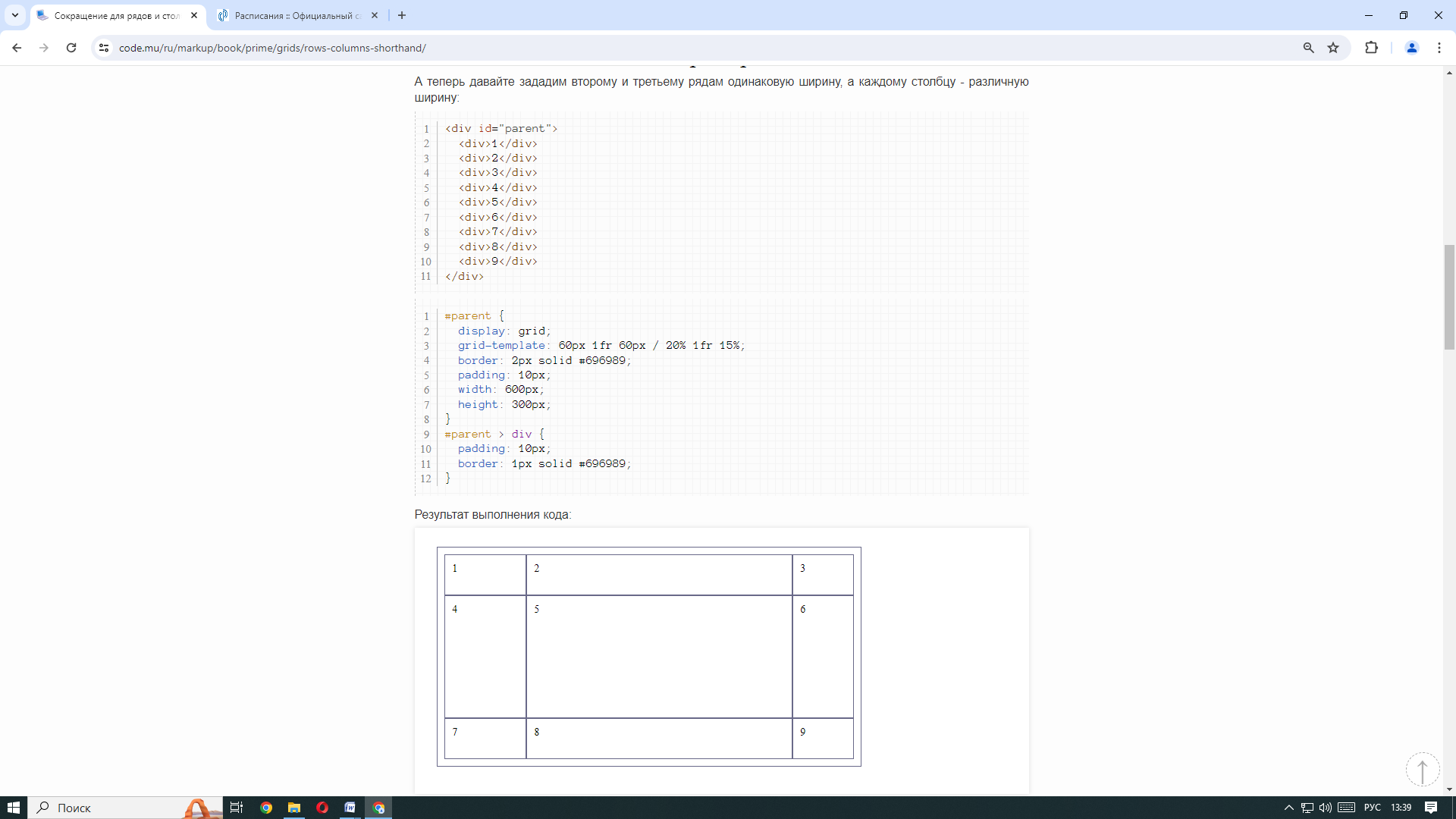
Элементы грида можно также разместить по рядам. Для этого предназначено свойство ***grid-template-rows***. Это свойство через пробел принимает размеры рядов. В нем можно применять те же значения, которые мы изучали и для столбцов.



**Сокращение для рядов и столбцов в CSS гридах**

Бывают ситуации, когда удобно указать размеры рядов и столбцов в краткой форме. Для этого мы используем свойство ***grid-template***, которое указывается в элементе-родителе и представляет собой сокращенную форму записи двух свойств ***grid-template-rows*** и ***grid-template-columns***.

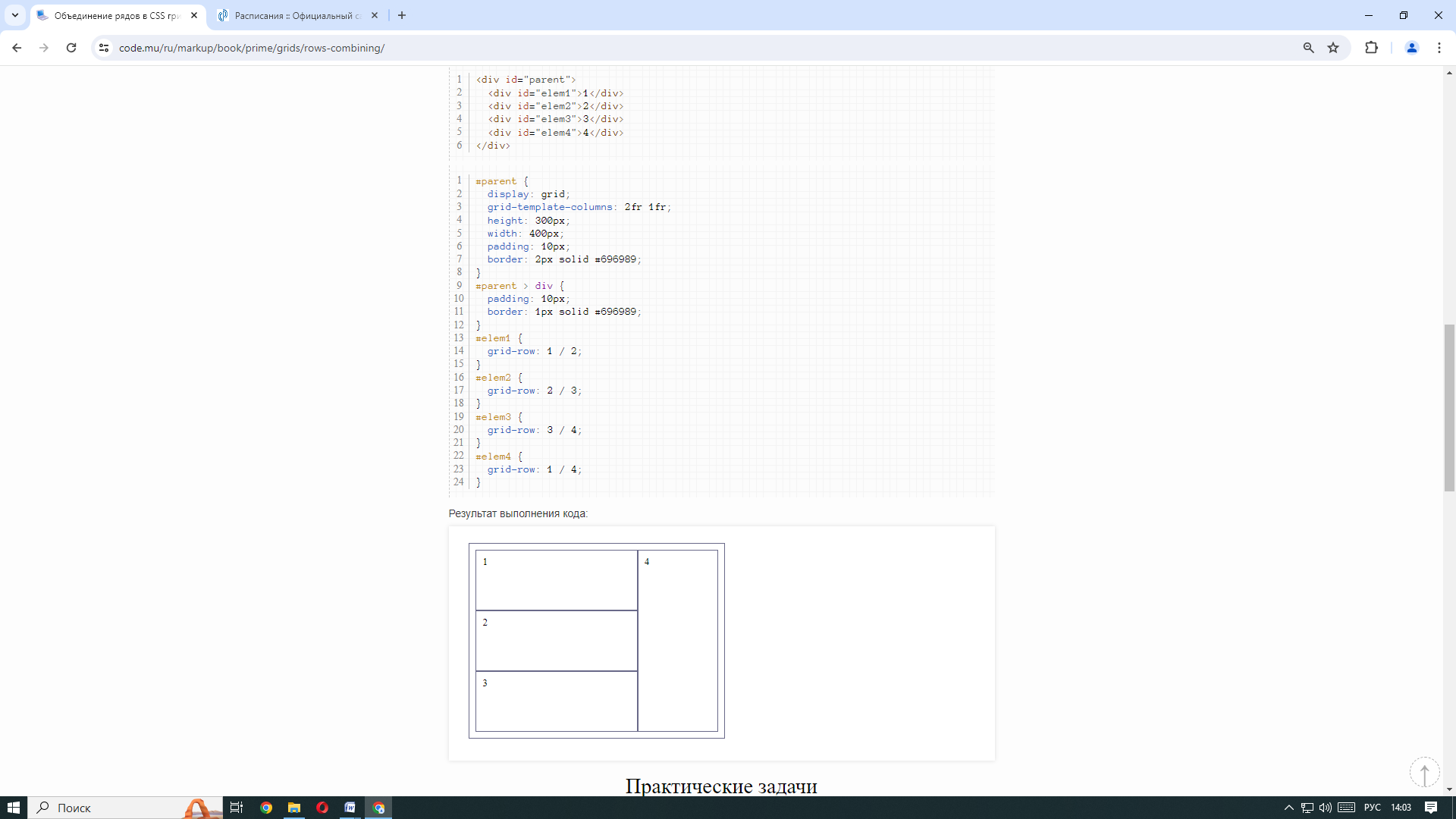
Ряды и столбцы для грид-контейнера указываются через слеш, их размеры определяются в единицах измерения.



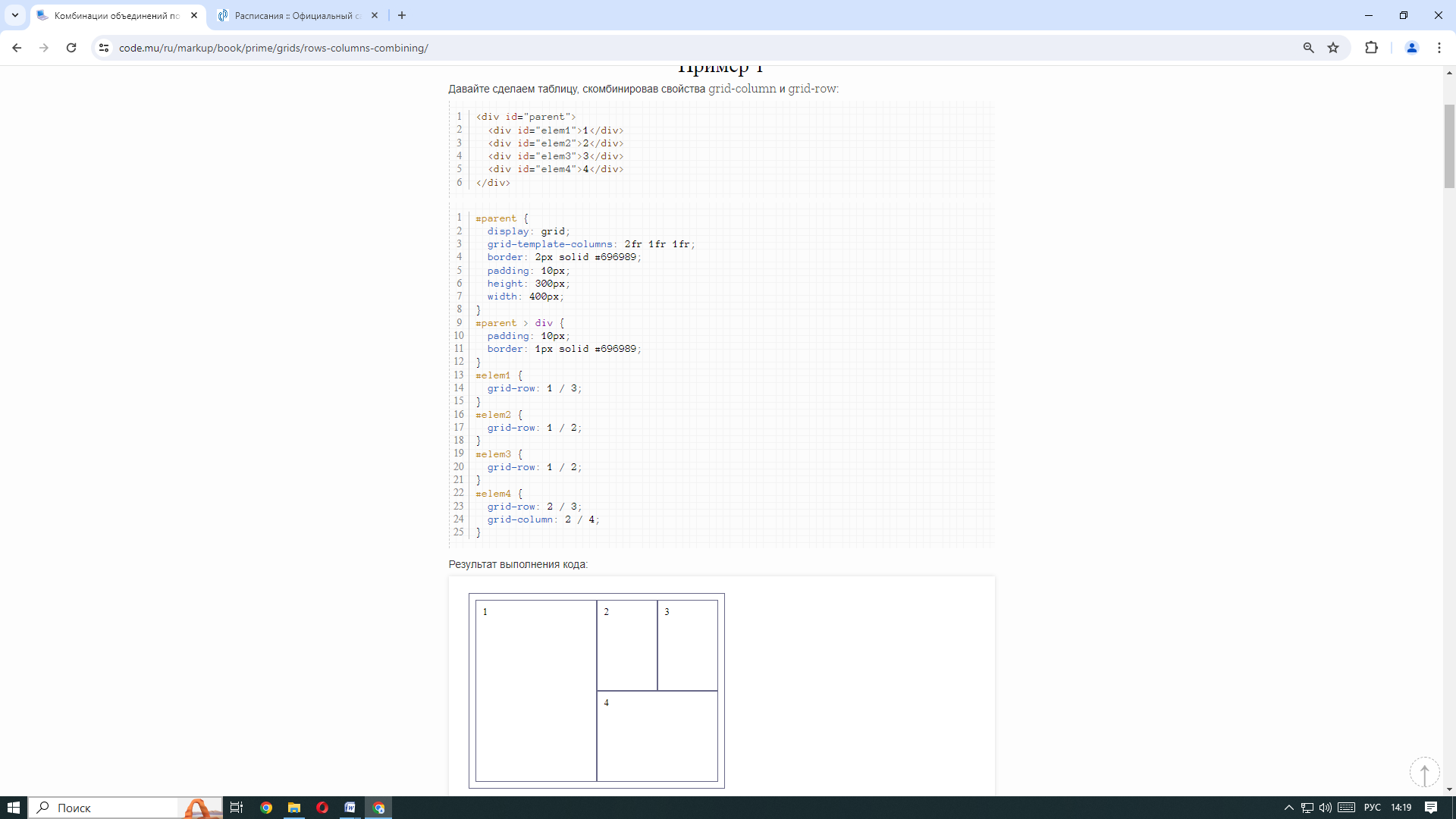
**Объединение рядов в CSS гридах**

Для того, чтобы элемент занял более одного ряда, ему следует задать свойство ***grid-row***. В этом свойстве через слеш задаются числа, указывающее начальную и конечную позицию элементов в гриде.

При этом элемент растянется от первой позиции до второй позиции (не включая ее). То есть значение 1 / 2 заставит элемент занять одну первую ячейку, а значение 1 / 3 заставит элемент занять первую и вторую ячейку (но не включая третью).



**Комбинации объединений по рядам и столбцам в CSS гридах**

****

**Расстояние между столбцами и рядами в CSS гридах**

С помощью свойства ***gap*** можно одновременно задать расстояние между столбцами и рядами грида.

Можно передать либо одно значение, либо два через пробел. Если передано одно значение, оно задает одновременно расстояние между столбцами и рядами. Если же передано два значения, то первое задает расстояние между рядами, а второе - между столбцами.

****