Чтобы сделать элемент грид-контейнером, нужно задать ему соответствующее значение свойства **display**:

**.container {**

**display: grid;**

**}**

В гриде элементы располагаются по двумерной сетке. То есть грид состоит из рядов и столбцов, располагающихся между линий, которые нумеруются по порядку. Одно «деление» грида называют ячейкой.

Существует также сокращённый синтаксис для этих свойств. Свойство **grid-column** объединяет в себе сразу два свойства: **grid-column-start/grid-column-end**

**grid-column-start: 1;**

**grid-column-end: 3;**

Аналогично, свойство **grid-row** — это сокращение для задания пары свойств: **grid-row-start/grid-row-end**

**grid-row-start: 1;**

**grid-row-end: -2;**

Если в свойстве **grid-row** или **grid-column** не задать второй параметр, то значение останется валидным, но применится только первый параметр.

Грид-элементы могут наслаиваться друг на друга, при этом они начинают себя вести как будто абсолютно спозиционированные, при этом на них так же действует свойство **z-index**. Чем больше значение **z-index**, тем выше грид-элемент в «стопке».

Чтобы задать гриду определённое количество столбцов и рядов, существуют свойства **grid-template-columns** и **grid-template-rows**.

Свойство **grid-template-columns** перечисляет количество и ширину будущих столбцов грида:

**.element {**

**grid-template-columns: 100px 200px 300px;**

**}**

Аналогично **grid-template-columns** работает и свойство **grid-template-rows**, только оно сообщает гриду, сколько рядов он будет содержать и какой они будут высоты:

**element {**

**grid-template-rows: 100px 200px 300px;**

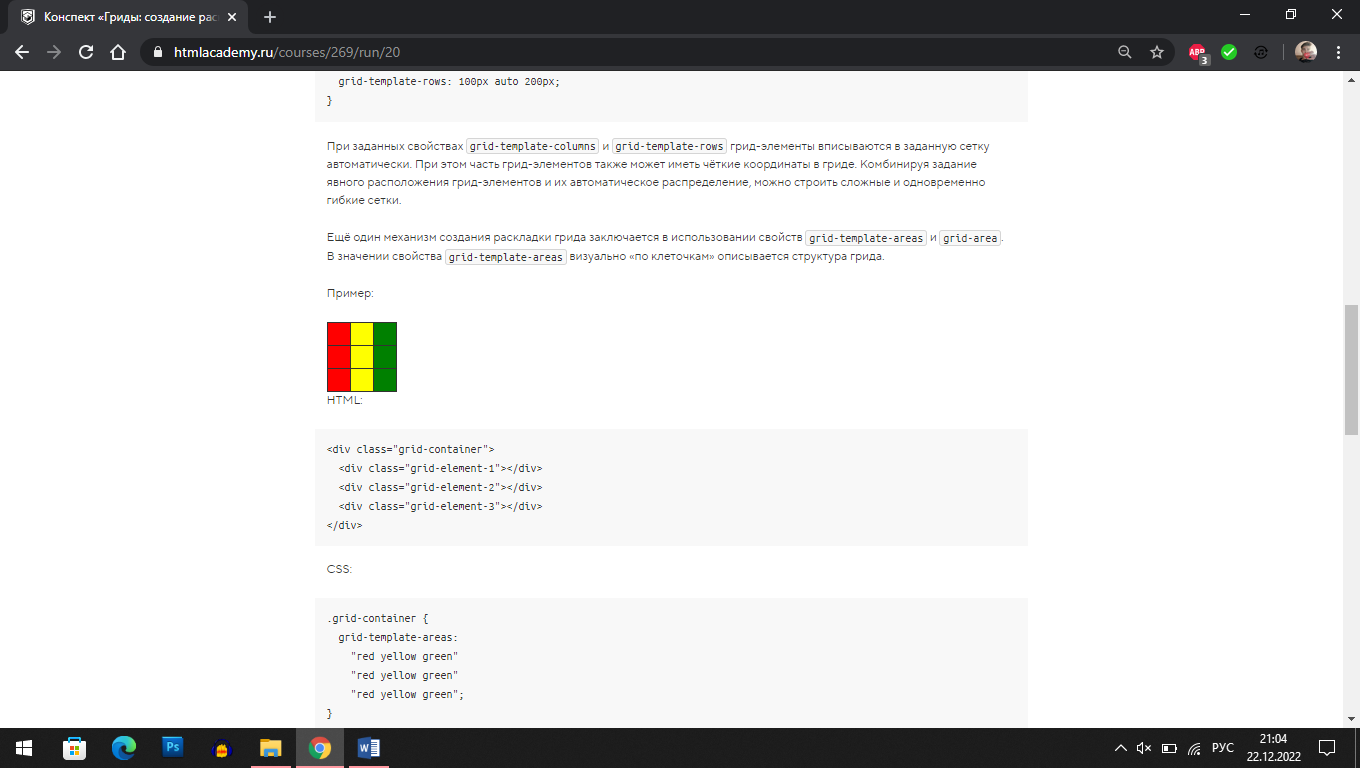
**}**

При заданных свойствах **grid-template-columns** и **grid-template-rows** грид-элементы вписываются в заданную сетку автоматически. При этом часть грид-элементов также может иметь чёткие координаты в гриде. Комбинируя задание явного расположения грид-элементов и их автоматическое распределение, можно строить сложные и одновременно гибкие сетки.

Ещё один механизм создания раскладки грида заключается в использовании свойств **grid-template-areas** и **grid-area**. В значении свойства **grid-template-areas** визуально «по клеточкам» описывается структура грида.

**grid-container {**

**grid-template-areas:**

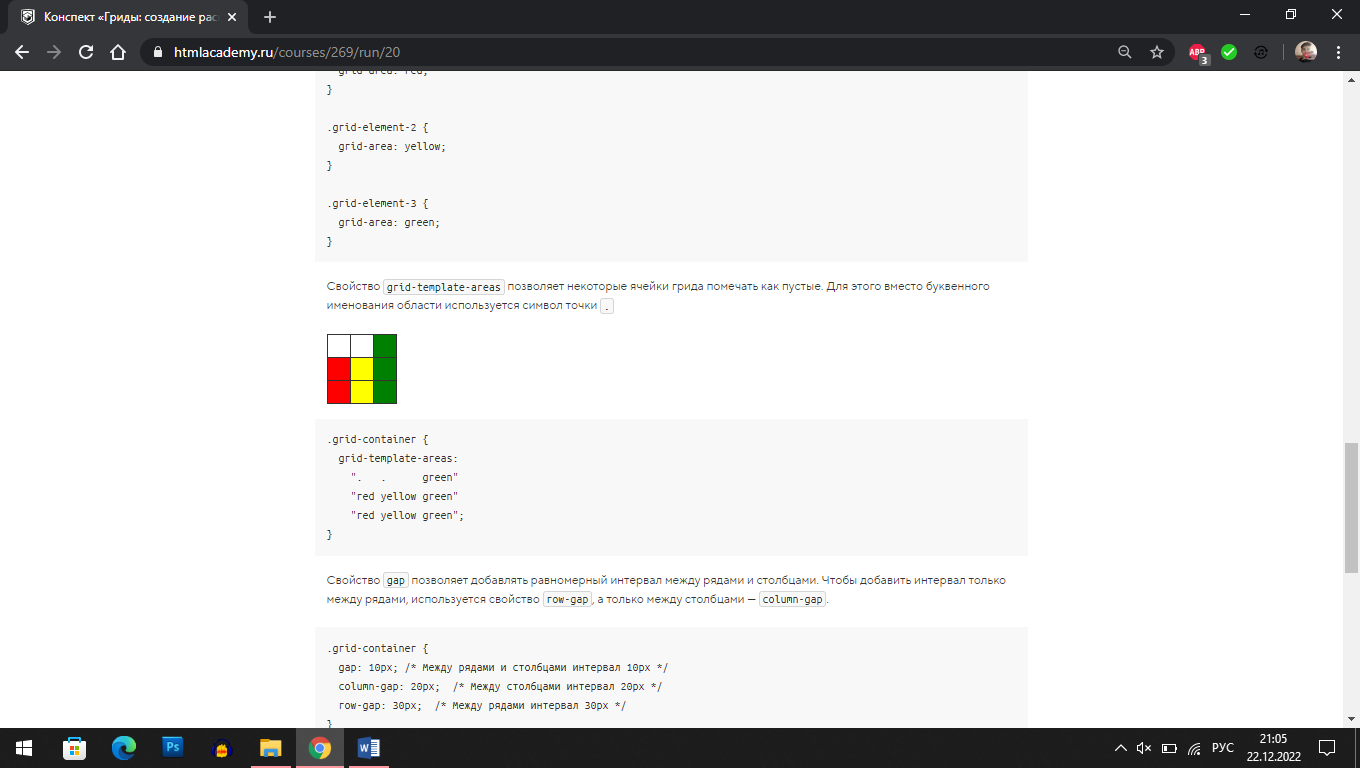
 **"red yellow green"**

**"red yellow green"**

**"red yellow green";**

**}**

Свойство **grid-template-areas** позволяет некоторые ячейки грида помечать как пустые. Для этого вместо буквенного именования области используется символ точки



.**grid-container {**

**grid-template-areas:**

**". . green"**

**"red yellow green"**

**"red yellow green";**

**}**

Свойство **gap** позволяет добавлять равномерный интервал между рядами и столбцами. Чтобы добавить интервал только между рядами, используется свойство **row-gap**, а только между столбцами — **column-gap**.