



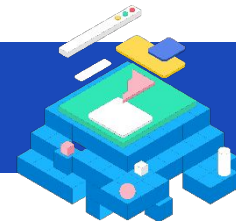
#ШПАРГАЛОЧКИ

СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

Материалы подготовлены отделом методической
разработки

Базовый уровень





Верстка сайта фотографа средствами CSS Grid layout





Повторяем grid

У контейнера нужно “включить” **grid** через свойство **display**, задать количество и размеры столбцов с помощью свойства **grid-template-columns**.

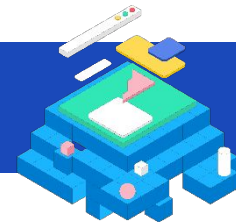
Высоту строк можно настроить с помощью **grid-template-rows**, а расстояние между строками и столбцами - свойством **grid-gap**.

Выровнять столбцы по горизонтали позволяет свойство **justify-content**.

Помимо контейнера, работать можно и с отдельными ячейками. Например, чтобы задать размер ячейки (сколько столбцов она займет) ей нужно указать свойства **grid-column-start** и **grid-column-end**.

А чтобы изменить строку ячейки, есть свойство **grid-row**.

Кроме того, в **grid** ячейкам можно задавать собственные имена с помощью свойства **grid-area**. А после в контейнере прописать свойство **grid-templates-area** и расположить именованные ячейки в нужном порядке.



Повторяем grid

У grid есть специальное свойство **grid-auto-rows**. Оно позволяет настроить высоту строк, размер которых не был задан явно. А если в качестве его значения указать функцию **minmax(*минимальная_высота*, auto)**, то можно решить проблему с переполнением элемента контентом (элемент просто растянется по высоте контента).

Еще одна полезная функция - **repeat(*количество_столбцов*, *размер*)**. Она упрощает работу с **grid-template-columns** (если размер столбцов одинаковый).



span

span - очень полезное значение свойства **grid-column**. Его
МОЖНО ИСПОЛЬЗОВАТЬ в качестве **grid-column-start**:

grid-column: span 2;



Здесь мы указали, что ячейка должна начинаться с того
столбца, в котором она уже стоит. А второе значение (число 2)
- это **количество столбцов**, которое ячейка должна занять.