html academy ■ Практикум ▼ 😚 Профессии ▼ У Курсы ▼ У Моё обучение ▼

+

Построение сеток на гридах

📴 Методика построения сеток на

🕑 Шаг 1. Создание общего контейнера для

🕑 Шаг 2. Строим сетку для блоков первого

🕑 Шаг 3. Выстраиваем отдельные разделы и

🕒 Шаг 4. Добавляем выравнивание

размещение и заполнение

Кейс 1, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

№ Кейс 5, средний уровень

🖸 Кейс 6, средний уровень

🖸 Кейс 7, средний уровень

🗔 Кейс 8, средний уровень

🖸 Кейс 9, сложный уровень

Кейс 10, сложный уровень

🕠 Кейс 2, лёгкий уровень

В Шаг 5*. Используем автоматическое

по макету

Выполнен на 0%

В Введение

Теория

гридах

сетки

уровня

блоки

🚱 Виталий Ризун 🔻

Шаг 5*. Используем автоматическое

Главная / Моё обучение / Построение сеток на гридах по макету / Методика построения сеток на гридах /

размещение и заполнение

Автоматическое размещение

```
На практике автоматическое размещение или свойство grid-auto-flow может быть полезным, когда страница
```

заполняется на основе данных, которые приходят с сервера, причём эти данные при отображении разного размера и мы не управляем их порядком. К примеру, у нас есть каталог товаров. Часть товаров нужно выделить, допустим, показать в карточке большего размера. Разметка каталога выглядит следующим образом:

```
<section class="card">
    <h2>Рюкзак BeastCraft</h2>
    1500₽
    <img src="./img/pic-23.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак походный 35л</h2>
    2000₽
    <img src="./img/pic-18.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Сумка спортивная</h2>
    1700₽
    <img src="/methodology/step-5/img/pic-53.jpg" width="552" height="670" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак Fjallraven Kanken</h2>
    4200₽
    <img src="/methodology/step-5/img/pic-21.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак Langly</h2>
    3900₽
    <img src="/methodology/step-5/img/pic-33.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак Haupes</h2>
    3300₽
    <img src="/methodology/step-5/img/pic-28.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Сумка кожаная</h2>
    5000P
    <img src="/methodology/step-5/img/pic-29.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак Vinta</h2>
    2900₽
    <img src="/methodology/step-5/img/pic-81.jpg" width="243" height="270" alt="">
   </section>
 <section class="card">
    <h2>Рюкзак детский</h2>
    800P
    <img src="/methodology/step-5/img/pic-58.jpg" width="243" height="270" alt="">
```

С помощью автоматического заполнения мы создадим сетку, в которой будут размещаться карточки разного

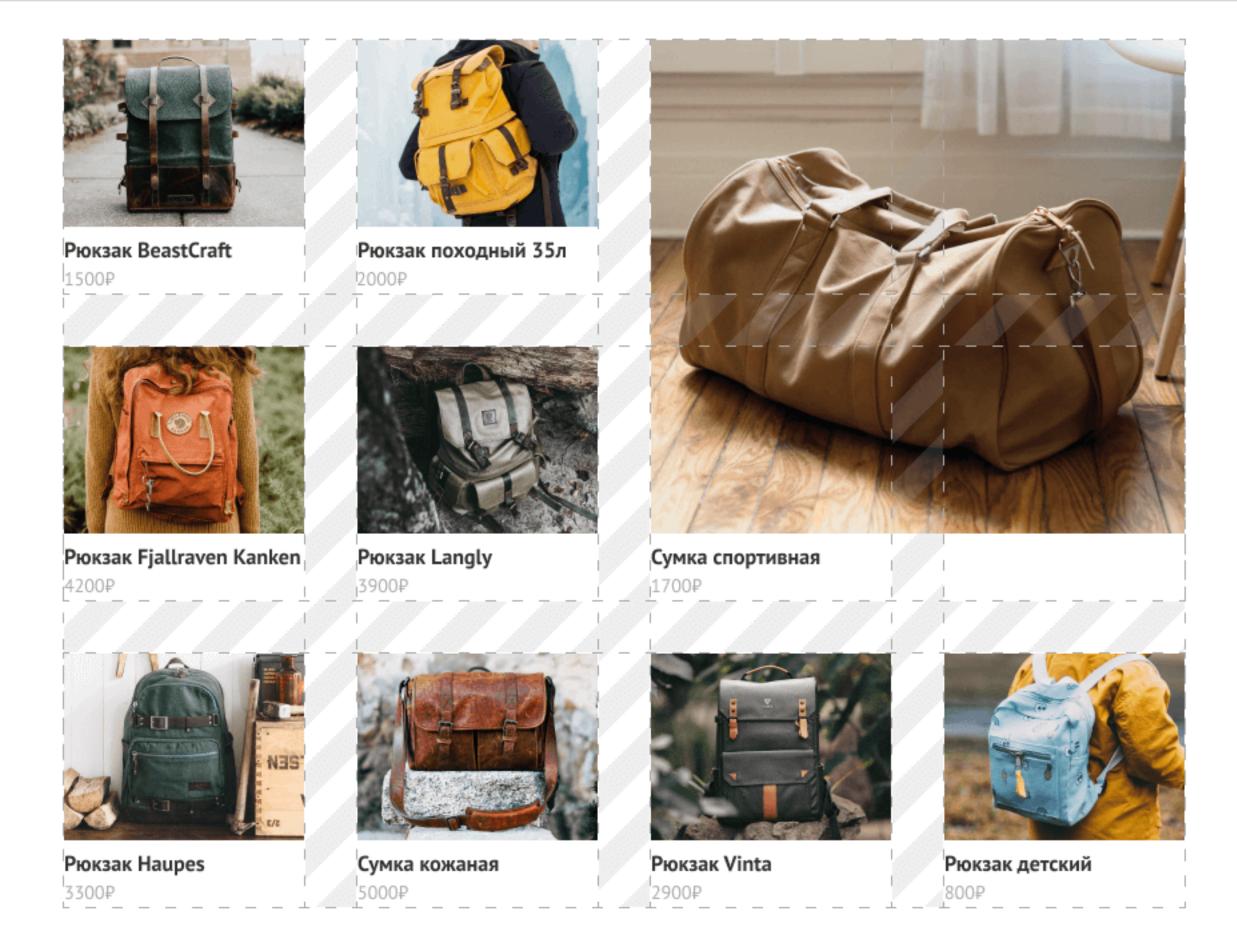
Карточку большего размера выделим классом catalog-item-big.

размера, не будет пустых ячеек и при этом в стилях добавиться лишь одно CSS-свойство. Браузер всё сделает за нас.

```
list-style: none;
   padding: 0;
   margin: 0;
   display: grid;
   grid-gap: 66px;
   grid-template-columns: repeat(4, 243px);
   grid-auto-flow: dense;
 .catalog-item-big {
   grid-column: span 2;
   grid-row: span 2;
Результат выглядит так:
```

</section>

.catalog-list {



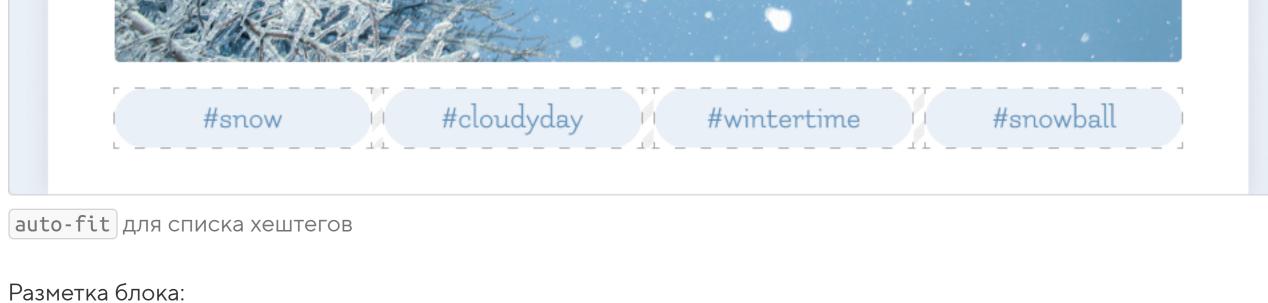
Автоматическое заполнение

grid-auto-flow: dense

Автоматическое заполнение чаще всего используется для создания адаптивных сеток. Но значение **auto-fit**

внешнего контейнера. Например, список хештегов для изображения:

можно использовать, если у нас заранее не известно количество элементов и нужно их растянуть по всей ширине



<section class="post">

```
<a href="#">#show</a>
   class="hashtag">
    <a href="#">#cloudyday</a>
   class="hashtag">
    <a href="#">#wintertime</a>
   <a href="#">#showball</a>
   </section>
Стили:
```

list-style: none; padding: 0; margin: 0;

.hashtag-list {

```
display: grid;
   grid-gap: 10px;
   grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
 .hashtag {
   padding: 10px 20px;
   text-align: center;
   background-color: #eeeeff;
Ещё один частый кейс, в котором можно применить значение auto-fit — это сетка для списка ссылок
на страницы в социальных сетях.
```

Продукты Компания Поддержка Соцсети

```
Наши решения
                                                    С чего начать?
                             О нас
                                                    Центр помощи
    Подписка
                             Культура
                                                    Сообщить об ошибке
    Триал-версии
                             Блог
auto-fit для списка ссылок на страницы в социальных сетях
Как правило, ссылки на страницы с социальных сетях отображаются в виде иконок одного размера. У CSS Grid
Layout в этом случае есть преимущество перед CSS Flex Layout, не нужно сбрасывать отступ у последнего
элемента (grid-gap). А за счёт auto-fit можно предусмотреть переполнение элементов, к примеру,
организация зарегистрировалась в новой соцсети и нужно добавить соответствующую ссылку. Единственное
```


ограничение, элемент со списком должен быть ограничен по ширине.

```
<img src="img/youtube.svg" width="32" height="32" alt="Найти нас в Youtube">
   </a>
 <a href="#">
    <img src="img/instagram.svg" width="32" height="32" alt="Найти нас в Instagram">
   </a>
 .social-links {
 display: grid;
 grid-template-columns: repeat(auto-fit, 32px);
 column-gap: 20px;
 margin: 0;
 padding: 0;
 list-style: none;
```


Добавим новые ссылки на Twitter и ВКонтакте.

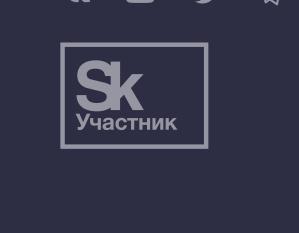
justify-self: end;

max-width: 500px;

```
<a href="#">
    <img src="img/youtube.svg" width="32" height="32" alt="Найти нас в Youtube">
   </a>
  <a href="#">
    <img src="img/instagram.svg" width="32" height="32" alt="Найти нас в Instagram">
   </a>
  <a href="#">
    <img src="img/twitter.svg" width="32" height="32" alt="Найти нас в Twitter">
   </a>
  <a href="#">
    <img src="img/vk.svg" width="32" height="32" alt="Найти нас в ВКонтакте">
   </a>
  Элементы заняли своё место в ряду ссылок, при этом наши стили не изменились.
                                             Соцсети
   Продукты
                Компания
                            Поддержка
```

(f) (D) (G) (T) Наши решения О нас С чего начать? Подписка

```
Культура
                                      Центр помощи
                                      Сообщить об ошибке
    Триал-версии
                      Блог
новые элементы в списке ссылок на страницы в социальных сетях
    Ознакомились со статьёй?
     Сохранить прогресс
```



Тренажёры Подписка Для команд и компаний Учебник по РНР Профессии Фронтенд-разработчик React-разработчик Фулстек-разработчик Бэкенд-разработчик Анимация для фронтендеров

Практикум

Услуги

Работа наставником

Для учителей

Стать автором

Node.js. Разработка серверов приложений и API

Вёрстка email-рассылок

Шаблонизаторы HTML

Анатомия CSS-каскада

Соглашение Конфиденциальность Сведения об образовательной организации Лицензия № 3026

© ООО «Интерактивные обучающие технологии», 2013-2022 **⟨/⟩ → ♥ → !!!**

Vue.js для опытных разработчиков

Алгоритмы и структуры данных

Регулярные выражения для фронтендеров

Курсы

🕻 Шаг 4. Добавляем выравнивание

HTML и CSS. Профессиональная вёрстка сайтов HTML и CSS. Адаптивная вёрстка и автоматизация JavaScript. Профессиональная разработка веб-интерфейсов JavaScript. Архитектура клиентских приложений React. Разработка сложных клиентских приложений РНР. Профессиональная веб-разработка PHP и Yii. Архитектура сложных веб-сервисов

Отчеты о курсах Информация Об Академии О центре карьеры Остальное Написать нам

Блог

С чего начать

Шпаргалки для разработчиков

Кейс 1, лёгкий уровень 🗦

Мероприятия Форум