**CSS Grid**

Выравнивание используя CSS. Мы уже изучали через bootstrap и flexbox

Шпаргалка [https://css-tricks.com/snippets/css/complete-guide-grid/#:~:text=CSS%20Grid%20Layout%20is%20the,largely%20a%201%2Ddimensional%20system](https://css-tricks.com/snippets/css/complete-guide-grid/" \l ":~:text=CSS%20Grid%20Layout%20is%20the,largely%20a%201%2Ddimensional%20system).

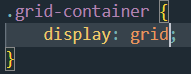
Документация Mozilla

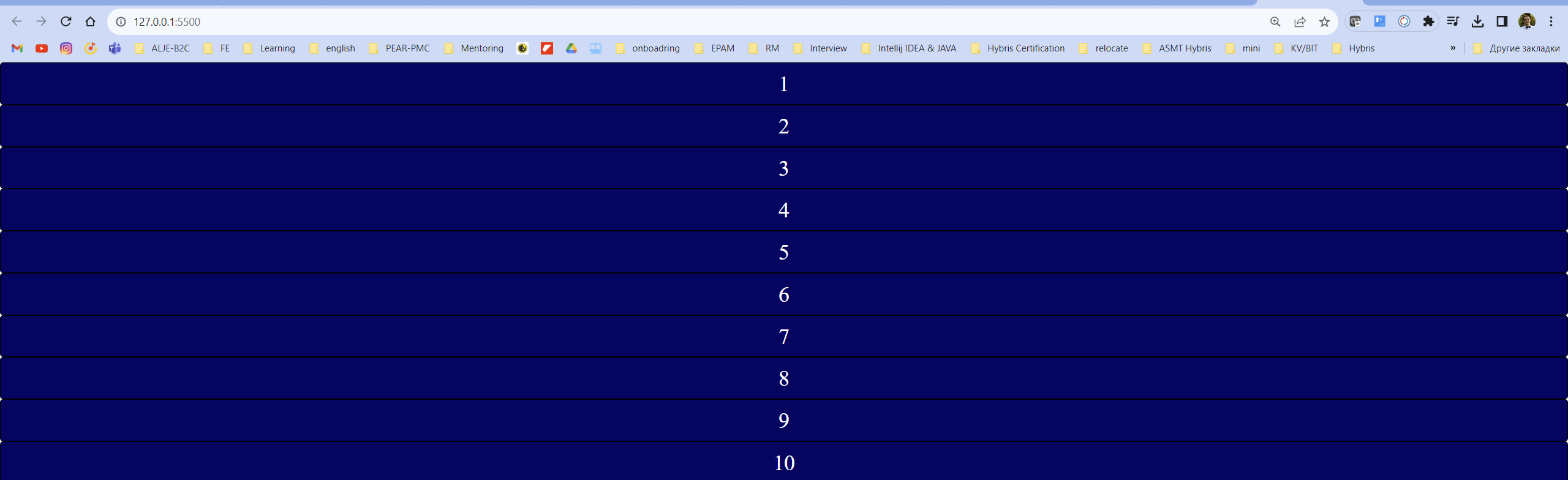
<https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout>

Can I use (поддержка браузеров)

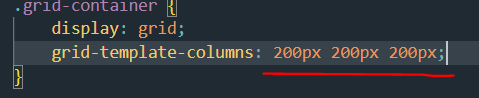
<https://caniuse.com/?search=css%20grid>

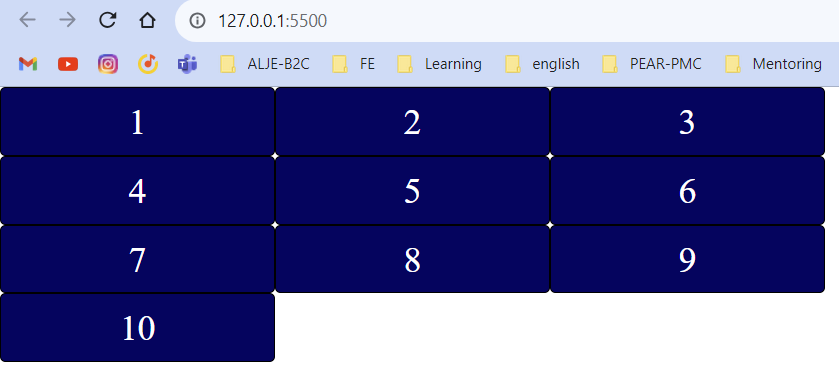
добавлены свойства **display: grid** и **inline grid** (похоже на flexbox)





Колонки **grid-template-columns** (пример 3 колонки) можно использовать другие единицы измерения

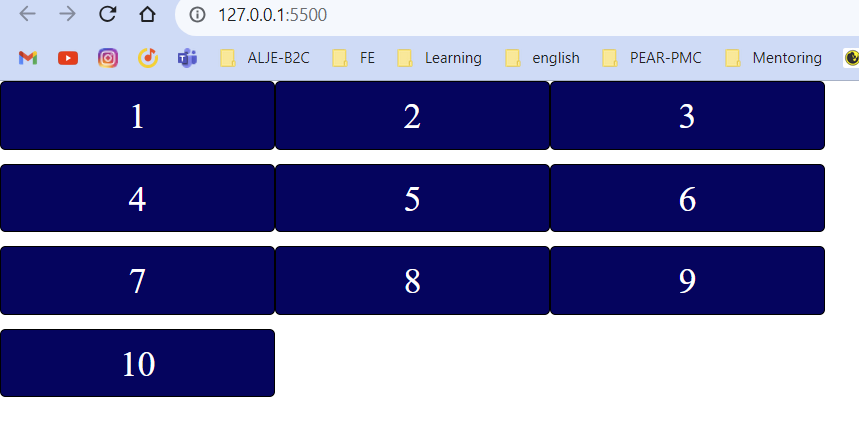




Grid elements могут называться **трэками**

Строки **grid-template-rows**





Зазоры из за того что у нас сами элементы по **50px**

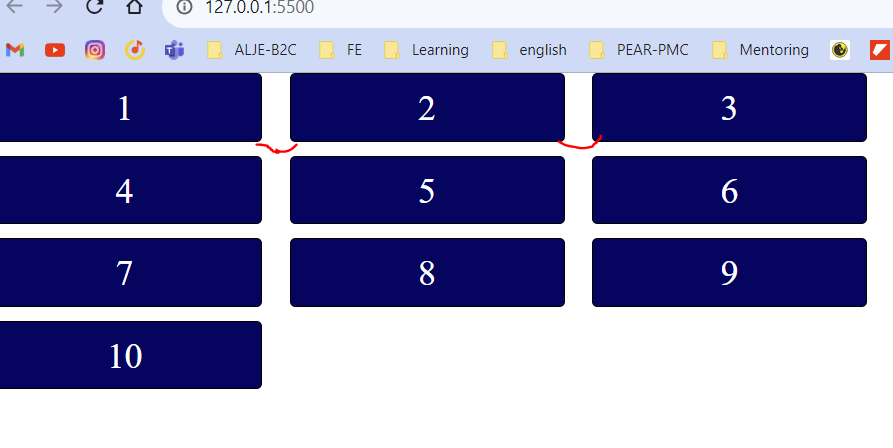
Разрывы между элементами

grid-column-gap: 20px;

deprecated

использовать

column-gap: 30px;



Для строк

row-gap: 10px;

Можно обьединить по принципу background

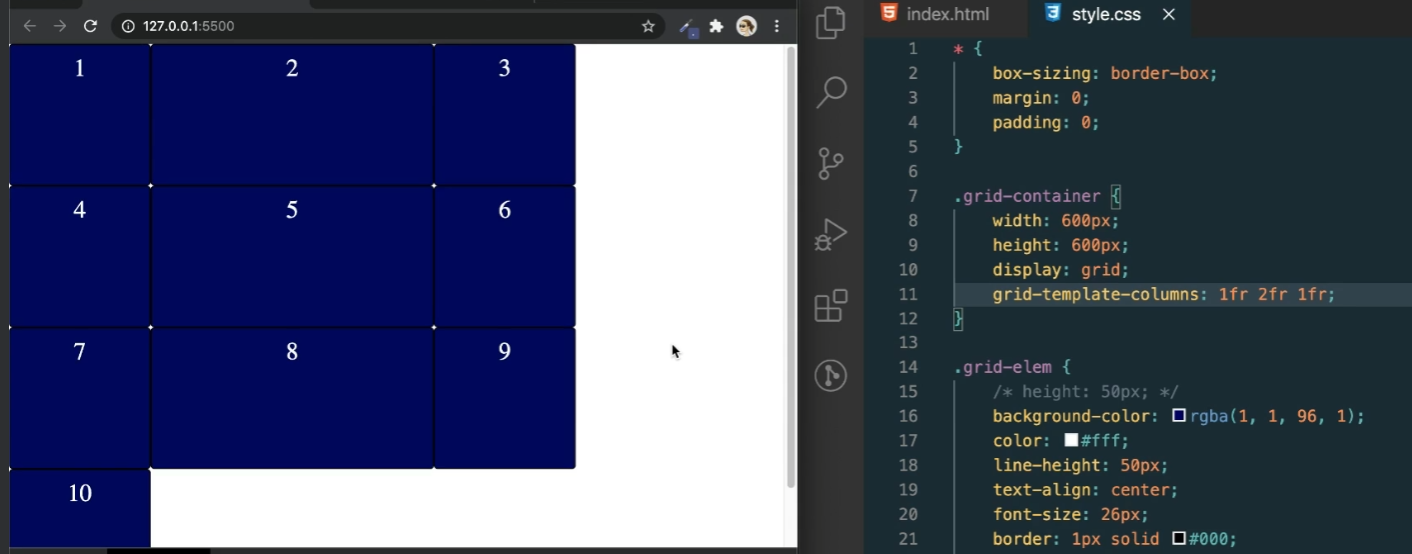
**Row-gap** и **column-gap** можно обьединить в **gap:20px;**

**Единица гибкости**

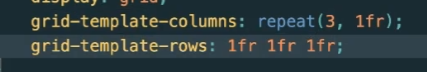
Отталкиваются от **height** или **width**

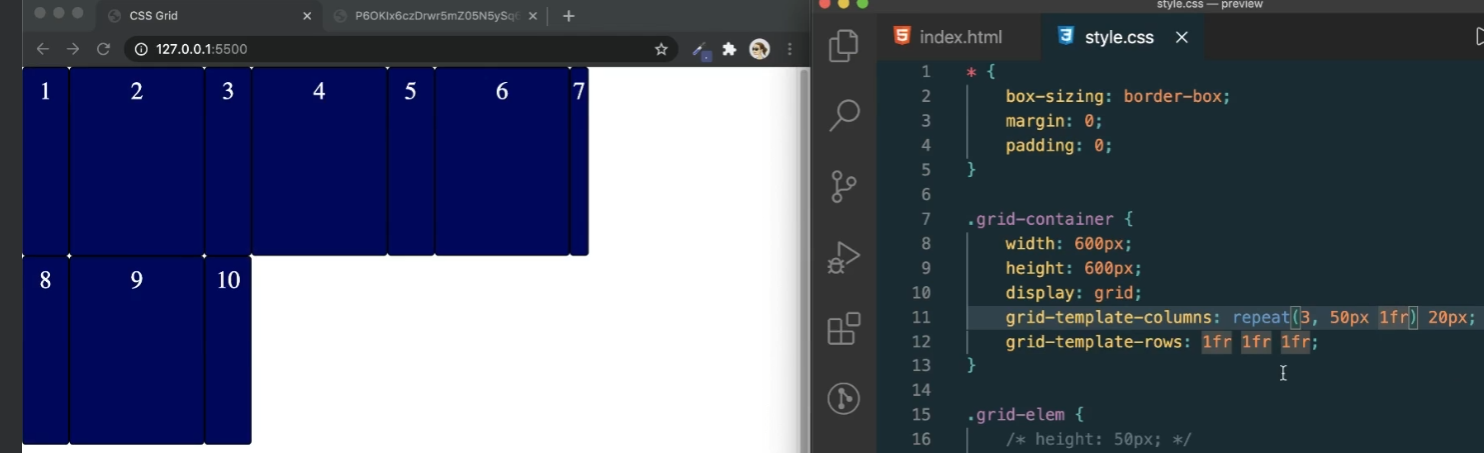
**fr -** <https://medium.com/@stasonmars/%D1%87%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-%D0%B5%D0%B4%D0%B8%D0%BD%D0%B8%D1%86%D0%B0-%D0%B3%D0%B8%D0%B1%D0%BA%D0%BE%D1%81%D1%82%D0%B8-fr-%D0%B2-css-%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BD%D1%8B%D0%BC-%D0%B8-%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%8B%D0%BC-%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%BC-2a3794c4444>





**repeat()** - https://developer.mozilla.org/en-US/docs/Web/CSS/repeat



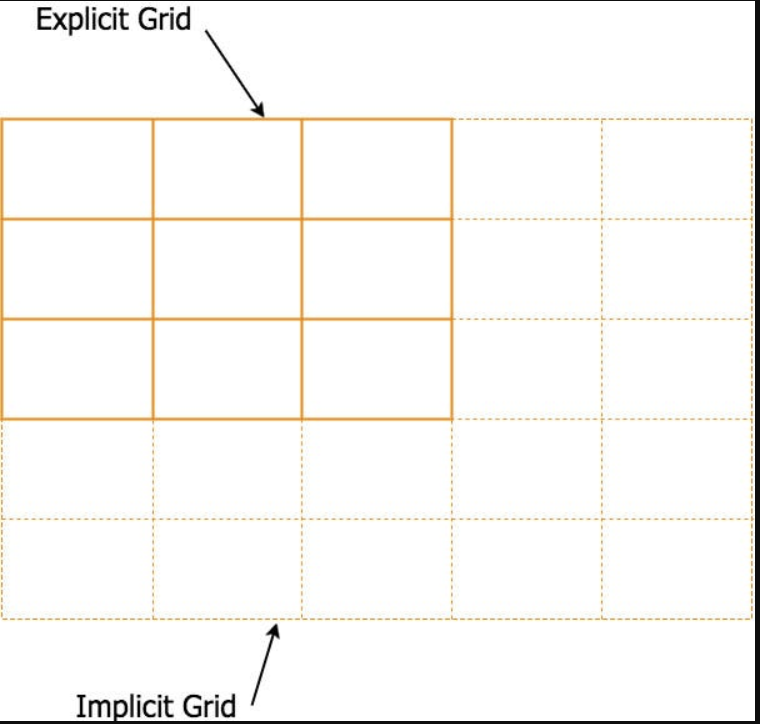
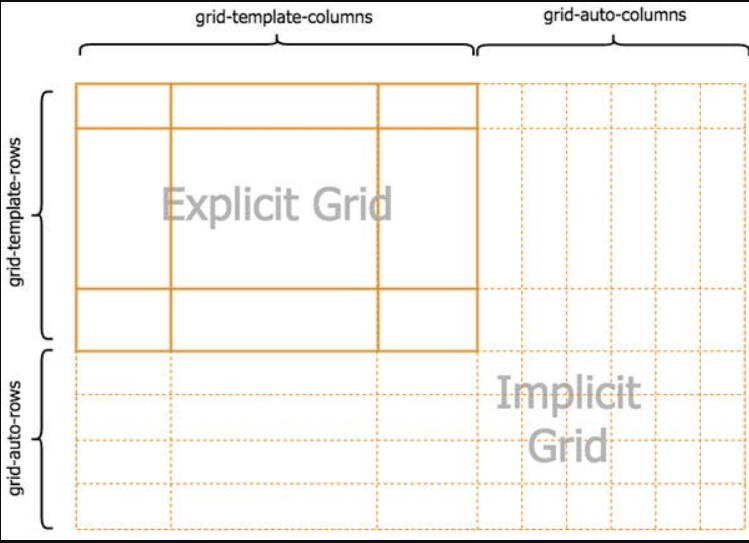


**Явные неявные Grid**

**Явные** – которые входит описан явно в CSS

**Неявные** – которые не входит (описали 3 элемента, а унас 4)

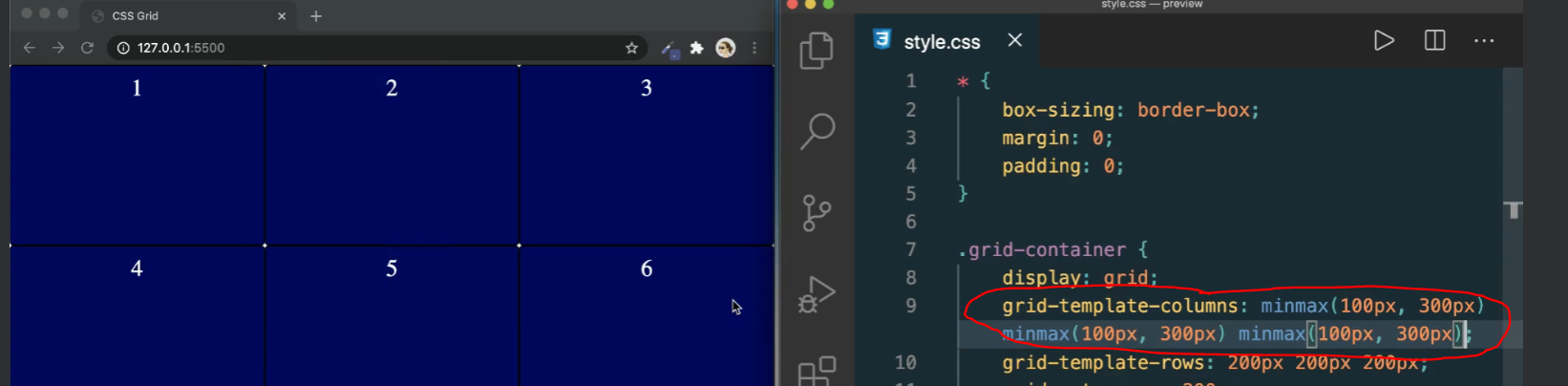
**Grid-auto-columns, Grid-auto-rows**

Figma [https://www.figma.com/file/nM2g64w6wGhxKH0cmtVnja/CSS-Grid-Example?type=design&node-id=0-1&mode=design](https://www.figma.com/file/nM2g64w6wGhxKH0cmtVnja/CSS-Grid-Example?type=design&node-id=0-1&mode=design) 

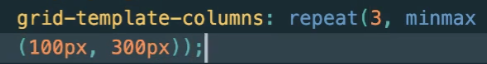
**Функция minmax() и масштабирование треков**

Минимально максимальное значение для трека (элемента)

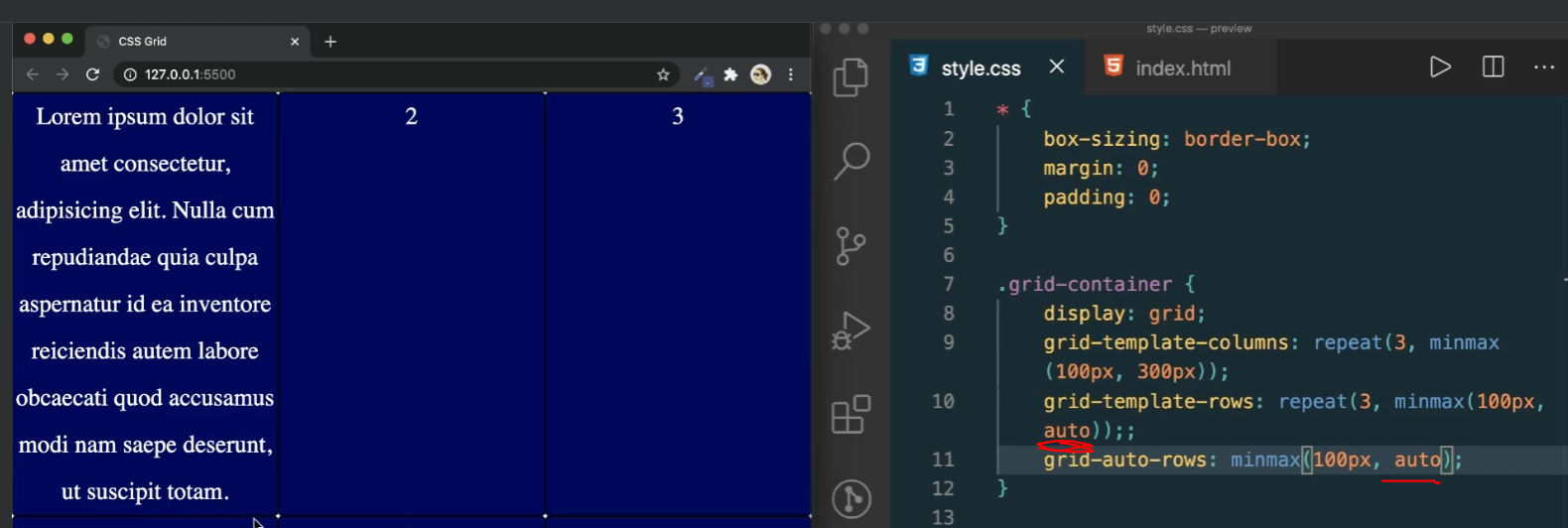
**minmax()** https://developer.mozilla.org/en-US/docs/Web/CSS/minmax



или



Команда **auto** – выравнивает автоматически



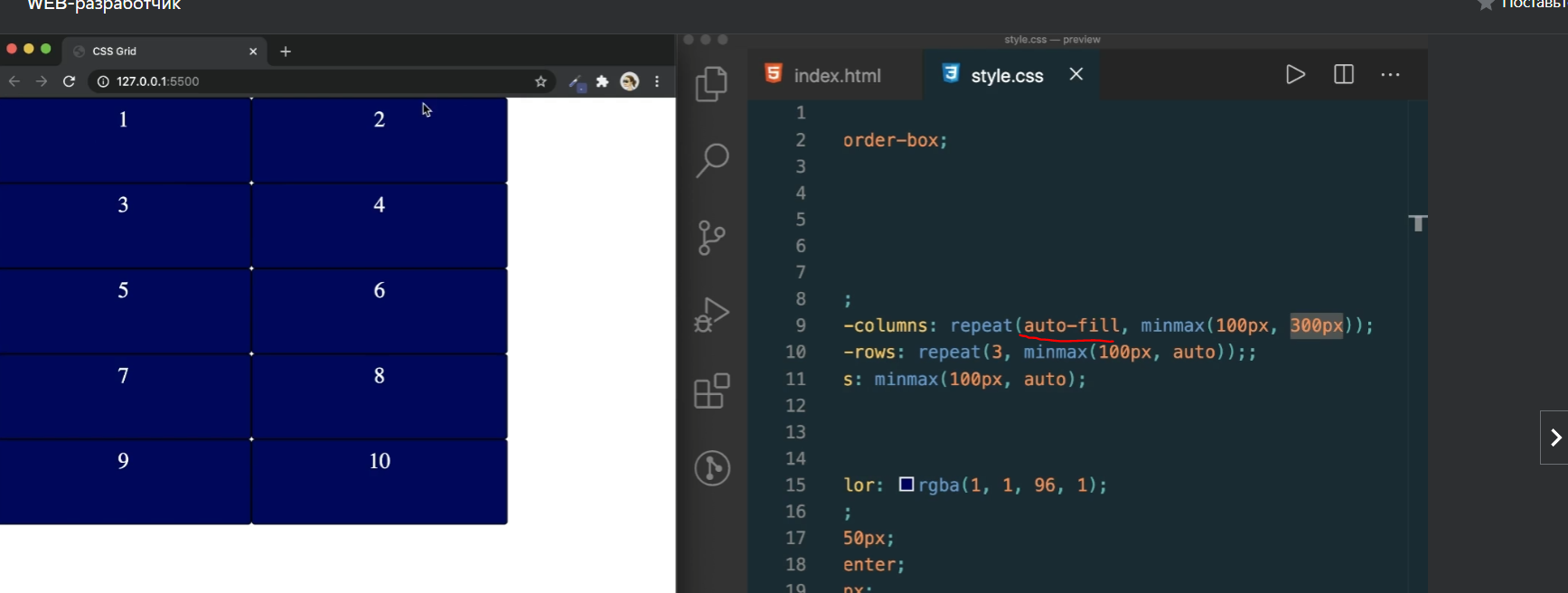
**auto-fill** автоматически заполняет пространство браузера (оставляет пустое место)

**auto-fit** работает также,за исключением не оставляет пустое место

<https://medium.com/@stasonmars/%D0%BA%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%8E%D1%82-auto-fill-%D0%B8-auto-fit-%D0%B2-css-grid-7d903a6c678e>

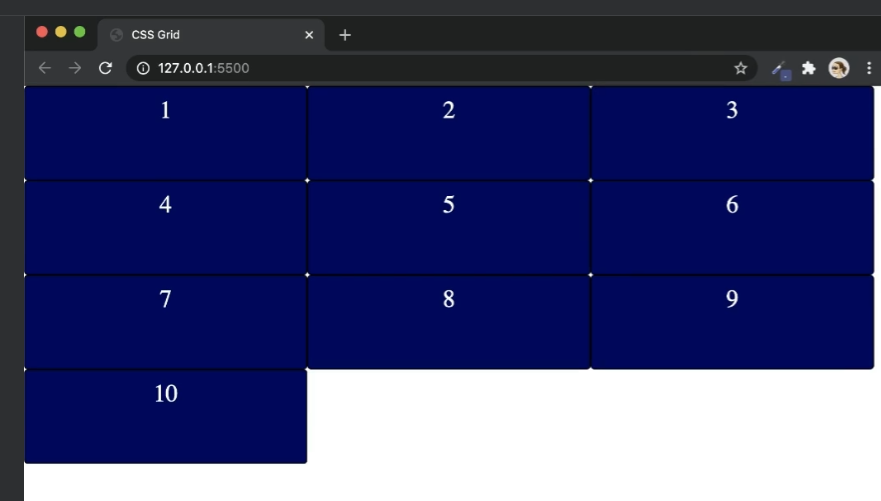
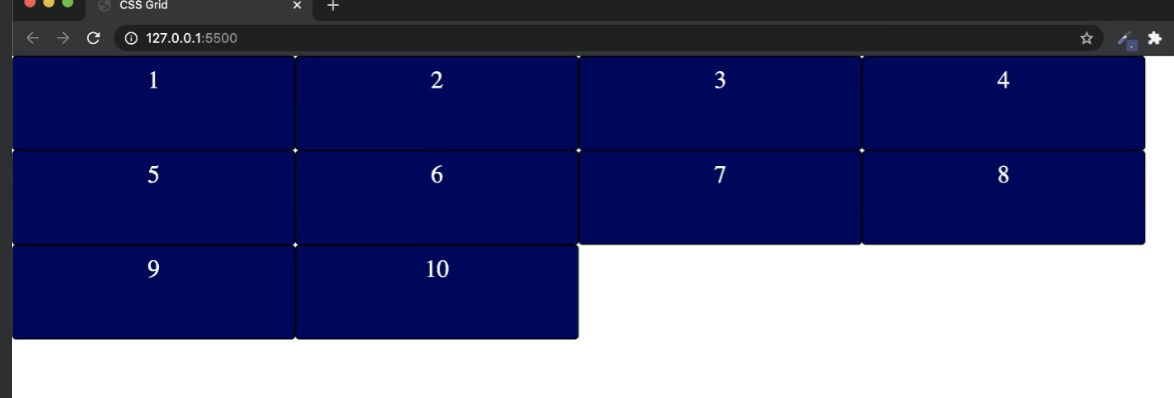
**grid-auto-flow** можно указать что использовать для автоматического выравнивания, строки или колонки (куда будет расти неявный грид в виде строк или колонок)

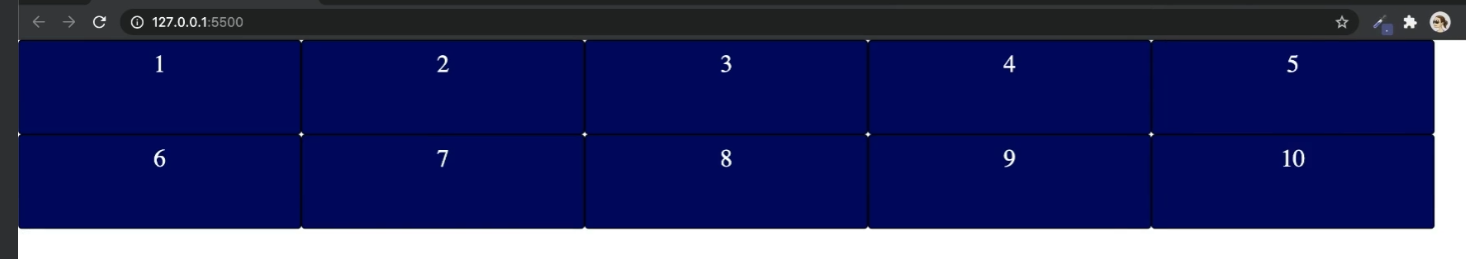
**row** by default

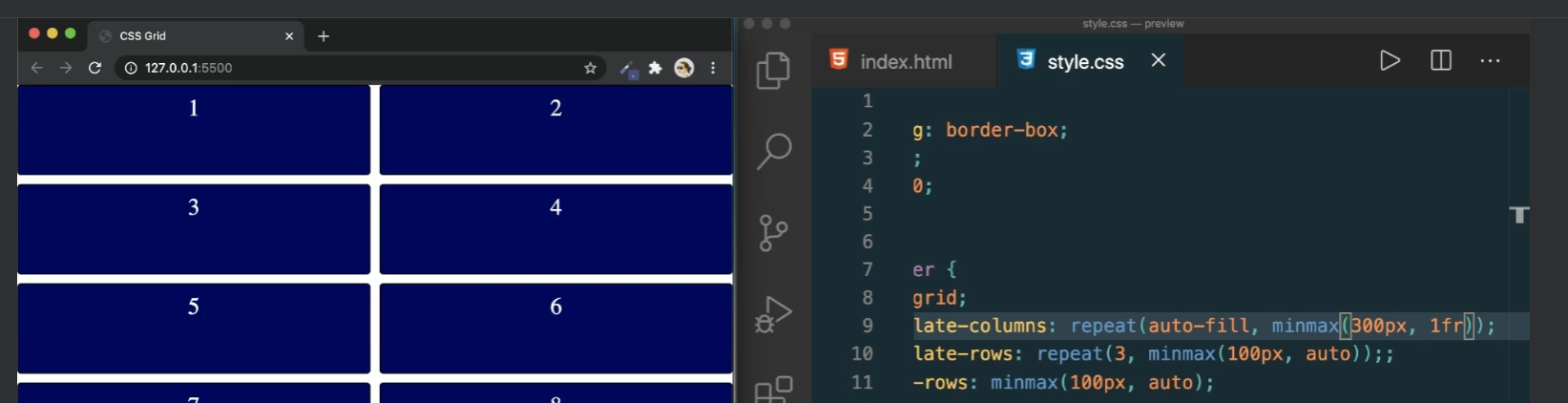


Понимает что не влезет плоэтому строит только 2 колонки

Растягиваем

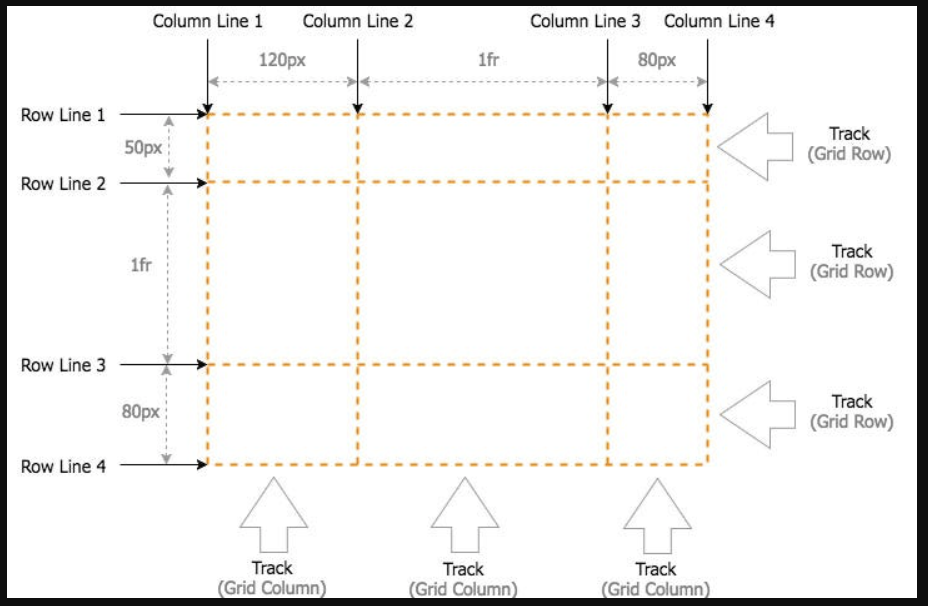


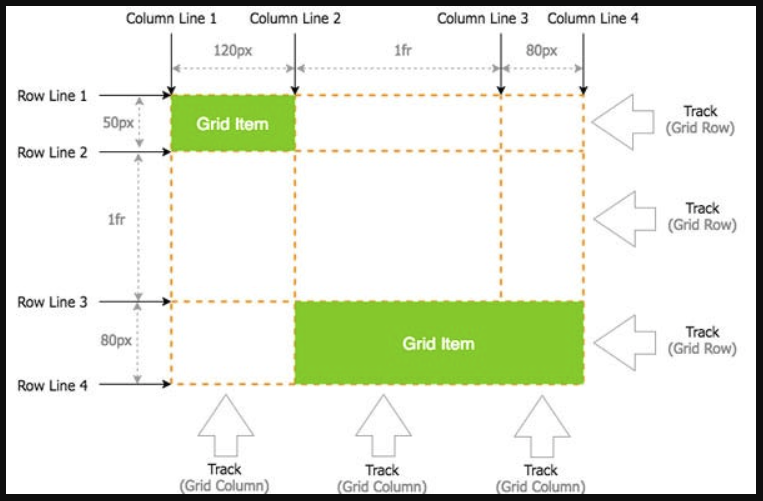


**Позиционирование треков**

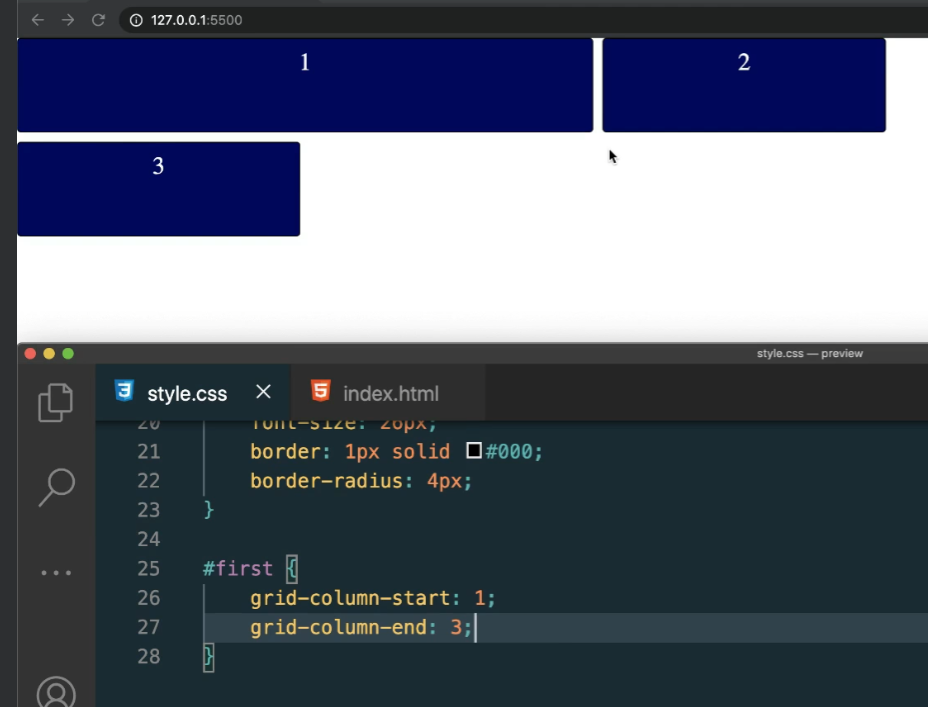
Пример

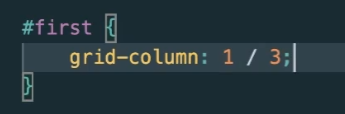




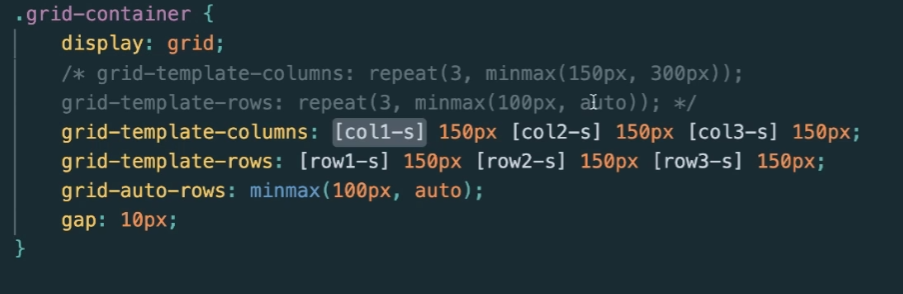


Пример

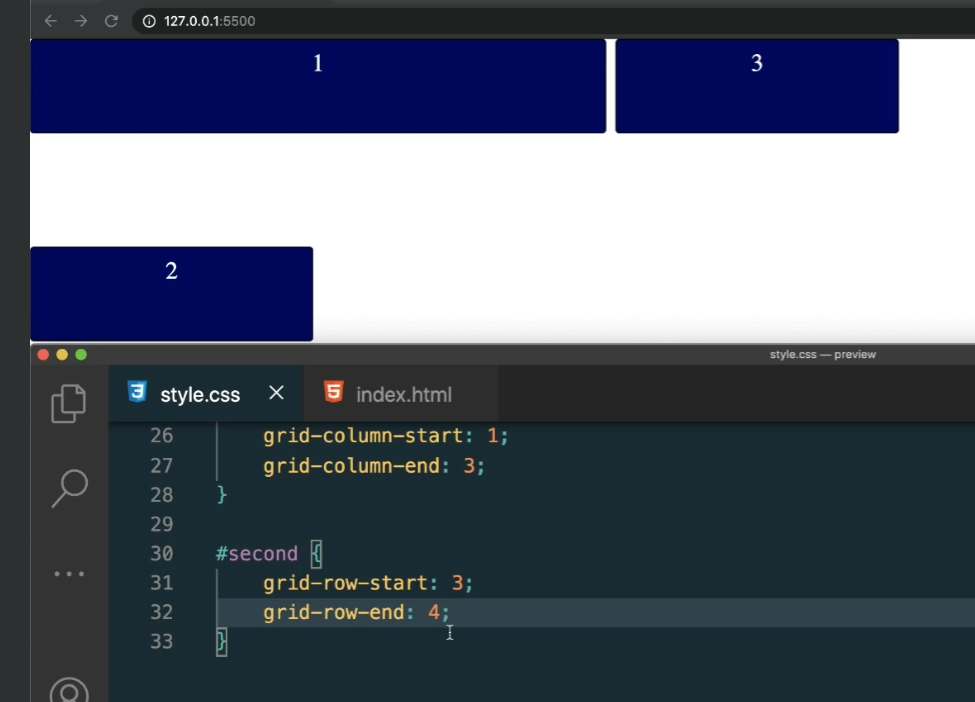




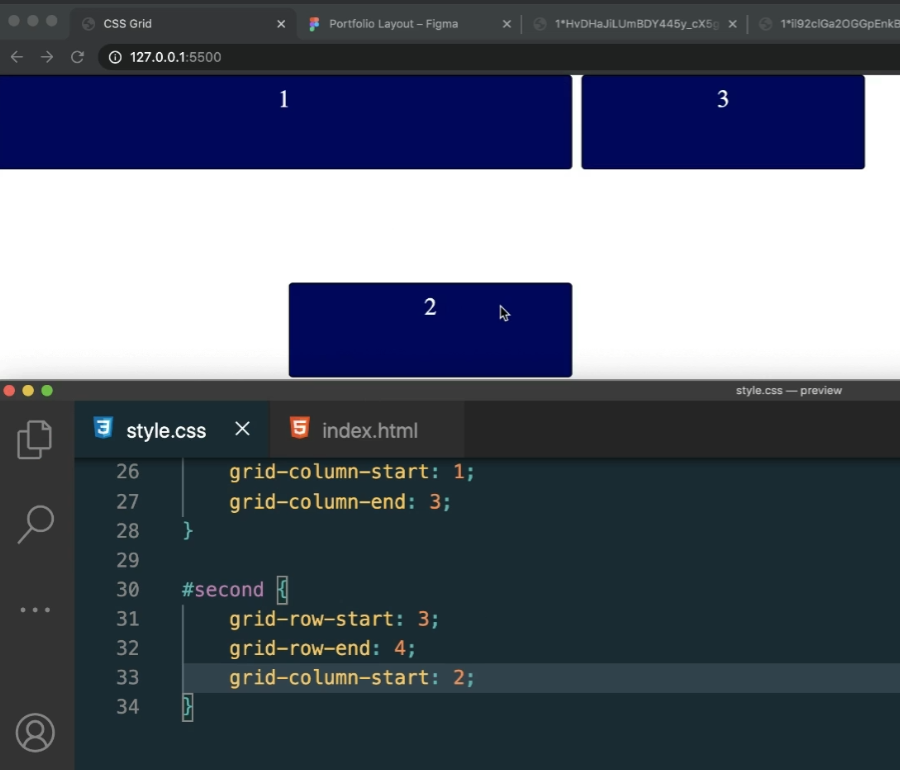
Можно сделать не через индекс, а через имя элемента (-s start):

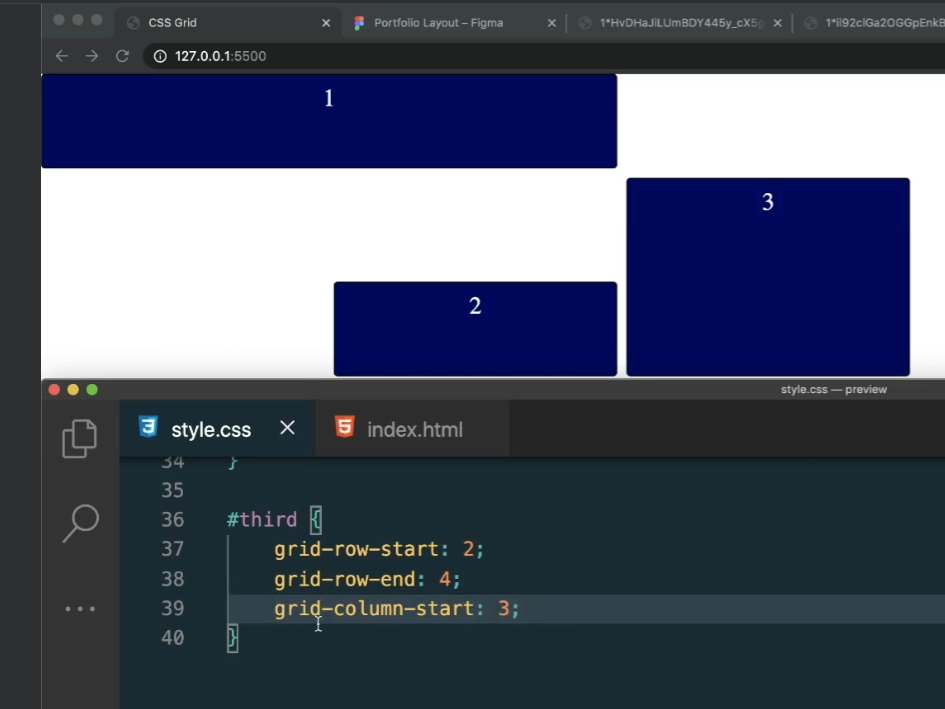














Выравнивание треков

Align-self

Self – само на себя

Items – на элементы внутри

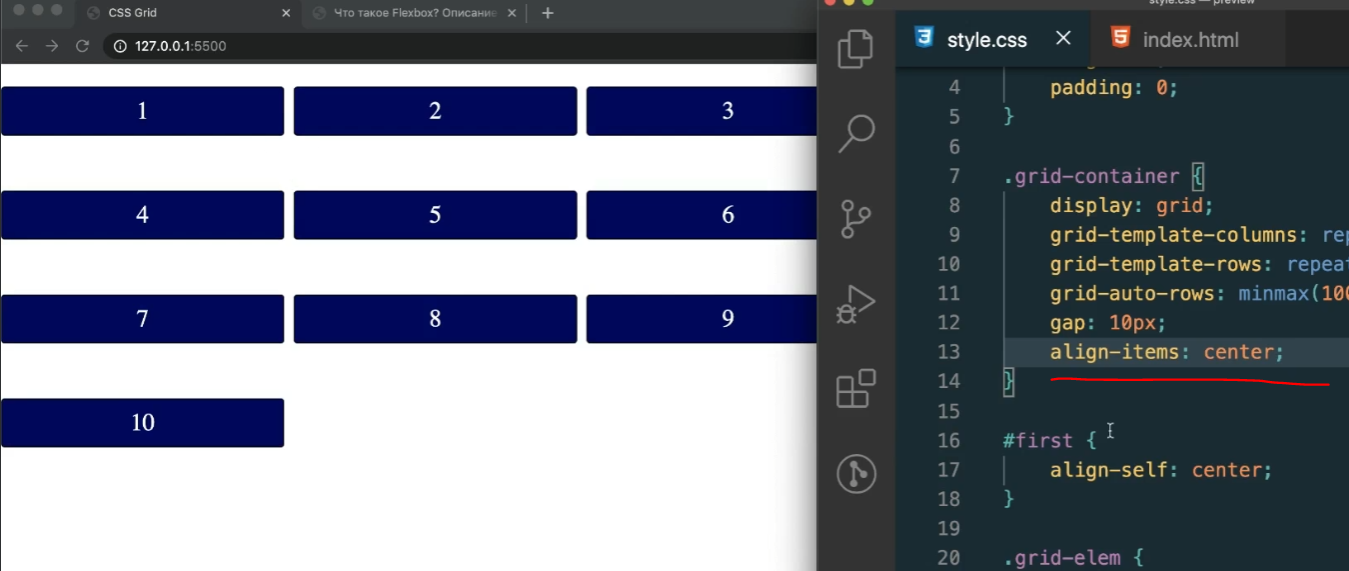
<https://web.archive.org/web/20210718095533/https://html5.by/blog/flexbox/>

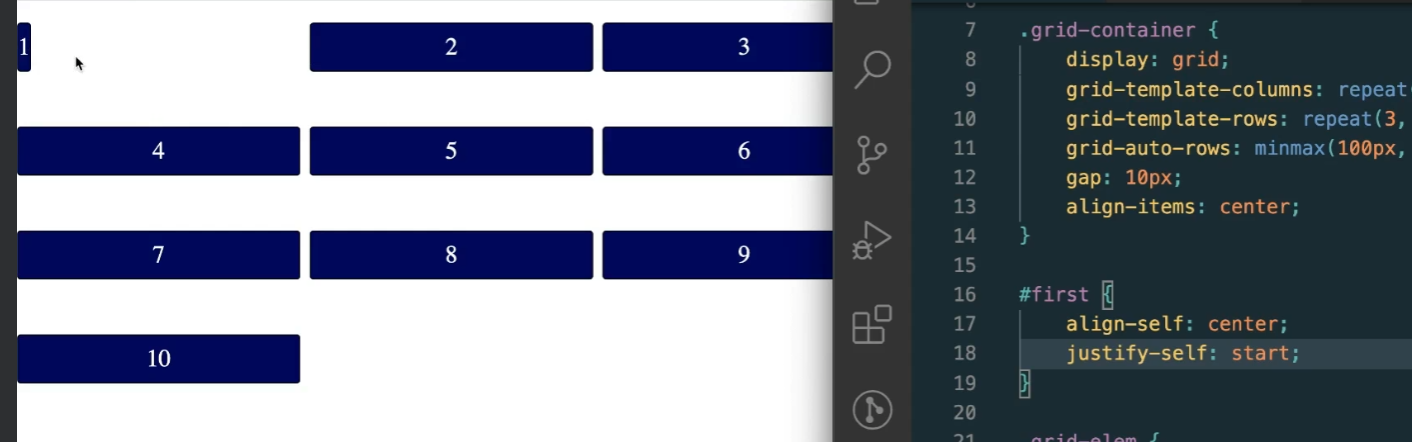


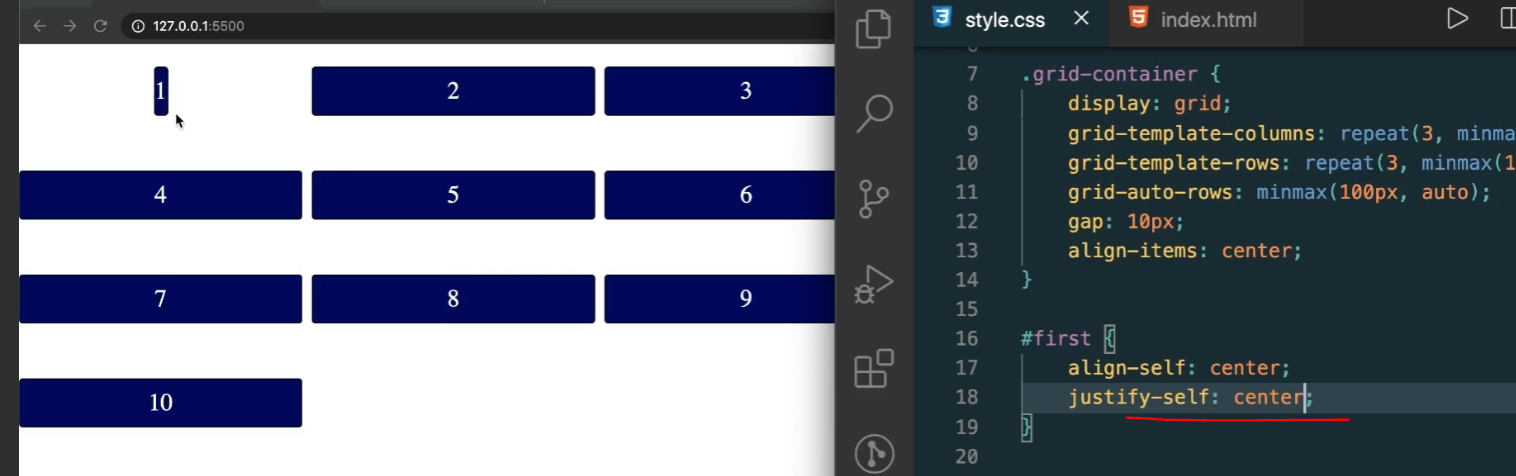
Start/center/end (by default stretch)

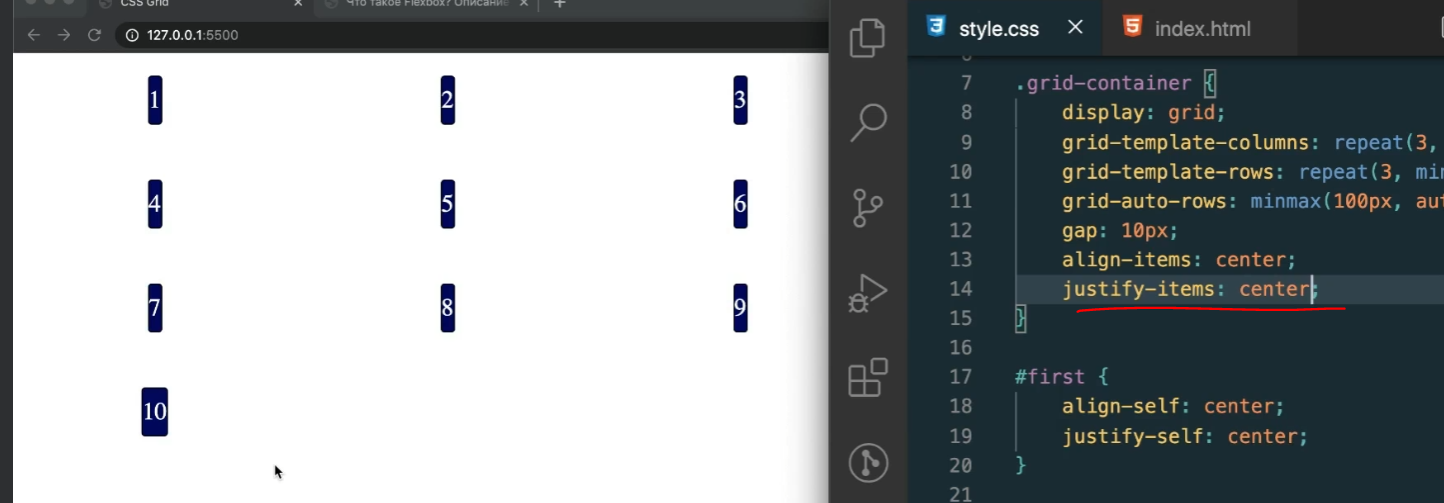
В гриде:

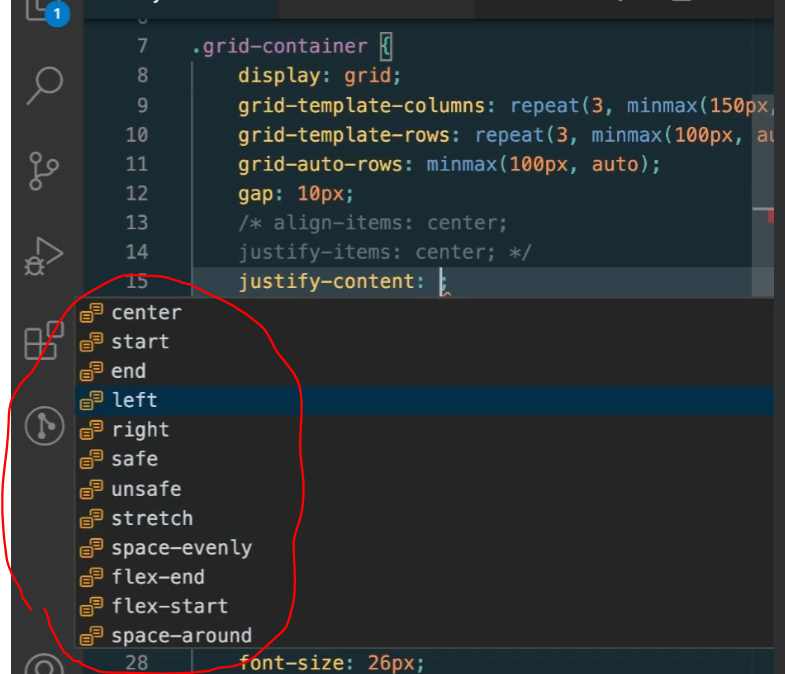
**Align-items**:

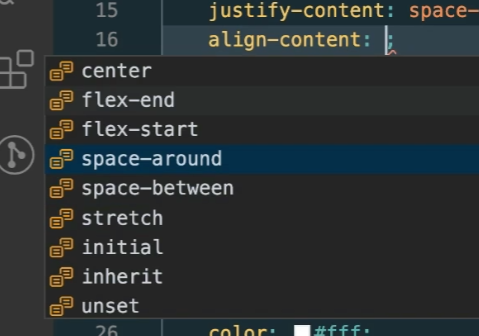












Комбинированное выравнивание

