Введение

🖪 Общая теория

📴 Углублённая теория

В Методика работы с DOM

🕠 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, сложный уровень

☑ Кейс 9, сложный уровень

☑ Кейс 10, сложный уровень

В Изменение свойств элементов страницы

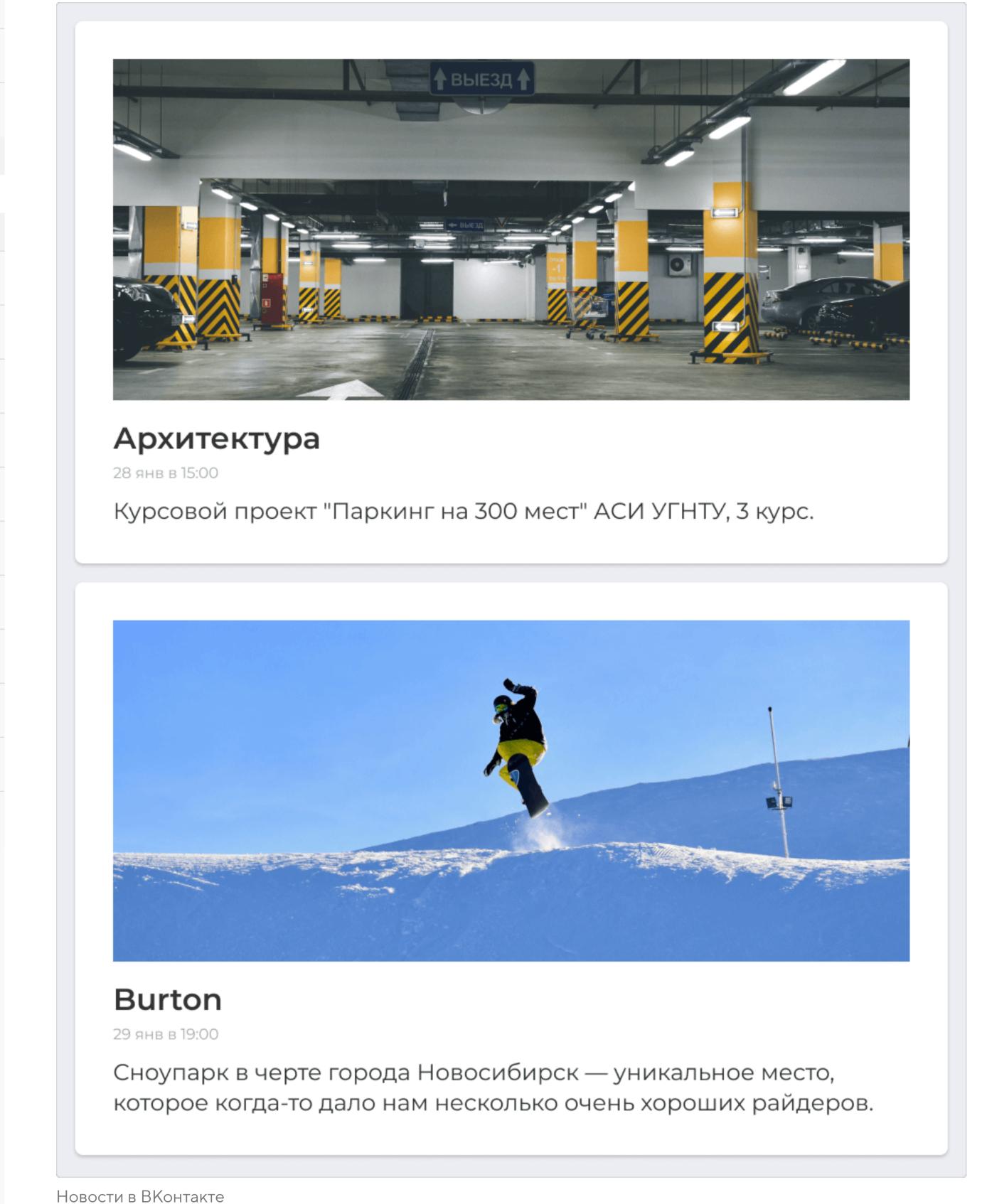
Добавление новых элементов в DOM

Рефакторинг и оптимизация кода

🚱 Виталий Ризун 🔻

Главная / Моё обучение / Работа с DOM в JavaScript / Методика работы с DOM /

Добавиться может группа однотипных элементов, например, новости в ВКонтакте.



Разберём эту ситуацию подробнее. Новые посты появляются ниже уже существующих (бесконечная загрузка и вставка после). Как это работает в упрощённом варианте.

const posts = [

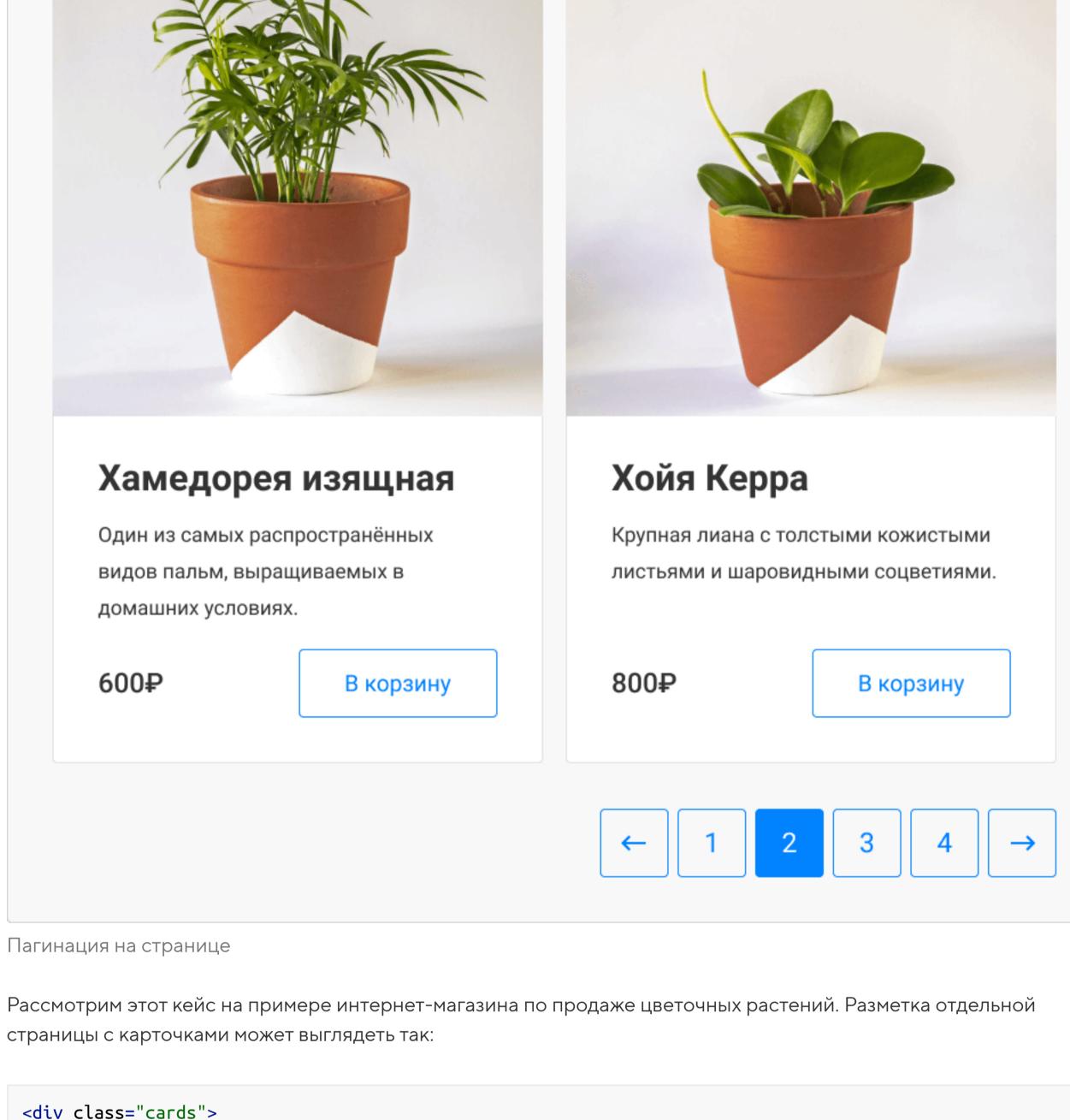
title: '"Мастера"',

У нас есть контейнер с записями **<div class="posts">**. В нём содержатся блоки с отдельными новостями <section class="post">. Опишем разметку блоков:

<div class="posts"> <section class="post"> <header class="post-header"> <h2>Apхитектура</h2> <time datetime="2021-01-28 15:00">28 янв в 15:00</time>

```
</header>
     <div class="post-content">
       <р>Курсовой проект "Паркинг на 300 мест" АСИ УГНТУ, 3 курс</р>
     </div>
   </section>
   <section class="post">
     <header class="post-header">
       <h2>Burton</h2>
       <img src="img/pic-25.jpg" width="200" height="150" alt="">
       <time datetime="2021-01-29 19:00">29 янв в 19:00</time>
     </header>
     <div class="post-content">
       <р>Сноупарк в черте города Новосибирск - уникальное место, которое когда-то дало нам несколько
 очень хороших райдеров.
     </div>
   </section>
 </div>
Сначала с помощью шаблонных строки и интерполяции сформируем HTML для элементов. А затем для
добавления их в DOM используем метод insertAdjacentHTML.
```

```
date: '2021-01-29 20:00',
     img: 'img/pic-21.jpg',
     text: 'Castella Shop - это единственная в Санкт-Петербурге пекарня, которая специализируется на
 приготвлении кастеллы - нежнейшего хлопкового японского бисквита',
     title: 'Секрет Фирмы',
     date: '2021-01-29 21:00',
     img: 'img/pic-25.jpg',
     text: 'Одного хештега в Twitter главы Tesla и SpaceX оказалось достаточно, чтобы инвесторы кинулись
 скупать криптовалюту',
 // находим контейнер с постами
 const postsContainer = document.querySelector('.posts');
 // формируем элементы с новыми постами
 // используем шаблонные строки и интерполяцию
 // склеиваем всё в одну строку
 const postsString = posts.map(({title, date, text}) => `
   <section class="post">
     <header class="post-header">
       <h2>${title}</h2>
       <img src="${img}" width="200" height="150" alt="">
       <time datetime="${date}">${date}</time>
     </header>
     <div class="post-content">
       ${text}
     </div>
   </section>
   `).join('');
 // добавляем посты в контейнер
 postsContainer.insertAdjacentHTML('beforeend', postsString);
Всем знакомая пагинация (или пейджинация) — это тоже создание новых элементов, которые замещают
существующие.
```



 <р>Один из самых распространённых видов пальм, выращиваемых в домашних условиях. 600 py6

const cards = [

title: 'Хамедорея изящная',

img: 'img/pic-34.jpg',

title: 'Хойя Керра',

img: 'img/pic-37.jpg',

price: '600',

price: '800',

<section class="card">

<h2>Хамедорея изящная</h2>

В корзину </section> <section class="card"> <h2>Xойя Керра</h2>

```
<img src="img/pic-37.jpg" width="100" height="100" alt="">
     <р>Крупная лиана с толстыми кожистыми листьями и шаровидными соцветиями.
     800 py6
     <a href="#">В корзину</a>
   </section>
 </div>
При переключении со страницы на страницу содержимое элемента <div class="cards"> очищается и взамен
добавляются данные для выбранной страницы.
Также как и в предыдущем случае, для создания элементов используем шаблонные строки и интерполяцию
и метод insertAdjacentHTML для добавления в DOM. Но предварительно будем очищать родительский элемент
от содержимого.
Напишем код, который реализует переключение между страницами.
```

description: 'Один из самых распространённых видов пальм, выращиваемых в домашних условиях.',

description: 'Крупная лиана с толстыми кожистыми листьями и шаровидными соцветиями.',

```
// находим родительский элемент с карточками
 const cardsContainer = document.querySelector('.cards');
 // очищаем содержимое <div class="cards">
 cardsContainer.innerHTML = '';
 // формируем HTML с новыми карточками
 // используем шаблонные строки и интерполяцию
 const cardsString = cards.map(({title, price, description}) => `
   <section class="card">
     <h2>${title}</h2>
     <img src="${img}" width="100" height="100" alt="">
     ${description}
     ${price}
     <a href="#">В корзину</a>
   </section>
   `).join('');
 // заполняем элемент <div class="cards"> новыми данными
 cardsContainer.insertAdjacentHTML('beforeend', cardsString);
Добавляться могут одиночные или разнотипные элементы. К примеру, в форме регистрации пользователь может
добавить верификацию по номеру телефона.
     Ваш E-mail:
                                                      Ваш E-mail:
       ivanov@mail.ru
                                                         ivanov@mail.ru
     Пароль:
                                                      Пароль:
       n ••••••
```

Верификация по номеру телефона Верификация по номеру телефона

```
Телефон:
             Зарегистрироваться
                                                              +7 (999) 999-99-99
                                                                Зарегистрироваться
Дополнительная верификация по номеру телефону
Поле с телефоном добавляем по нажатию на переключатель «Верификация по телефону».
 <form class="registration" action="#" method="POST">
   <label>
     Baш E-mail:
     <input type="email" name="email" placeholder="ivanov@mail.ru">
   </label>
   <label>
     Пароль:
     <input type="password" name="password" placeholder="password">
   </label>
   <label>
     Валидация по номеру телефона
     <input type="checkbox" class="more">
   </label>
   <div class="verification-container">
   <button type="submit">Зарегистрироваться</button>
```

```
</form>
 // находим родительский элемент для новых полей
 const verificationContainer = document.querySelector('.verification-container');
 // переключатель дополнительных полей
 const switchButton = document.querySelector('.more');
 switchButton.addEventListener('click', (evt) => {
   // предварительно почистим родительский элемент
   verificationContainer.innerHTML = '';
   if (switchButton.checked) {
     // используем шаблонные строки
     const fieldsString = `
     <label>
       Телефон:
       <input type="phone" name="phone" placeholder="+7(999)999-99-99">
     </label>`;
     // добавляем поля в родительский элемент
     verificationContainer.insertAdjacentHTML('beforeend', fieldsString);
 });
После анализа описанных кейсов мы можем выделить следующие шаги:
— Ищем нужный нам DOM-элемент на странице. Этот элемент будет родительским, целевым элементом или
```

— Создаём новый (ые) элемент (ы). Используем методы: document.createElement, тег <template>, createContextualFragment или HTML-строки. — Устанавливаем значение атрибутов, заполняем содержимое, если требуется. Используем методы:

требуется заменить прежнее содержимое элемента.

— Чистим прежнее содержимое найденного целевого элемента **innerHTML** = ''. Этот пункт выполняем, если

- setAttribute, textContent, innerHTML и другие. — Добавляем созданный (e) элемент (ы) в *DOM*. Используем элемент, который нашли в первом пункте как родительский, целевой элемент или точку отсчёта, и для этого элемента вызываем методы appendChild,
- append, insertAdjacentElement, insertBefore или insertAdjacentHTML. Выбор методов зависит от того, какая браузерная поддержка нужна, добавляется один или несколько элементов, а также есть ли пользовательский ввод.

Ознакомились со статьёй? Сохранить прогресс



Курсы Практикум HTML и CSS. Профессиональная вёрстка сайтов Тренажёры HTML и CSS. Адаптивная вёрстка и автоматизация Подписка Для команд и компаний JavaScript. Профессиональная разработка веб-интерфейсов Учебник по РНР Профессии

К Изменение свойств элементов страницы

точкой отсчёта.

JavaScript. Архитектура клиентских приложений React. Разработка сложных клиентских приложений Фронтенд-разработчик РНР. Профессиональная веб-разработка React-разработчик Фулстек-разработчик Бэкенд-разработчик

Шаблонизаторы HTML

Анатомия CSS-каскада

Соглашение Конфиденциальность Сведения об образовательной организации Лицензия № 3026

Алгоритмы и структуры данных

Шпаргалки для разработчиков Отчеты о курсах Информация

Рефакторинг и оптимизация кода 🗦

Блог

С чего начать

Об Академии

Остальное

Написать нам

Мероприятия

Форум

О центре карьеры