

Построение сеток на гридах по макету

Выполнен на 0%

Введение

Теория

Методика построения сеток на гридах

- Шаг 1. Создание общего контейнера для сетки
- Шаг 2. Строим сетку для блоков первого уровня
- Шаг 3. Выстраиваем отдельные разделы и блоки
- Шаг 4. Добавляем выравнивание

Шаг 5*. Используем автоматическое размещение и заполнение

Кейс 1, лёгкий уровень

Кейс 2, 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

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

Шаг 5*. Используем автоматическое 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:

```
<ul class="catalog-list">
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак BeastCraft</h2>
      <p>1500P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак походный 35л</h2>
      <p>2000P</p>
      
    </section>
  </li>
  <li class="catalog-item catalog-item-big">
    <section class="card">
      <h2>Сумка спортивная</h2>
      <p>1700P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак Fjallraven Kanken</h2>
      <p>4200P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак Langly</h2>
      <p>3900P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак Haupes</h2>
      <p>3300P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Сумка кожаная</h2>
      <p>5000P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак Vinta</h2>
      <p>2900P</p>
      
    </section>
  </li>
  <li class="catalog-item">
    <section class="card">
      <h2>Рюкзак детский</h2>
      <p>800P</p>
      
    </section>
  </li>
</ul>
```

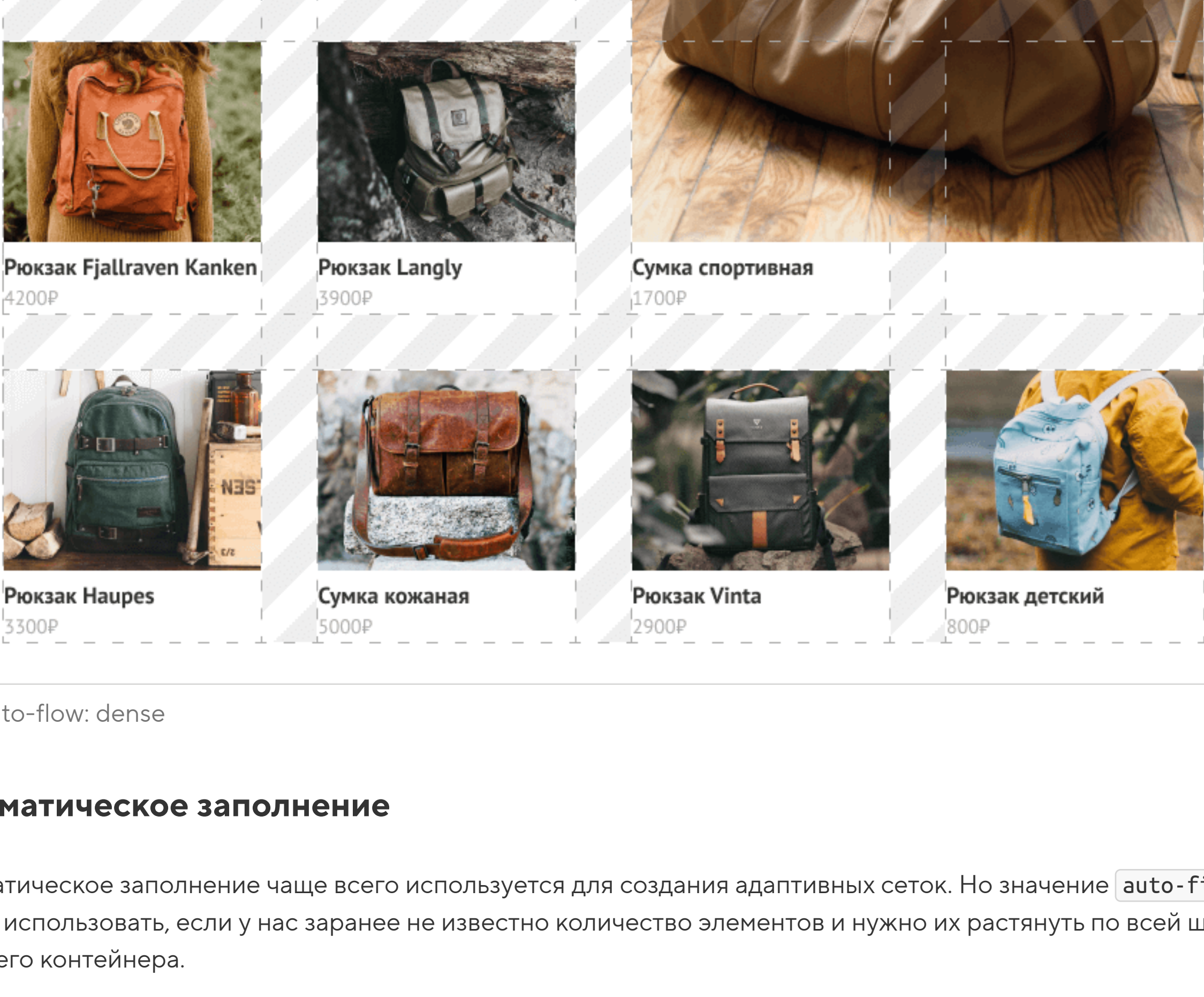
Карточку большего размера выделим классом `catalog-item-big`.

С помощью автоматического заполнения мы создадим сетку, в которой будут размещаться карточки разного размера, не будет пустых ячеек и при этом в стилях добавиться лишь одно CSS-свойство. Браузер всё сделает за нас.

```
.catalog-list {
  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:



`grid-auto-flow: dense`

Автоматическое заполнение

Автоматическое заполнение чаще всего используется для создания адаптивных сеток. Но значение `auto-fit` можно использовать, если у нас заранее не известно количество элементов и нужно их растянуть по всей ширине внешнего контейнера.

Например, список хештегов для изображения:



`auto-fit` для списка хештегов

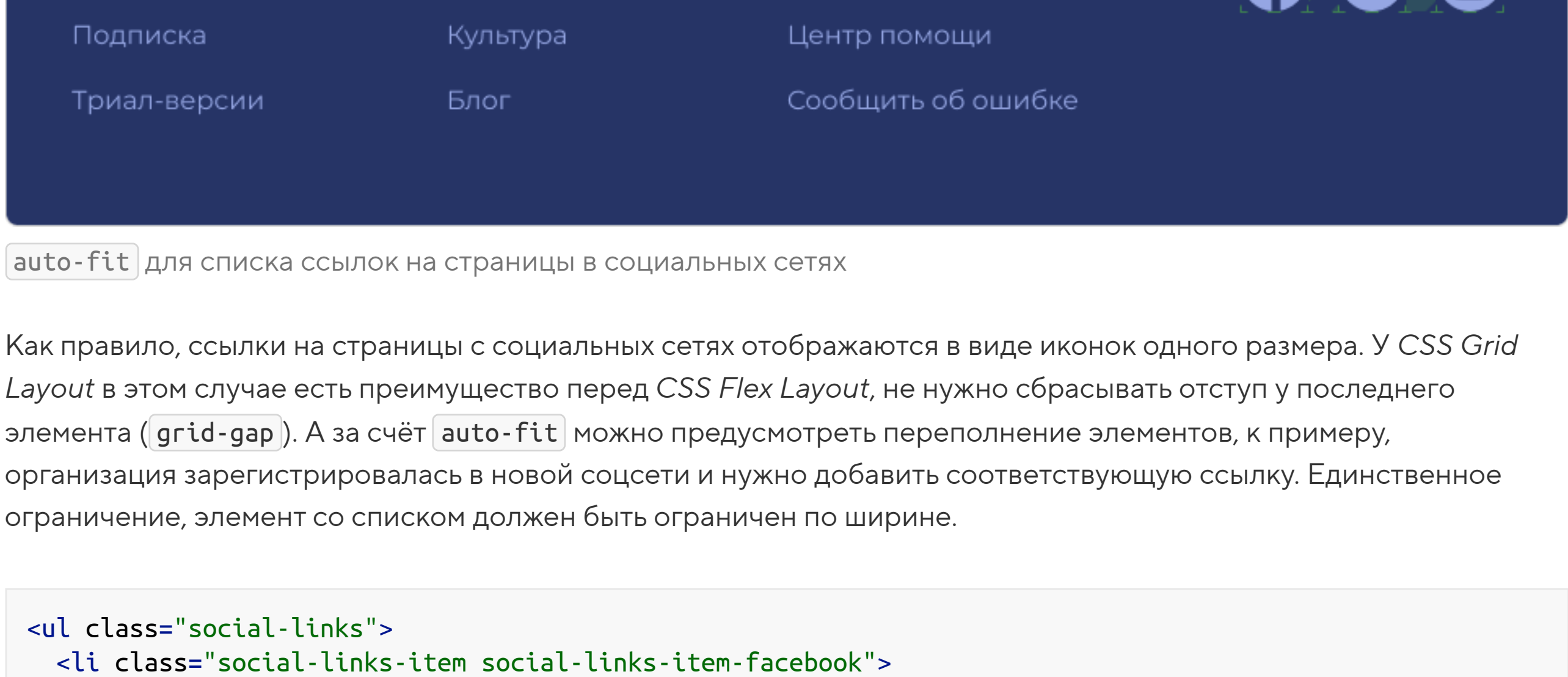
Разметка блока:

```
<section class="post">
  
  <ul class="hashtag-list">
    <li class="hashtag">
      <a href="#">#show</a>
    </li>
    <li class="hashtag">
      <a href="#">#cloudyday</a>
    </li>
    <li class="hashtag">
      <a href="#">#wintertime</a>
    </li>
    <li class="hashtag">
      <a href="#">#snowball</a>
    </li>
  </ul>
  ...
</section>
```

```
.hashtag-list {
  list-style: none;
  padding: 0;
  margin: 0;
  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.



`auto-fit` для списка ссылок на страницы в социальных сетях

Как правило, ссылки на страницы с социальных сетей 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.

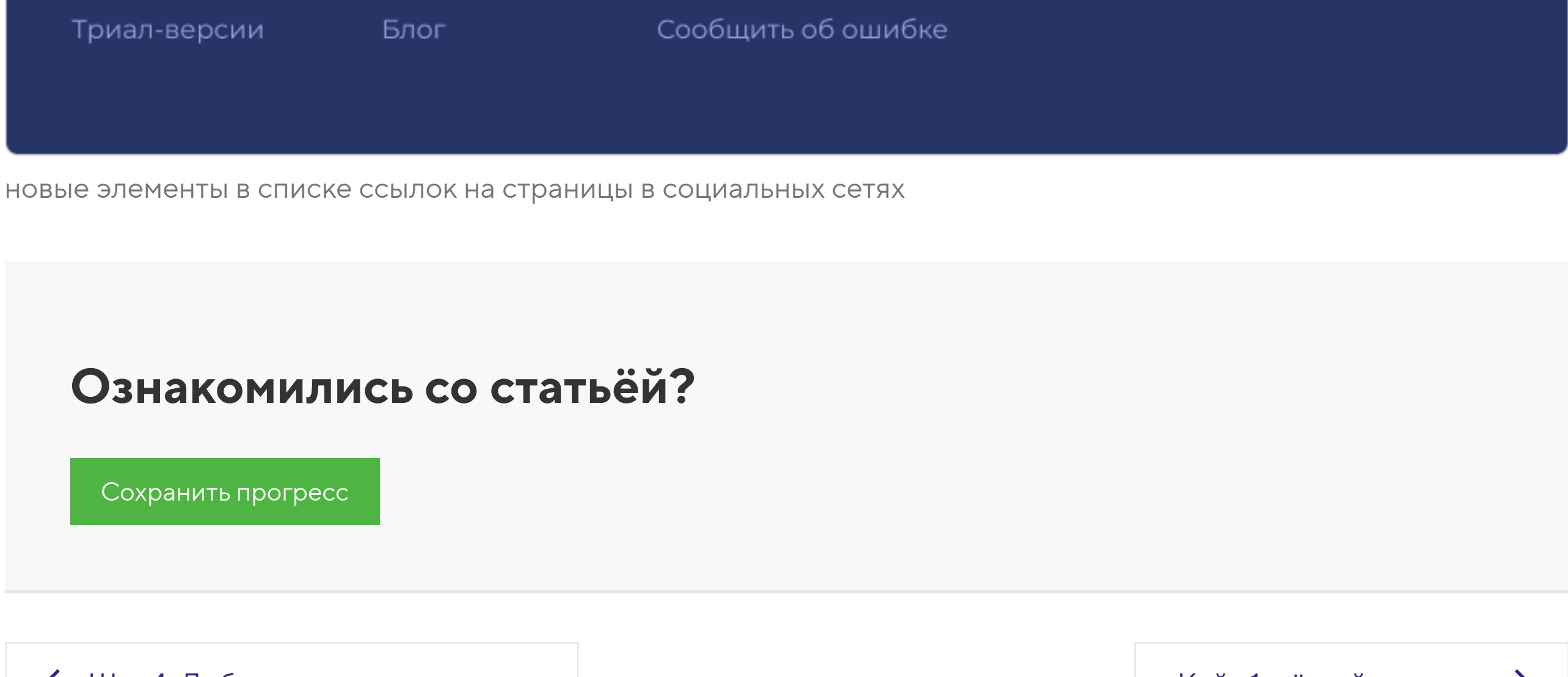
```
<ul class="social-links">
  <li class="social-links-item social-links-item-facebook">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-youtube">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-instagram">
    <a href="#">
      
    </a>
  </li>
</ul>
```

```
.social-links {
  display: grid;
  grid-template-columns: repeat(auto-fit, 32px);
  column-gap: 20px;
  margin: 0;
  padding: 0;
  list-style: none;
  justify-self: end;
  max-width: 500px;
}
...
```

Добавим новые ссылки на *Twitter* и *ВКонтакте*.

```
<ul class="social-links">
  <li class="social-links-item social-links-item-facebook">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-youtube">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-instagram">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-twitter">
    <a href="#">
      
    </a>
  </li>
  <li class="social-links-item social-links-item-vk">
    <a href="#">
      
    </a>
  </li>
</ul>
```

Элементы заняли своё место в ряду ссылок, при этом наши стили не изменились.



новые элементы в списке ссылок на страницы в социальных сетях

Ознакомьтесь со статьёй?

Сохранить прогресс

← Шаг 4. Добавляем выравнивание

Кейс 1, лёгкий уровень →