**Практическая работа №3**

**Мартинович Екатерина, Яровая Екатерина, ПИ-231**

Любая веб-страница имеет следующую структуру:

<!DOCTYPE html> – объявление типа документа.

<html> – корневой элемент документа.

<head> – содержит метаинформацию о документе, которая не отображается напрямую на странице.

<body> – содержит всё содержимое страницы, отображаемое пользователю.

Performance (Производительность) проверяет скорость загрузки страницы, оптимизацию изображений и медиафайлов, использование кэширования.

Accessibility (Доступность) проверяет контрастность текста и фона, наличие альтернативного текста у изображений, корректное использование семантических тегов, возможность навигации с клавиатуры и скринридеров.

Best Practices (Лучшие практики) проверяет использование безопасных соединений, отсутствие небезопасных или устаревших библиотек, корректную работу JavaScript, поддержку адаптивности и корректность работы на мобильных устройствах.

SEO (Поисковая оптимизация) проверяет наличие и корректность мета-тегов, правильную структуру заголовков, читаемость и доступность ссылок.

1. <https://www.chess.com/home>

1) <html>: Служит контейнером для всего документа. Содержит атрибуты, определяющие язык (lang=«ru») и цветовую тему сайта (data-color-mode=«dark»). Внутри него размещаются теги <head> и <body>.

<head>: Содержит метаинформацию и инструкции для браузера. Здесь находятся метатеги – кодировка (charset=«utf-8»), настройки viewport для адаптивности, инструкции для поисковых роботов, верификационные ключи; предзагрузка критических ресурсов: шрифты ChessGlyph (для символов фигур) и Chess Sans; подключение большого количества CSS-стилей для оформления всех компонентов сайта; заголовок страницы (<title>) и Open Graph-разметка для соцсетей; предзагрузка скриптов для систем рекламы и аналитики (Prebid, Google GPT, Amazon).

<body>: Содержит структуру для одностраничного приложения (SPA). Вместо готового контента здесь размещены: пустые контейнеры <div> с идентификаторами (например, toaster-center, subscription-status-modal), которые являются точками монтирования для Vue.js-приложения; множество скриптов, которые инициализируют приложение, подключают аналитику (Sentry), live-обновления (PubSub) и управляют логикой страницы; скрытые <iframe> для работы систем рекламы и отслеживания.

2) Внутри <head>: Семантические элементы, описывающие содержание страницы, отсутствуют. Размещаются только служебные теги: <meta>, <title>, <link>, <script> и <style>.

Внутри <body>: Классические семантические элементы (такие как <header>, <main>, <article>) практически не используются, что характерно для SPA. Их роль выполняют абстрактные контейнеры <div> с уникальными id, которые наполняются смыслом только после выполнения JavaScript-кода.

3) Атрибуты ориентированы на оптимизацию высоконагруженного интерактивного приложения:

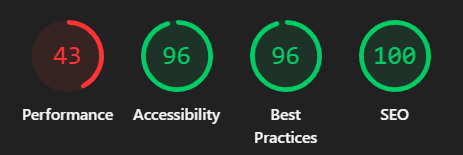
В <link>: rel=«preload», as=«font» – для приоритетной загрузки шрифтов и избежания задержек в отображении текста и фигур. crossorigin – для корректной загрузки ресурсов с CDN.

В <script>: async, defer – чтобы скрипты рекламы и аналитики не блокировали отрисовку главного контента и приложения. crossorigin=«anonymous» – для безопасной загрузки скриптов со сторонних доменов.

В <meta>: http-equiv=«origin-trial» – для тестирования экспериментальных функций браузера, обеспечивая обратную совместимость.

В <div>: data-\* атрибуты (напр., data-color-mode) – для передачи данных в JavaScript-приложение (управление темой, состоянием модальных окон).

Такой подход диктуется необходимостью совместить высокую производительность сложного SPA с интенсивной монетизацией через программатик-рекламу, обеспечивая при этом кросс-браузерную совместимость и безопасность.



Очень высокие показатели доступности, соответствия стандартам и поисковой оптимизации. Главная слабая сторона – низкая производительность, которая снижает общее качество и удобство использования.

2. <https://www.postcrossing.com>

1) Внутри тега <html>, который является корневым контейнером для всего документа, размещается два основных блока: <head> и <body>. Тег <head> не отображается для пользователя и содержит всю метаинформацию, критически важную для браузера и поисковых систем. Здесь это, прежде всего, технические мета-теги (такие как кодировка charset=“utf-8” и viewport), заголовок страницы (<title>), подробное описание проекта (<meta name=“description” content=“A postcard exchange project...”) и ключевые слова для SEO. Также здесь размещаются многочисленные ссылки на иконки и splash-экраны для Progressive Web App (PWA) на самых разных устройствах Apple, что указывает на сильную ориентацию проекта на мобильных пользователей. В <head> подключаются все внешние ресурсы: таблицы стилей CSS и скрипты, включая системы аналитики и рекламные модули от Google. В отличие от него, тег <body> содержит всё, что увидит конечный пользователь. На Postcrossing это, React-приложение, монтируемое в контейнер <div id=“root”>, а также множество скрытых служебных iframe для работы рекламы и капчи.

2) Внутри <head>: <title>: Заголовок документа. <meta>: Различные метаданные (кодировка, описание, viewport). <link>: Определяет отношения с внешними ресурсами (стили, иконки, манифест веб-приложения).

Внутри тега <body> семантическая структура минимальна, так как основной контент генерируется скриптами. Однако присутствуют: <div id=«root»>: Семантически обозначает корневой контейнер для клиентского приложения. <iframe>: Семантически обозначает встроенные фреймы для сторонних сервисов (реклама, капча). <script>: Семантически обозначает исполняемый код.

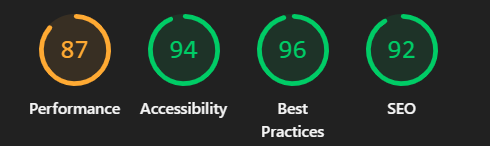
3) В <html>: lang=«en» – важен для доступности (скринридеры выбирают правильное произношение) и SEO (поисковики понимают, для какой аудитории контент).

В <meta>: charset=«utf-8» – для корректного отображения всех символов. name=«viewport» content=«width=device-width, ...» – для адаптивности под мобильные устройства. name=«description» – для SEO-оптимизации.

В <link>: rel=«icon», rel=«apple-touch-startup-image» – определяют тип подключаемого ресурса (иконка, заставка). href – путь к ресурсу. media=«(...)» и sizes – атрибуты тщательно подобраны для точного определения условий отображения splash-экранов на огромном спектре устройств Apple (iPhone, iPad разных поколений), что критически важно для мобильного пользовательского опыта.

В <script>: src – загружает указанный файл с сервера. async/defer – атрибуты для асинхронной загрузки скриптов, чтобы не блокировать отрисовку страницы, что повышает скорость её загрузки. crossorigin=«anonymous» – обеспечивает безопасную загрузку ресурсов с других доменов в соответствии с политикой CORS.

В <iframe>: Служебные атрибуты (например, name=«googlefcPresent») требуются для корректной работы конкретных рекламных технологий от Google.



Сбалансирован по всем направлениям. Особенно хорошо проработаны доступность, лучшие практики и SEO.

3. <https://open.spotify.com>

1) Внутри тега <html> размещается вся структура документа, включая метаинформацию в <head> и видимое содержимое в <body>. Его атрибуты, такие как lang=“ru” и dir=“ltr”, сразу указывают на то, что страница локализована для русскоязычных пользователей. Классы вроде encore-dark-theme и spotify\_container показывают использование фирменной дизайн-системы Spotify (Encore) и то, что приложение изначально загружается в тёмной теме, что соответствует его узнаваемому стилю.

2) Внутри тега <head> семантические элементы носят исключительно служебный характер и предназначены для браузера и поисковых систем: <title>, <meta>, <link>, <script>, <style>, <noscript>.

Внутри тега <body> на момент первоначальной загрузки семантических элементов очень мало, так как SPA строит DOM-дерево динамически. Можно обнаружить базовые структуры: <div> (основной контейнер приложения), элементы для модальных окон. После полной загрузки JavaScript добавляются семантические теги, такие как <header>, <main>, <nav>, <button>, <section> и другие, которые формируют полноценный, доступный интерфейс плеера.

3) class и id – для применения CSS-стилей и selektirovania элементов в JavaScript. Обилие классов с префиксами (напр., encore-) говорит об использовании строгой дизайн-системы и методологии CSS (напр., БЭМ) для поддержания порядка в стилях большого проекта.

src и href – для указания путей к внешним ресурсам: скриптам, стилям, шрифтам, изображениям. Spotify активно использует CDN для быстрой доставки статики пользователям по всему миру.

Атрибуты производительности и безопасности:

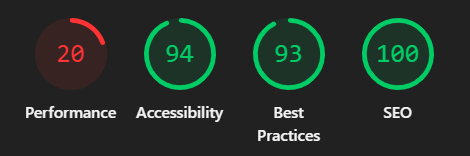
rel=«preconnect» – предподключается к сторонним серверам (API, CDN) для ускорения загрузки.

crossorigin=«anonymous» – обеспечивает безопасное выполнение запросов к ресурсам на других доменах в соответствии с политикой CORS.

defer / async – управляют порядком выполнения скриптов, чтобы не блокировать парсинг и отрисовку страницы.

type – сообщает браузеру, какой тип контента загружается (например, type=«text/javascript»).

Количество атрибутов обусловлено масштабом и сложностью проекта. Spotify – это высоконагруженное приложение, которое использует множество внешних сервисов (аналитика, API, CDN), фреймворк для построения интерфейса и строгие правила дизайна. Все эти атрибуты необходимы для оптимизации скорости загрузки, обеспечения безопасности, корректного отображения интерфейса и сбора аналитических данных.



Три ключевых направления – доступность, стандарты и SEO – реализованы на отлично. Единственным и критически важным недостатком является низкая производительность, подрывающая общую эффективность ресурса.

4.https://www.wildberries.by

1) <html>: Служит контейнером для всего содержимого документа. Его основная функция – объявить для браузера, что это веб-страница, а также указать язык её содержимого. Информация внутри: Весь контент страницы, включая <head> и <body>, также содержит ключевой атрибут lang, указывающий код языка.

<head>: Содержит метаинформацию, инструкции по обработке страницы и ссылки на внешние ресурсы. С помощью <meta> можно сообщать браузеру, поисковому роботу или другому устройству различную служебную информацию (или метаинформацию) о сайте: кодировку текста, описание контента и так далее. Например: <meta name=«description» content=«краткое описание»> – краткое описание (или аннотация) страницы. <meta charset=«UTF-8»> – определяет кодировку символов (UTF-8 для корректного отображения кириллицы и других символов). Также в <head> есть ссылки на ресурсы, обозначенные тегом <link>, подключающие внешние файлы стилей, иконки сайта. Здесь также подключен внешний script через тег <script>, а также прописаны стили через <style> тег. <title> – заголовок страницы (отображается на вкладке браузера и в результатах поиска).

<body>: Содержит всё видимое содержимое сайта. Здесь используются flex box. Внутри блока <body> располагаются блоки <div> с атрибутами <class> и блоки <script>. В первом <div> находится тег <header> отражающий шапку сайта. Еще один блок <div>, отражающий меню выбора категории товара в поисковой системе. <main> содержит сам каталог и карточки товаров, фильтры. Завершающим в первом <div> является подвал <footer> с контактами, политикой сайта и ссылками. Следующий <div> отражает иконки снизу сайта с небольшой навигацией: главная страница, поиск по категориям, корзина, профиль и т.д. В нем находятся еще тег <div>, а также теги <a>, отвечающие за ссылки. Также есть отдельный <div> для сообщений в поддержку. А также множество блоков <script> с атрибутами type=«text/javascript» и src=«ссылка на файл JavaScript».

2)Внутри тега <head> находятся следующие семантические теги: <title>,<meta>,<link>.  
Внутри тега <body>: <header>, <main>, <footer>, <h1>, <h2>, <section>, <article>.

3)class, id – стилизация и работа JS.

В <html>:

lang=«ru» важно для доступности (скринридеры выбирают правильное произношение) и SEO (поисковики понимают, для какой аудитории предназначен контент)

translate=«no» служит для указания браузеру, что содержимое данного HTML-элемента не нужно переводить.

class=«adaptive» атрибут для применения CSS-стилей и JavaScript

В <meta>:

charset – корректное отображение символов.

description – краткое описание (или аннотация) страницы

viewport– адаптивность под устройства.

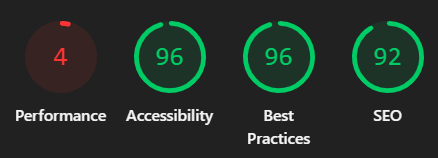
В<link>:

rel, href – подключение ресурсов.

в <script>:

src – загружает указанный файл с сервера и выполняет его код

type – сообщает браузеру, какой тип контента.



Наблюдается выраженный дисбаланс: по трем параметрам сайт показывает эталонное качество, в то время как низкая производительность выступает его главным слабым местом, ухудшая итоговое качество продукта.

5.https://www.codewars.com/dashboard

1)<html>: Служит контейнером для всего содержимого документа. Его основная функция – объявить для браузера, что это веб-страница, а также указать язык её содержимого. Информация внутри: Весь контент страницы, включая <head> и <body>, содержит ключевой атрибут lang, указывающий код языка.

<head>: Содержит метаинформацию, инструкции по обработке страницы и ссылки на внешние ресурсы. С помощью <meta> можно сообщать браузеру, поисковому роботу или другому устройству различную служебную информацию (или метаинформацию) о сайте: кодировку текста, описание контента, ключевые слова и так далее. <meta charset=«UTF-8»> – определяет кодировку символов (UTF-8 для корректного отображения кириллицы и других символов). Также в <head> есть ссылки на ресурсы, обозначенные тегом <link>, подключающие внешние файлы стилей, иконки сайта. Здесь также подключены внешние script через теги <script>, а также прописаны CSS стили через <style> тег. <title> – заголовок страницы.

<body>:

Содержит всё видимое содержимое сайта.Внутри блока <body> располагаются 3 больших блока <div> с атрибутами <class> и <id> и блок <script>. В первом <div> находится еще один блок <div>, отражающий меню с тремя полосками («бургер»). Внутри него тег <header> отражающий шапку сайта(профиль, уведомления и т.д). <main> содержит саму страницу с контентом. Здесь тоже есть использование flex box. И <nav> навигационная панель сбоку сайта с информацией о сайте, ссылками на чаты и обсуждения, а также тренировками. Завершающим в первом <div> является подвал <footer>. А в следующем <div> находятся теги <script>.

2)Внутри тега <head> находятся следующие семантические теги:<title>,<meta>,<link>, <style>.

Внутри тега <body>: <header>, <main>, <footer>, <h3>, <h4>, <section>, <nav>, <li>, <ul>.

3)class, id – стилизация и работа JS.

В <html>:

lang=«en» важно для доступности и поисковики понимают, для какой аудитории предназначен контент.

В <meta>:

charset – корректное отображение символов.

description – краткое описание (или аннотация) страницы

keywords – ключевые слова, с помощью которых можно найти сайт

viewport– адаптивность под устройства.

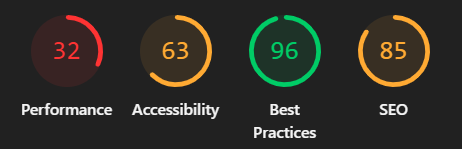
В <link>:

rel, href – подключение ресурсов.

в <script>:

src – загружает указанный файл с сервера и выполняет его код

type – сообщает браузеру, какой тип контента.



Сайт технически совершенен и хорошо оптимизирован для поиска, но при этом практически непригоден для комфортного использования и исключает часть аудитории.

6.https://translate.yandex.ru/ocr

1)<html>: Служит контейнером для всего содержимого документа. Его основная функция – объявить для браузера, что это веб-страница, а также указать язык её содержимого. Информация внутри: Весь контент страницы, включая <head> и <body>.

<head>: Содержит метаинформацию, инструкции по обработке страницы и ссылки на внешние ресурсы. С помощью <meta> можно сообщать браузеру, поисковому роботу или другому устройству различную служебную информацию о сайте: кодировку текста, описание контента, ключевые слова и так далее. В meta прописаны тип (website), описание сайта, контент заголовка. <meta charset=«UTF-8»> – определяет кодировку символов (UTF-8 для корректного отображения кириллицы и других символов). Также в <head> есть ссылки на ресурсы, обозначенные тегом <link>, подключающие внешние файлы стилей, иконки сайта.Здесь также подключены внешние script через теги <script>, а также прописаны стили через <style> тег. Тег <title> – заголовок страницы.

<body>: Содержит всё видимое содержимое сайта.

Внутри блока <body> располагаются блоки <div> с атрибутами <class> и <id>, блоки <svg> и <script>. Здесь же тег <header> отражающий шапку сайта: выбор типа перевода (с картинки, текстом и т.д.), название и аккаунт. Внутри <header> содержатся блоки <div>. Здесь тоже есть использование flex box. <main>, находящийся внутри блоков <div>, содержит саму страницу с контентом. В этих же блоках находится элемент, меняющий местами перевод (с русского на английский и обратно, например), а также боковая панель. Имеется <footer>, в котором находятся полезные ссылки, чат с поддержкой, социальные сети, выбор языка страницы. В <body> есть блок <style> с атрибутом id.

2)Внутри тега <head> находятся следующие семантические теги: <title>,<meta>,<link>.

Внутри тега <body>: <header>, <main>, <footer>, <h1>, <h2>, <section>, <label>, <textarea>, <img>.

3)class, id – стилизация и работа JS.

В <html>:

data-align=«left» выравнивание по левому краю

lang=«ru» важно для доступности и поисковики понимают, для какой аудитории предназначен контент.

В <meta>:

charset – корректное отображение символов.

description – краткое описание (или аннотация) страницы

keywords – ключевые слова, с помощью которых можно найти сайт

viewport– адаптивность под устройства.

В <link>:

rel, href – подключение ресурсов.

В <script>:

src – загружает указанный файл с сервера и выполняет его код

type – сообщает браузеру, какой тип контента

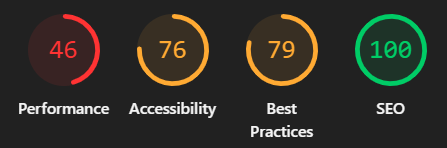
async – управляет асинхронной загрузкой и выполнением скрипта

В <style>:

nonce – разрешает выполнение только тех встроенных стилей, которые были сгенерированы сервером и имеют правильный одноразовый ключ

В <g> (атрибут SVG-графики):

stroke – это атрибут, определяющий цвет обводки элемента.



Сайт демонстрирует выдающееся качество SEO-оптимизации, что является его главным преимуществом, но есть проблемы с производительностью.