СОДЕРЖАНИЕ

ВВЕДЕНИЕ …………………………………………………………….... 3

1. Сookie ……………………………………………………………...……4

2. Назначения cookie ……………………………………………...……...6

ЗАКЛЮЧЕНИЕ …………………………………………………………..8

ПРАКТИЧЕСКАЯ ЧАСТЬ ………………………………….....………....9

ВВЕДЕНИЕ

Протокол HTTP устроен так, что при каждом обновлении страницы браузер забывает все предыдущие действия на этом сайте. Cookies решают эту проблему.

Куки — это инструмент, который помогает сайтам «запоминать» и идентифицировать пользователей. Например, если пользователь случайно закрыл страницу сайта или браузер, после следующего входа его корзина в интернет-магазине и просмотренные товары сохранятся. Покупателям это удобно, ведь им не приходится искать выбранные товары заново.

Механизм закладок (cookie) является составной частью HTTP-протокола. Данный механизм позволяет производить сохранение небольших количеств информации, переданных сервером, на стороне клиента. При следующих запросах эта информация может пересылаться клиентом серверу, который может использовать ее в своих целях. В настоящее время cookie используются практически повсеместно. Типичными задачами, решаемыми при помощи установки cookie являются: автоматическая авторизация пользователя на web-ресурсе, поддержка сессий, и.т.д.

В большинстве случаев cookie устанавливаются сервером и без изменений передаются серверу агентом пользователя при последующих запросах. Однако, доступ к cookie и управление cookie на стороне браузера из клиентских сценариев часто оказывается весьма полезным. Все современные браузеры имеют поддержку доступа к cookie из скриптов. В этой статье мы рассмотрим общие вопросы работы с cookie из сценариев, написанных на JavaScript.

1. Cookie

Сookie являются составной частью протокола HTTP. От сервера клиенту cookie передаются в заголовке HTTP-ответа сервера. Точнее - в поле Set-Cookie этого заголовка. Поле Set-Cookie может содержать данные нескольких кук. Также в заголовке HTTP-ответа может содержаться несколько полей Set-Cookie. Далее приведен типичный пример одного поля Set-Cookie заголовка HTTP-ответа сервера.

Строка Set-Cookie заголовка ответа сервера состоит из последовательности подстрок, разделенных символами ";" (точка с запятой). Вначале следуют одна или несколько пар <name>=<value>. Каждая из этих пар соответствует одной куке с именем name и значением value. Потом следуют атрибуты, с которыми устанавливаются куки, содержащиеся в данном поле Set-Cookie. Далее приведено описание значений атрибутов cookie.

Атрибут expires. Задается в формате: expires=<date>. Определяет дату и время окончания действия устанавливаемых данным полем Set-Cookie заголовка HTTP-ответа сервера cookie. Значение <date> должно являться строкой, задающей время по гринвичскому меридиану (GMT) в формате "DAY, DD-MMM-YYYY HH:MM:SS GMT" (например "Tue, 28 Feb 2006 12:41:04 GMT"). После истечения срока действия cookie удаляются браузером. Если атрибут expires не задан (время действия cookie не установлено), cookie удаляются при окончании работы браузера. Такие cookie называют "сессионными" (действуют только до окончания текущей сессии).

* атрибут domain. Задается в формате domain=<domain\_name>. Определяет окончание множества имен доменов, в которых действительны устанавливаемые cookie. К примеру, если при установке cookie, значение <domain\_name> равно .codeguru.ru, установленные cookie будут отсылаться серверу при запросе документов с сайтов, расположенных на доменах web.codeguru.ru, forum.codeguru.ru, и.т.д. При этом дополнительным условием отсылки cookie будет являться проверка значения атрибута path. Атрибут domain весьма полезен. Он может использоваться, например, для организации единой системы аутентификации пользователя для различных web-сервисов, расположенных на различных поддоменах. Например, на основном сайте и на форуме сайта. Во избежание неконтролируемой отсылки cookie, значение <domain\_name> не должно содержать только имя зоны (например, .com, .net.ru, и.т.д.);
* атрибут path. Задается в формате path=<some\_path>. Определяет множество URL, для которых действительны устанавливаемые cookie. Например, если значение <some\_path> при установке cookie равно /temp, то установленные cookie будут отсылаться при запросе документов, включая пути /temp000 и /temp/temp.htm. Если значение path не определено, в качестве него принимается путь к ресурсу, при запросе которого были установлены cookie;
* атрибут secure. Это атрибут логического типа. Указание атрибута secure в поле Set-Cookie заголовка HTTP-ответа сервера делает устанавливаемые cookie защищенными. Такие cookie должны отсылаться на сервер только по протоколу https. HTTPS - это, практически, тот же протокол http, только данные между клиентом и сервером передаются через защищенное (данные шифруются) SSL (Secure Socket Layer) соединение;
* атрибут HttpOnly. Это атрибут логического типа. Указание атрибута HttpOnly в поле Set-Cookie заголовка HTTP-ответа сервера делает устанавливаемые cookie не доступными клиентским скриптам (это применяется из соображений безопасности).

1. Назначения cookie

Насколько хорошо сайты знают вас по информации из куки? Это зависит от конкретного интернет-ресурса.

С помощью cookies сайт может запомнить любую информацию о сеансе и использовать её при следующем визите пользователя.

Cookies хранятся на устройстве пользователя и содержат данные о посещённых ресурсах. Их используют для авторизации, хранения индивидуальных настроек для каждого пользователя, сбора статистики и отслеживания активности.

Основные задачи, для которых применяют cookies:

* Сохранение личных данных на протяжении одной сессии (имя, логин, email, пароль). При отсутствии cookie пользователю пришлось бы авторизоваться на сайте после каждого обновления страницы.
* Настройка профиля (язык, геолокация, включенные функции). Если пользователь настроил содержимое страницы по личным предпочтениям, то при повторном посещении эти настройки сохранятся.
* Фиксация действий пользователей (реакции, активности, просмотренные товары). Собранную информацию используют разные сервисы и интернет-магазины. Например, чтобы запомнить ответ при голосовании, собрать статистику.
* Идентификация пользователей (тип используемого устройства, время посещения, количество просмотренных страниц). С помощью этих cookie сайты собирают необходимые данные о поведении посетителей. В большинстве случаев cookie позволяют получить практически полную карту «путешествий» пользователя по сайту. Если же говорить о куки-файлах для отдельных рекламных баннеров, то они могут собрать информацию о вас, даже если вы посещали разные домены, на которых имелись баннеры от одного разработчика.

Каждая текстовая запись (кука) содержит:

* имя файла;
* сохранённые данные;
* период хранения;
* домен сервера, создавшего файл;
* сведения о потребности доступа к записям с помощью HTTP-соединения;
* место расположения документов, которым разрешён доступ к записям.

Куки — это в большинстве своём персональные данные. Потому сайты обязаны запрашивать у пользователей согласие на использование cookie. На территории Евросоюза этого требует GDPR, в России — ФЗ «О персональных данных».

В российском законодательстве персональные данные определяют, как «любую информацию, прямо или косвенно относящуюся к определяемому лицу». То есть конкретно о cookie ничего не сказано, но по факту они считаются персональной информацией. Если докажут, что сайт без спроса обрабатывал данные посетителей, то владельцу ресурса грозит крупный штраф или даже блокировка. Например, в 2016 году LinkedIn заблокировали именно за несогласованное использование данных пользователей.

Для соблюдения законодательных требований на сайте, использующем cookie, необходимо опубликовать «Политику конфиденциальности» или в европейском варианте — «Соглашение о приватности» (privacy agreement). В этом документе должна быть указана информация о том:

* какие пользовательские данные собирает сайт;
* как эти данные будут храниться и применяться;
* каковы цели обработки собранной информации;
* в каких ситуациях данные передают третьим лицам; как можно изменить или удалить свои данные.

ЗАКЛЮЧЕНИЕ

Появление cookie-файлов упростило работу в интернете. Благодаря им легче проходить авторизацию на сайтах, оплачивать покупки и быстро обращаться к нужным ресурсам. При этом вы контролируете большинство процессов, ведь cookie-файлы несложно удалить. Но не забывайте: обеспечить безопасность устройства – главная задача любого пользователя.

Файлы куки — удобный и полезный инструмент, который отлично подходит для настройки рекламы и маркетингового анализа. Простому юзеру он экономит время и килобайты памяти в его мозге — благодаря куки не нужно держать в голове кучу данных аутентификации.

Если вы не доверяете куки, то можете хранить логины и другие данные авторизации в специальной программе или обычном блокноте. Есть и другой вариант — время от времени заботиться о компьютере. Периодически устраивайте генеральную уборку и стряхивайте крошки от печенек с жесткого диска.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Было создано 3 html страницы, связанные между собой ссылками.

Отрисовка ссылок была реализована с помощью js файла import-nav.js который создает на каждой страничке шапку сайта реализация представлена ниже.

const link = document.querySelector("[data-import='nav']");

document.body.querySelector("header").innerHTML = `

<nav>

<div class="container">

<a href="./index.html" class="link">Главная</a>

<a href="./news.html" class="link">Новости</a>

<a href="./register.html" class="link">Регистрация</a>

</div>

</nav>`;

На рисунке 1 изображена шапка сайта



Рисунок 1 – Шапка сайта

HTML код главной страницы представлен ниже

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Главная</title>

<link href="../css/main.css" rel="stylesheet" />

</head>

<body> <header></header> <main></main>

<script src="../js/import-nav.js"></script>

<script src="../js/index.js"></script>

</body>

</html>

По коду видно, что подключение внешних стилей происходит через link к внешнему файл css код представлен ниже.

\* {

margin: 0;

padding: 0;

}

.news-block {

display: flex;

background-color: #dbdbdb;

padding: 10px;

flex-direction: column;

margin-bottom: 15px;

}

.news-title {

font-size: 20px;

margin: 0 0 10px 5px;

}

.news-text {

margin: 0;

text-indent: 30px;

}

.container {

max-width: 1170px;

padding: 0 15px;

margin: 0 auto;

}

nav {padding: 15px 0;}

.form-container {margin-bottom: 10px;}

Также в главной странице подключает index.js скрипт код которого представлен ниже.

const queryString = location.search

const params = new URLSearchParams(queryString)

const name = params.get("name");

const checkBxodOne = params.get("checkBxodOne");

const pol = params.get("pol");

const comment = params.get("comment");

const about = params.get("about");

function createDiv(text, val) {

const div = document.createElement("div");

div.innerHTML = `${text}: <b>${val}<b>`;

document.querySelector("main").append(div);

}

if (name) { createDiv("Ваше имя", name); }

if (checkBxodOne) { createDiv("Вы зашли в 1 раз", checkBxodOne); }

if (pol) { createDiv("Ваш пол", pol); }

if (comment) {createDiv("Ваше комментарии", comment); }

if (about) { createDiv("Вы узнали о нас", about); }

Страница новости представлена на рисунке 2.

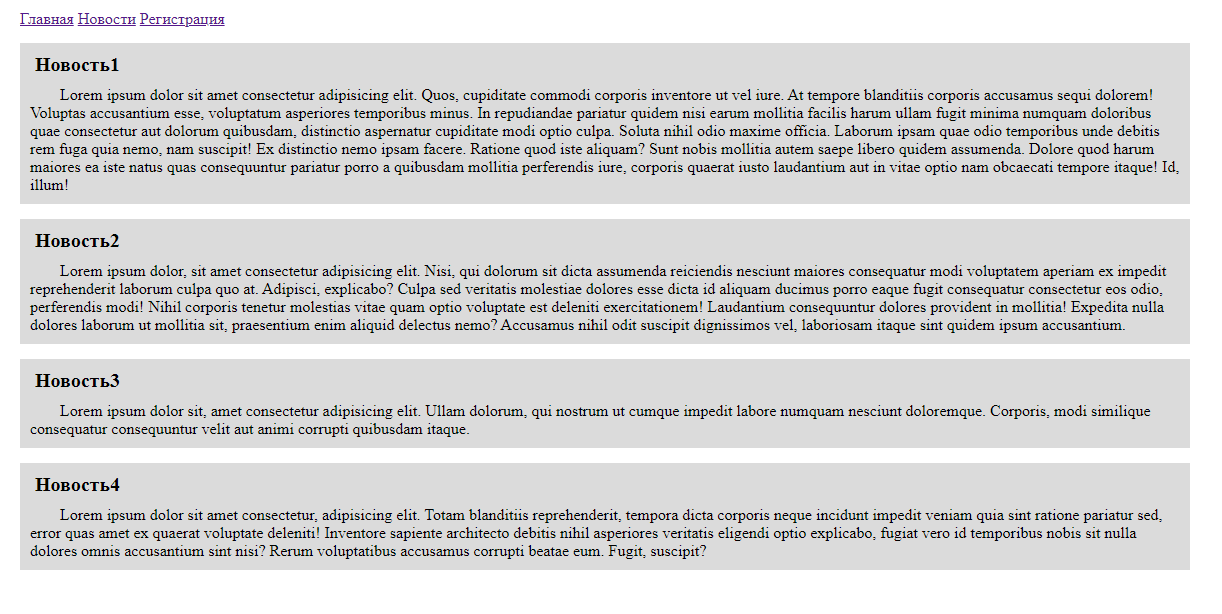


Рисунок 2 – Страница новости

Код страницы новости представлен ниже.

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Новости</title>

<link href="../css/main.css" rel="stylesheet" />

</head>

<body>

<header></header>

<main>

<div class="container">

<div class="news-container">

<div class="news-block">

<h3 class="news-title">Новость1</h3>

<p class="news-text">Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos, cupiditate commodi corporis inventore ut vel iure. At tempore blanditiis corporis accusamus sequi dolorem! Voluptas accusantium esse, voluptatum asperiores temporibus minus. In repudiandae pariatur quidem nisi earum mollitia facilis harum ullam fugit minima numquam doloribus quae consectetur aut dolorum quibusdam, distinctio aspernatur cupiditate modi optio culpa. Soluta nihil odio maxime officia. Laborum ipsam quae odio temporibus unde debitis rem fuga quia nemo, nam suscipit! Ex distinctio nemo ipsam facere. Ratione quod iste aliquam? Sunt nobis mollitia autem saepe libero quidem assumenda.Dolore quod harum maiores ea iste natus quas consequuntur pariatur porro a quibusdam mollitia perferendis iure, corporis quaerat iusto laudantium aut in vitae optio nam obcaecati tempore itaque! Id, illum!</p> </div>

<div class="news-block">

<h3 class="news-title">Новость2</h3>

<p class="news-text">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nisi, qui dolorum sit dicta assumenda reiciendis nesciunt maiores consequatur modi voluptatem aperiam ex impedit reprehenderit laborum culpa quo at. Adipisci, explicabo?

Culpa sed veritatis molestiae dolores esse dicta id aliquam ducimus porro eaque fugit consequatur consectetur eos odio, perferendis modi! Nihil corporis tenetur molestias vitae quam optio voluptate est deleniti exercitationem! Laudantium consequuntur dolores provident in mollitia! Expedita nulla dolores laborum ut mollitia sit, praesentium enim aliquid delectus nemo? Accusamus nihil odit suscipit dignissimos vel, laboriosam itaque sint quidem ipsum accusantium.</p></div>

<div class="news-block">

<h3 class="news-title">Новость3</h3>

<p class="news-text">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ullam dolorum, qui nostrum ut cumque impedit labore numquam nesciunt doloremque. Corporis, modi similique consequatur consequuntur velit aut animi corrupti quibusdam itaque.</p> </div>

<div class="news-block">

<h3 class="news-title">Новость4</h3>

<p class="news-text">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Totam blanditiis reprehenderit, tempora dicta corporis neque incidunt impedit veniam quia sint ratione pariatur sed, error quas amet ex quaerat voluptate deleniti! Inventore sapiente architecto debitis nihil asperiores veritatis eligendi optio explicabo, fugiat vero id temporibus nobis sit nulla dolores omnis accusantium sint nisi? Rerum voluptatibus accusamus corrupti beatae eum. Fugit, suscipit?</p> </div> </div></div>

</main>

<script src="../js/import-nav.js"></script>

</body>

</html>

Страница регистрация представлена на рисунке 3.

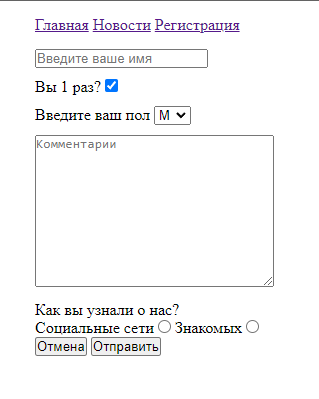


Рисунок 3 – страница регистрация

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Новости</title>

<link href="../css/main.css" rel="stylesheet" />

</head>

<body>

<header></header>

<main>

<div class="container"> <form action="" name="register">

<div class="form-container"><input name="name" type="text" placeholder="Введите ваше имя"></div> <div class="form-container"><label for=""> Вы 1 раз?<input name="checkBxodOne" type="checkbox" checked></label></div><div class="form-container">

<label for="">Введите ваш пол

<select name="pol" placeholder="Введите ваш пол"><option value="М">М</option><option value="Ж">Ж</option> </select>

</label> </div>

<div class="form-container"><textarea name="comment" placeholder="Комментарии" name="" id="" cols="30" rows="10"></textarea></div>

<div class="form-container">

<label for="">Как вы узнали о нас?</label>

<div class="container-radio">

<label for="">Социальные сети<input name="about" value="Социальные сети" type="radio" /></label>

<label for="">Знакомых<input name="about" value="Знакомых" type="radio" /></label></div>

<button class="cancel">Отмена</button>

<button class="register">Отправить</button>

</div>

</form>

</div>

</main>

<script src="../js/import-nav.js"></script>

<script src="../js/register.js"></script>

</body>

</html>

Обработка этой страницы реализована в файле register.js представлена ниже.

const buttonOk = document.querySelector("button.register");

const buttonCancel = document.querySelector("button.cancel");

const form = document.querySelector("form[name=register]");

if (buttonOk) {

buttonOk.onclick = (e) => {

let params = "";

e.preventDefault();

const formData = new FormData(form);

if (formData) {

for (const data of formData.entries()) {

if (data[1]) {

if (data[0] == "checkBxodOne") {

const val = data[1] == "one" ? true : false;

params += `${data[0]}=${val}&`

continue;}

params += `${data[0]}=${data[1]}&`}}

}

window.location.replace(`./index.html?${params}`)}

}

if (buttonCancel) {

buttonCancel.onclick = (e) => {

e.preventDefault();

form.reset();

}

}

После заполнение формы регистрация происходит переход на страницу главную с введенными данными результат представлен на рисунке 4.

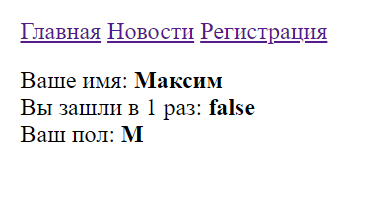


Рисунок 4 – главная страница после заполнения формы регистрации.