Санкт-Петербургский государственный университет

**Кафедра технологии программирования**

**Перминов Сергей Александрович**

**Выпускная квалификационная работа бакалавра**

**Продвижение веб-сайта**

Направление 010400

Прикладная математика и информатика

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Должиков В. В.

Санкт-Петербург

2017

# Содержание

Введение 3

Постановка задачи 4

Обзор литературы 5

Глава 1. Принципы работы поисковых систем 6

Глава 2. Внутренние факторы ранжирования и их оптимизация 9

2.1. Факторы ранжирования 9

2.2. Текстовое содержание 10

2.3. Заголовки 12

2.4. URL 14

2.4.1. Редиректы 15

2.5. Внутренние ссылки 16

2.6. Schema.org и структурированные данные 17

2.7. Мета-теги и атрибуты 18

2.8. Коды состояний 20

2.9. Мобильная оптимизация 21

Глава 3. Внешняя оптимизация 24

Глава 4. Работа над сайтом 26

4.1. Описание сайта 26

4.2. Описание программы для анализа сайта 27

4.3. Обнаруженные проблемы и их устранение 30

4.4. Результаты 36

Выводы 37

Заключение 38

Список литературы 39

Приложение 42

Введение

C каждым днем сеть Интернет привлекает к себе всё больше новых пользователей, и вот уже для большинства из нас Интернет является основным источником информации. Естественно, подобное явление не могло не сказаться на развитии разного рода бизнеса. К примеру, число посетителей крупных сетевых изданий в наше время превышает тиражи многих российских газет, что приносит огромную прибыль владельцам данных ресурсов. Каждой компании в наше время необходимо не только своё представительство в сети, но и уметь повышать популярность своего веб-сайта, ведь в противном случае потенциальные клиенты попросту не узнают о необходимой для них компании.

Большую роль в этом играет поисковая оптимизация (Search Engine Optimization, SEO) – комплекс мер по работе с сайтом, направленный на увеличение его позиции в поисковой выдаче по релевантным сайту запросам, ведь именно из поисковиков как правило приходит большая часть посетителей. Поисковая оптимизация затрагивает много аспектов: от текста на странице до накопления внешних ссылок на сайт (ссылочной массы).

Задачи поисковой оптимизации меняются с течением времени ввиду постоянных изменений алгоритмов поисковых систем, и тем не менее существуют общие правила, которые продолжают иметь вес на протяжение нескольких последних лет.

В данной работе было решено рассмотреть и применить меры по улучшению внутренних частей сайта ввиду того, что задачи внешней оптимизации носят в основном маркетинговый характер.

Постановка задачи

Главной целью данной работы является продвижение сайта astro-consul.ru на более высокие позиции в поисковых системах по соответствующим тематике сайта запросам для привлечения большего количества посетителей.

Требуется исследовать сайт на предмет проблем, мешающих его продвижению в поисковых системах, и устранить их. Для этого необходимо:

1. Изучить факторы, влияющие на ранжирование сайта поисковыми системами и способы устранения проблем, связанных с ними;
2. Путем написания программы, автоматизировать процесс обнаружения проблемных факторов и выдачи рекомендаций по устранению этих проблем
3. Используя результаты программного анализа сайта, произвести необходимые изменения внутренней составляющей и получить улучшенные показатели в статистике по запросам для данного сайта, подтверждающие правильность предпринятых действий.

Обзор литературы

Так как для поискового продвижения необходимо было понять основные принципы работы поисковых систем, полезными оказались статьи сайтов Devaka.ru [3] и Moz.com [1], посвященных данному вопросу. По ходу изучения вопроса, была необходимость получить больше знаний о работе поискового робота и способах его регулирования, в чем помогла статья от Google о Googlebot [2] и статьи Moz.com [4] и Yandex [5] о файле robots.txt. Далее для повышения осведомленности в области поисковой оптимизации был полезен цикл статей компании Moz, посвященный внутренним факторам ранжирования. В статье [8] были опубликованы результаты исследований компании касательно факторов ранжирования, что позволило понять степень влияния тех или иных факторов. В статьях [9] и [10] описывались основные аспекты внутренней оптимизации, а также коротко некоторые факторы, что позволило определить направление для дальнейшего поиска информации. В статьях [14], [16], [19], [22], [26], [29], [30], [31] более подробно описывались внутренние факторы по отдельности и давались рекомендации по работе с ними. В статье [32] было рассказано о внешней оптимизации, которая является неотъемлемой частью мер по поисковому продвижению. При изучении внутренних факторов полезными были и статьи других источников. Так статьи сайта seo-akademiya.com [12] и [13] были полезны при изучении вопроса о текстовом содержании страниц и плотности вхождения слов, а статья [15] блога Devaka.ru позволила расширить понимание вопроса об оптимизации заголовков страниц.

Глава 1. Принципы работы поисковых систем

Поисковые системы выполняют две важные функции: собирают данные и выполняют индексацию. После этого пользователь получает ранжированный список сайтов, который согласно системе является наиболее релевантным пользовательскому запросу [1].

У поисковой системы имеется специальная программа для сбора информации – поисковый робот (также называемый «пауком» или «краулером»), чья задача – обнаруживать и добавлять в индекс поисковой системы новые или изменённые страницы [2]. При создании сайта, необходимо дать роботу знать о появлении нового ресурса, что можно сделать при помощи сервисов добавления URL, после чего робот сможет прийти на новый сайт, просканировать его, перейдя по всем внутренним ссылкам. Также робот составляет для сайта расписание, согласно которому он вновь посещает ресурс с целью обновления данных и внесения в индекс новых страниц [3]. Переходы по внутренним ссылкам особенно важны, так как переходы от документа к документу по внутренним ссылкам являются наиболее быстрым способом обхода всемирной сети [1].

Для установки ограничений для поискового робота используется файл с названием robots.txt (название и расширение файла пишутся строго строчными буквами), который должен находиться в каталоге верхнего уровня веб-сайта. Ниже представлено базовое содержание данного файла.

User-agent: [название поискового робота]

Disallow: [строка URL, который не разрешается сканировать]

Эти две строки образуют полноценный файл robots.txt, хотя строк с названиями поисковых роботов и URL может быть несколько [4]. В файле можно использовать спецсимволы \* и $, которые обозначают любую последовательность символов и конец правила соответственно. Также в нём можно указать главное зеркало сайта через директиву Host и адрес карты сайта sitemap.xml, который указывает, как организовано содержимое сайта, позволяя упростить роботу перемещение по нему и более точно индексировать страницы [5]. Ниже приведён пример содержимого robots.txt с указанием главного зеркала и карты сайта.

User-agent: \*

Disallow: /example-subfolder/

Host: example.com

Sitemap: http://www.example.com/sitemap.xml

Под «индексированием» подразумевается предварительная обработка документов, в результате которого создается вспомогательный файл индекса. Это инвертированный файл, то есть состоящий из двух частей: словаря, содержащего все встретившиеся термы, со ссылкой на массив данных пост-листа для каждого из них, и пост-листов, которые являются массивами информации обо всех вхождениях данного слова в документе [6]. Он нужен для того, чтобы быстро осуществлять по нему поиск. Процесс индексации происходит быстро после сбора информации с сайта, и лишь после того, как процесс завершится, страницы будут появляться в поисковой выдаче. Найти страницы можно по ключевым словам, присутствующим в тексте.

При поиске информации, введенный пользователем запрос анализируется, и в результате вычисляются веса для каждого из слов. Затем поиск проводится по инвертированным индексам и находятся все документы в базе данных поисковой системы, наиболее подходящие поисковому запросу. Найденные для данного запроса документы должны быть ранжированы таким образом, чтобы пользователь увидел первыми в списке те, которые наибольшим образом отвечают его потребностям [3].

Факторов, согласно которым поисковая система ранжирует результаты поисковой выдачи, достаточно много, однако их можно разделить на внешние факторы (off-page) и внутренние (on-page). Внешние факторы включают в себя:

* возраст сайта
* ссылочную массу
* другие факторы, на которые сложно повлиять веб-мастеру.

Внутренние факторы связаны со структурой сайта и его содержимым и находятся под контролем разработчика сайта, а потому всё, что с ними связано, может быть изменено в любой момент [7].

Глава 2. Внутренние факторы ранжирования и их оптимизация

2.1. Факторы ранжирования

В 2015 году компанией Moz было проведено исследование влияния факторов ранжирования на алгоритм Google, которое состояло из опроса экспертов в области поисковой оптимизации и исследования взаимосвязи факторов с результатами ранжирования. С их слов, результат нельзя считать прямым подтверждением того, чем на самом деле пользуются поисковые системы при ранжировании веб-сайтов, однако он показывает характеристики страниц, находящихся на высоких позициях. В итоге, сгруппировав по категориям, факторы были расположены следующим образом в порядке убывания их влияния.

1) Факторы, относящиеся к ссылкам, на уровне домена (основанные на количестве ссылок на сайт, уровне цитирования, доверия, PageRank домена и т.д.)

2) Факторы, относящиеся к ссылкам, на уровне страницы (PageRank, количество ссылок, текст ссылки (анкор), качество ссылающихся сайтов и т.д.)

3) Факторы, основанные на ключевых словах и содержании страницы (оптимизация использования ключевых слов, количество контента, его качество и релевантность, и т.д.)

4) Факторы на уровне страницы, не связанные с ключевыми словами (длина контента, его удобочитаемость, уникальность, использование набора атрибутов Schema.org или иной разметки структурированных данных, и т.д.)

5) Информация о трафике и поведении пользователя

6) Показатели, связанные с брендом, на уровне домена (использование бренда за пределами сети, упоминание в СМИ, и т.д.)

7) Использование ключевых слов на уровне домена (наличие ключевых или близких с ними по значению слов в названии домена)

8) Факторы на уровне домена, не связанные с ключевыми словами (длина названия домена, домен верхнего уровня и т.д.)

9) Социальные показатели (количество твитов со ссылками, репосты в социальных сетях и т.д.) [8]

2.2. Текстовое содержание

Контент – это то, ради чего пользователи в первую очередь ищут сайт, поэтому он так важен для поисковых систем, а значит владельцам сайтов важно создавать хорошее содержимое. С точки зрения поисковой оптимизации, хороший контент обладает двумя свойствами: на него есть спрос и на него ссылаются. При отсутствии внешних ссылок поисковая система вряд ли сможет ранжировать сайт и, следовательно, не сможет обеспечить его трафиком [9].

В прошлом внутренняя поисковая оптимизация опиралась на использование ключевых слов, в особенности использование наиболее значимых в основных местах на сайте. В настоящее время же поисковики способны понять смысловое содержание страницы путем использования синонимов, контекста или обращая внимание на частоту использования определенных словосочетаний. И хотя ключевые слова всё ещё имеют значение, методы вроде использования полностью совпадающих ключевых слов в специальных местах определенное количество раз уже не оказывают на поисковую оптимизацию огромного влияния. Что действительно важно, так это релевантность. Таким образом, внутренняя поисковая оптимизация заключается не в повторении и размещении ключевых слов, а в понимании того, кем являются пользователи сайта, что они ищут, и на какие темы можно создать контент, удовлетворяющий нужды пользователя [10].

Хотя отдельных ключевых слов недостаточно для формирования основы контента, всё начинается с исследования ключевых слов. Требуется выяснить, на какие слова стоит сделать упор, какова их популярность. В конечном итоге, цель – связать контент сайта с теми ключевыми словами, которые пользователи вводят в поисковик [11]. Для этого обычно пользуются инструментами подбора слов, такими как Yandex Wordstat или Google Keyword Planner, которые приводят статистику запросов по введенным словам.

Обычно поисковые запросы классифицируют по частотности следующим образом:

1) Высокочастотные запросы. Это, как правило, короткие фразы, которые запрашиваются чаще всего, и обычно имеют более 10000 показов в месяц. Ввиду их частотности, вокруг них соответственно высокая конкуренция, поэтому продвижение по таким словам является крайне трудной задачей.

2) Среднечастотные запросы. Имеют от 1000 до 10000 показов в месяц и включают в себя от двух до четырех слов. Они также являются популярными, но при этом более конкретными, что позволяет обеспечить большую конверсию.

3) Низкочастотные запросы. Как правило, это большие сочетания слов, запрашиваемые реже всего, однако продвижение под них позволит привлечь посетителей, которые вероятнее всего выполнят на сайте желаемое действие [12].

Путём экспериментов, специалистами сайта seo-akademiya.com было установлено, что плотность вхождения ключевых слов не должна превышать 4-5%, но при этом она не должна быть менее 1.5%. При превышении этого числа текст будет считаться "переоптимизированным", а страница его содержащая может оказаться под фильтром поисковых систем [13].

2.3. Заголовки

Заголовки страниц, находящиеся внутри тега <title> – один из важнейших факторов, позволяющий поисковым системам понять, что находится на странице. Они же оставляют первое впечатление на людей, нашедших страницу через поисковик.

Содержимое тега <title> используется в трех ключевых местах: на странице результатов поиска, в самих веб-браузерах и в социальных сетях. Заголовок, показывающийся в поисковой выдаче, и оставляет первое впечатление, поэтому даже при высоких позициях сайта заголовок может как привести потенциального пользователя на сайт, так и отвернуть от него.

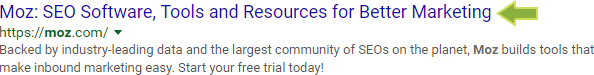


Рис. 1. Содержимое тега <title> на странице поисковой выдачи.

В браузерах же заголовок отображается в верхней части, и в случае, когда у пользователя открыто множество вкладок, хороший и уникальный заголовок поможет ему не упустить нужный сайт из виду.

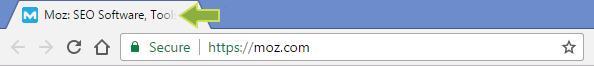


Рис. 2. Содержимое тега <title> в верхней части браузера.

Некоторые сторонние сайты, в особенности социальные сети, отображают текст из тега <title> в случаях, когда пользователь решил «поделиться» страницей.



Рис. 3. Содержимое тега <title> на странице Facebook.

Существуют рекомендации по оптимизации заголовка страницы, основанные как на экспериментах специалистов, так и на советах от сотрудников поисковых систем.

1) Если заголовок слишком длинный, поисковые системы могут урезать его, добавив в конце троеточие, а значит некоторые важные слова могут не отобразиться. Следует придерживаться лимита в 60 символов, однако более точный лимит базируется на ширине секции, отображающей заголовок, которая примерно 600 пикселей.

2) Несмотря на отсутствие штрафов за длинные заголовки, не следует переполнять их ключевыми словами. Заголовки, состоящие из вариаций одного и того же ключевого слова, рассматриваются поисковыми системами как спам.

3) Следует давать каждой странице уникальный заголовок. Таким образом поисковые системы смогут понять, что контент на сайте уникальный и представляет ценность.

4) Ключевые слова стоит помещать в начало заголовка. Согласно экспериментам компании Moz, ключевые слова, стоящие в начале, имеют большее влияние на ранжирование. Их исследование также показало, что пользователи обычно обращают внимание на первые два слова в заголовке, в связи с чем рекомендуется использовать заголовки, в которых уникальная составляющая страницы отображается в самом начале.

Оптимальный формат содержимого тега <title> выглядит следующим образом:

*Первичное ключевое слово - Вторичное ключевое слово | Название бренда*

*Название продукта - Категория продукта | Название бренда* [14]

Помимо тега <title>, существует тег <h1>. С его помощью выделяется заголовок внутри страницы. Поисковые системы также смотрят на его содержимое, поэтому наличие ключевых слов в нём имеет вес. Не рекомендуется делать <title> и <h1> одинаковыми, но их содержание должно пересекаться по наиболее важным для страницы ключевым словам. Так поисковым системам легче будет понять, по каким именно словам страница релевантна. Ввиду того, что содержимое <title> потенциальный пользователь видит ещё до перехода на сайт, <h1> следует делать коротким и чтобы он наилучшим образом описывал суть страницы. Это дает возможность добавить в <title> дополнительные ключевые слова, что и будет отличать его от <h1> [15].

2.4. URL

Хорошо составленный URL (или «веб-адрес») оказывает положительное влияние на поисковую оптимизацию по нескольким причинам.

Во-первых, он предоставляет как пользователям так и поисковым системам возможность быстро понять, что находится на странице. http://www.empireonline.com/movies/beauty-beast-2/review/ – пример хорошего URL, который также называют семантическим. Даже если заголовок страницы будет скрыт или обрезан, подобного рода адрес позволит получить четкое представление о содержании страницы. Более того, в данном URL прослеживается иерархия информации, находящейся на странице (рецензия, относящаяся к фильму «Красавица и чудовище», который в свою очередь относится к разделу «Фильмы»). Таким образом, поисковая система понимает, что на этой странице находится не просто некий обзор, а рецензия на конкретный фильм. И хотя URL может содержать идентификационные номера, наилучшей практикой является использование слов, понятных человеку.

Во-вторых, веб-адреса являются фактором ранжирования, пусть и незначительным. И хотя поисковые системы дают вес авторитетности домена в целом, наличие ключевых слов в URL также влияет на ранжирование. Тем не менее, не рекомендуется создавать не несущие пользы URL лишь для того, чтобы включить в них ключевые слова. Стоит отметить утверждение Google о том, что первые 3-5 слов в URL имеют больший вес, и чем длиннее адрес, тем меньший вес получают слова, стоящие далеко от начала.

В-третьих, в крайнем случае, хороший URL может служить текстом анкора для самого себя, когда его копируют и вставляют на форумах, в блогах, социальных сетях и т.д.

В случае необходимости слова в веб-адресе следует разделять дефисом. В URL не следует использовать нижние подчеркивания, пробелы или какие-либо другие символы для разделения слов. Также не следует использовать верхний регистр во избежание проблем со дублированным контентом и при возможности не использовать параметры или же использовать как можно реже [16].

2.4.1. Редиректы

Редирект – это способ перенаправить пользователей и поисковые системы с одного URL на другой, отличный от первоначального запроса. Далее будут описаны наиболее часто используемые виды редиректа.

1) 301 Moved Permanently

Редирект 301 – это постоянный редирект, который передает 90-99% ссылочного веса странице, на которую идёт перенаправление. В большинстве случаев это наилучший способ применения редиректа. Под весом ссылки подразумевается её характеристика, связанная с качеством и количеством её внешнего использования, и использующаяся поисковыми системами как показатель её значимости при подсчете PageRank и тематического индекса цитирования (тИЦ). На ссылочный вес оказывает влияние тематическая близость ссылающегося сайта [17].

2) 302 Found (HTTP 1.1) / Moved Temporarily (HTTP 1.0)

Данный редирект предполагается временным и не передаёт ссылочный вес.

3) 307 Moved Temporarily (Только для HTTP 1.1)

Редирект 307 – это преемник редиректа 302 в протоколе HTTP 1.1. Если поисковые системы уже признали ваш сервер совместимым с HTTP 1.1 и контент действительно был временно перемещён по другому адресу, то тогда стоит использовать редирект с кодом 307. Но так как определить, признали ли поисковые системы совместимость, практически невозможно, то в случае временного перемещения контента лучше использовать редирект 302. Во всех остальных случаях рекомендуется использовать 301 [18].

2.5. Внутренние ссылки

Посредством внутренних ссылок осуществляется навигация по сайту, и они помогают создать иерархию информации и распределить вес ссылки по сайту. Для того, чтобы проиндексировать отдельно взятую страницу, краулеры должны видеть на ней контент, но помимо этого им требуется, чтобы страницы сайта были связаны между собой ссылками, чтобы найти все страницы. Каким бы хорошим не было содержимое страницы, она практически не существует для поисковой системы, если робот не может до неё добраться [19], поэтому при поисковой оптимизации важен такой этап как внутренняя перелинковка. Она выполняется как при помощи ссылок в тексте страницы, так и путём составления карты сайта [20], представляющей из себя страницу со списком из ссылок на все имеющиеся на сайте страницы [21].

Оптимальная структура веб-сайта похожа на пирамиду, где верхушка – это домашняя страница.

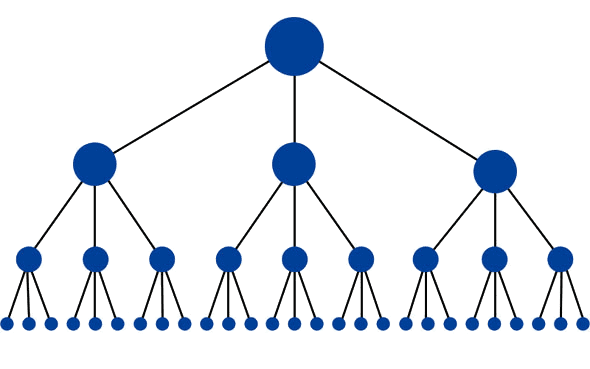


Рис. 4. Оптимальная структура веб-сайта.

Подобная структура имеет минимально возможное количество ссылок с домашней страницы на любую другую, что полезно для распределения ссылочного веса по всему сайту, увеличивая рейтинг каждой страницы. К тому же, это значительно упрощает навигацию [19].

2.6. Schema.org и структурированные данные

Schema.org – это специальный набор атрибутов, который при использовании его в HTML-коде позволяет улучшить представление сниппета – информации о странице, отображающейся в результатах поиска .



Рис. 5. Различие в сниппетах при использовании Schema.org.

Первый сниппет содержит рейтинг и дату публикации, которые можно добавить при помощи Schema.org. Второй же отображает либо текст из мета-тега description либо иную информацию, выбранную поисковой системой [22].

При помощи этой разметки можно структурировать данные самых разных видов, от продуктов до информации о мероприятиях или рецептах. Чаще всего она используется для предоставления дополнительной информации о персонах, местах, организациях, продуктах, творческих работах. У каждого вида данных есть свойства, которые можно использовать для более детального описания. Так у вида данных "Person" есть такие свойства как "name", "telephone", "email", "jobTitle" и другие [23].

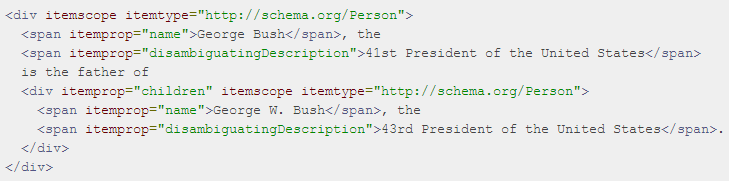


Рис. 6. Пример использования Schema.org [24].

2.7. Мета-теги и атрибуты

Мета-теги – это способ, посредством которого вебмастеры могут предоставить поисковым системам информацию об их сайтах. Мета-теги добавляются в секцию <head> HTML-страницы. Каждая система обрабатывает лишь те мета-теги, которые понимает, игнорируя остальные [25].

Есть определённые мета-теги, которые должны быть на каждой странице.

1) Content-Type. Этот тег необходим для определения кодировки, используемой страницей, а также типа содержимого. Отсутствие тега может отрицательным образом сказаться на отображении страницы в браузере. Пример: <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />.

2) Description – Данный тег не влияет на ранжирование, но тем не менее является важным. Содержимое тега выводится на странице результатов выдачи в качестве описания и оно призвано привлекать внимание пользователя. Текст для этого тега следует держать в пределах 160 символов.

3) Viewport – Просмотр интернет-страниц с мобильных устройств весьма распространен в наше время, поэтому следует использовать данный тег для адаптации сайта к просмотру на мобильных устройствах. Стандартный тег выглядит так: <meta name=viewport content="width=device-width, initial-scale=1">. С данными параметрами он адаптирует ширину окна просмотра к экрану устройства и позволяет обеспечить соотношение 1:1 между пикселями CSS и независимыми пикселями устройства.

Содержимое тега keywords не используется большинством поисковых систем [26], тем не менее робот Яндекса может учитывать его при определении соответствия страницы поисковым запросам [27]. Основной рекомендацией является не переполнять содержимое этого мета-тега всевозможными ключевыми словами и их вариациями, так как для некоторых поисковых систем это может послужить сигналом для наложения штрафа, особенно если окажется, что содержимое страницы также переоптимизировано [28].

Атрибут alt позволяет дать описание изображения или другой графической информации, находящейся на странице. Если загрузить изображение по какой-либо причине невозможно, то значение атрибута alt будет выведено вместо него. Это позволяет поисковым системам лучше понять смысл изображения и индексировать его должным образом. Заполнение этого атрибута является не только одним из принципов доступного веб-дизайна, позволяя описывать изображения тем, кто не может их увидеть (браузер блокирует изображения или у человека проблемы со зрением и он пользуется экранным диктором), но и может положительно повлиять на поисковую оптимизацию. Хотя распознавание изображений поисковыми системами со временем становится только лучше, они все равно не могут видеть изображения так, как видит их человек. Результаты интерпретации изображений без дополнительной информации могут оказаться неверными, в связи с чем страница может быть ранжирована по не интересующим владельца сайта ключевым словам, а то и вовсе не попасть в рейтинг. К тому же, текст атрибута alt предоставляет ещё одну возможность использовать ключевое слово, а так как использование ключевых слов на странице всё ещё является важным фактором, в интересах владельца сайта написать текст атрибута alt, который не только хорошо описывал бы изображение, но и включал в себя ключевое слово или фразу, однако не переполняя ими текст. Помимо этого, рекомендуется составлять текст относительно коротким, так как популярные экранные дикторы обрезают текст alt примерно около 125-го символа. Атрибут alt стоит прописывать и изображениям, выполняющим функцию кнопок, как и любым кнопкам в используемых на сайте формах. Оставлять значение атрибута пустым также не рекомендуется [29].

2.8. Коды состояний

Кодов состояний HTTP очень много, однако для поисковых систем и продвижения сайта в них, помимо вышеупомянутых редиректов, важны следующие:

1) 200 OK. Запрос выполнен успешно. В большинстве случаев это корректный ответ сервера.

2) 404 File Not Found. Сервер не нашёл ничего, что соответствовало бы запрашиваемому веб-адресу. При этом не указывается, временное это состояние или постоянное. Данный ответ сервера должен происходить всякий раз, когда по запросу не было найдено соответствующей страницы.

3) 410 Gone. Запрашиваемый ресурс более недоступен. Данное состояние предполагается постоянным. Если сервер не может определить, является ли состояние постоянным, то вместо кода 410 следует использовать 404.

4) 503 Service Unavailable. Либо сервер не способен справиться с большим количеством запросов, либо ведутся технические работы. Данный код следует использовать всякий раз, когда страница или сайт недоступны временно. Таким образом поисковые роботы смогут понять, что спустя время они могут вернуться на сайт или страницу.

Если страница имеет важные внешние ссылки, получает большой объем трафика, или имеет очевидный URL, по которому хотят перейти пользователи, но при этом существует более релевантная на данный момент страница, то следует применять редирект на эту страницу с кодом 301. В иных случаях может появиться необходимость намеренно выдавать ошибку 404 по требуемому адресу, таким образом предотвратив индексацию и повторное сканирование поисковыми системами.

Существует ошибочное представление, что наилучшей практикой для поисковой оптимизации будет обеспечивать редирект 301 со страниц, которые выдают ошибку 404, на главную страницу. В большинстве случаев так делать не рекомендуется ввиду того, что пользователь может не понять, что страница, на которую они пытаются перейти, просто не существует. Поэтому, когда пользователь перешел по адресу, который выдаёт ошибку 404, у него должна быть возможность перемещаться по сайту, не покидая его. С этой целью создаются страницы ошибок 404, которые призваны уведомлять пользователей о том, что искомая страница не существует, при этом имея возможность дальнейшего перемещения по сайту [30].

2.9. Мобильная оптимизация

С каждым годом люди всё больше времени проводят со своими мобильными устройствами, будь то телефоны или планшеты, однако дизайн многих сайтов все еще не учитывает разные размеры экранов и скорость загрузки. Мобильная оптимизация затрагивает дизайн сайта, его структуру, скорость загрузки и многое другое дабы не отвернуть от сайта потенциальных посетителей с мобильных устройств. Если сайт уже достаточно хорошо оптимизирован под поисковые системы, следует уделить внимание данному аспекту продвижения. Помимо обеспечения адаптивного дизайна, необходимо учесть следующие моменты:

1) Скорость загрузки. Ввиду возможных проблем с соединением и аппаратным обеспечением, скорость загрузки важна для пользователей мобильных устройств куда больше, чем для пользователей ПК. Под этот пункт попадает оптимизация изображений, минимизация кода, уменьшение количества редиректов и многое другое.

2) Не нужно блокировать CSS, JavaScript и изображения. Всё это поддерживается современными мобильными устройствами и помогает поисковой системе понять, адаптирован сайт или просто имеет другое решение для мобильных устройств.

3) Не следует использовать Flash. Данный плагин может быть недоступен пользователю, из-за чего он не сможет увидеть эту важную часть страницы.

4) Не используйте всплывающие уведомления. Их может быть тяжело закрыть на мобильных устройствах, и это может привести к тому, что пользователь уйдёт с первой же страницы.

5) Нужно учитывать размер пальца. Навигация при помощи сенсорного экрана может привести к случайным кликам, если кнопки слишком большие, слишком маленькие или попадаются под палец во время прокрутки страницы.

6) Оптимизация текстов тега <title> и мета-тега <description>. Пространства на экране мобильного устройства меньше, поэтому следует быть кратким в данных тегах, при этом не забывая про качество информации.

7) Использование Schema.org позволит выделить расширенные сниппеты на фоне остальных, что будет особенно заметно на экране мобильного устройства [31].

Глава 3. Внешняя оптимизация

Под внешней оптимизацией понимают меры по влиянию на позицию сайта в поисковой выдаче, которые применяются за пределами продвигаемого веб-сайта. Путем оптимизации внешних факторов поисковые системы и пользователи получают более хорошее представление о популярности сайта, его релевантности, авторитетности и надежности. Этого можно достичь при наличии ссылок на продвигаемый сайт с других сайтов с хорошей репутацией, которые тем самым «ручаются» за качество контента на том сайте, на который они ссылаются. Внешняя оптимизация важна потому, что, несмотря на постоянные изменения в алгоритмах поисковых систем, эти меры по-прежнему играют большую роль в ранжировании сайта.

В основе внешней оптимизации лежит наращивание ссылочной массы. Поисковые системы используют внешние ссылки как показатель качества контента сайта, на который они ссылаются, поэтому сайт с большим количеством внешних ссылок будет находиться выше в поисковой выдаче нежели точно такой же сайт, но на который ссылаются реже.

В зависимости от способа получения, ссылки можно разделить на три типа:

1. Естественные ссылки, полученные без какого-либо вмешательства со стороны владельца сайта
2. Ссылки, полученные путём целенаправленных действий по наращиванию ссылочной массы, например, посредством убеждения клиентов ссылаться на ваш сайт
3. Ссылки, созданные самостоятельно, например, добавляя их в подписи на форумах, оставляя в комментариях и т.п.

На ссылки третьего типа поисковые системы смотрят с подозрением, поэтому не рекомендуется злоупотреблять подобными методами. Тем не менее, несмотря на то, каким способом были получены ссылки, наибольший вклад внесут те, которые обладают наибольшей ценностью. Популярность ссылающегося сайта, его надежность, авторитетность, тематическая схожесть, а также текст анкора и то, насколько она свежая – всё это показатели ценности ссылки.

И хотя наращивание ссылочной массы является наиболее часто практикуемым способом внешней оптимизации, любую деятельность, направленную на улучшение позиций сайта в поисковиках и при этом не касающаяся его внутренней составляющей, можно назвать внешней оптимизацией. К таким действиям относят продвижение в социальных сетях, гостевой блоггинг и другие [32].

Глава 4. Работа над сайтом

4.1. Описание сайта

В качестве объекта исследования был взят личный сайт астролога Александра Турсина. Тематикой сайта является оказание астрологических услуг в форме консультирования. Сайт был создан в 2015 году полупрофессиональным образом. Вёрстка осуществлена студентом технического ВУЗа на основании идей, разработки и дизайна самого автора. Углублённых мероприятий по поисковой оптимизации не проводилось. Были предприняты только общие меры, чтобы не допускать грубых ошибок.

После индексирования поисковыми системами в течение нескольких месяцев с даты открытия 15 сентября 2015 года сайт достиг некоторого невысокого уровня посещаемости, который продолжал оставаться в дальнейшее время на примерно одинаковом низком уровне.

В частности, анализ поисковых запросов с помощью системы Yandex давал статистику по запросам в 1.5 клика и около 15 показов в день за весь период с отклонениями 0.14-3.25 и 8-35 соответственно. Разброс по статистике с существенными отклонениями возникал в связи с новыми публикациями на сайте либо отстутствием таковых, дающими отдельные всплески активности посещений или её снижение.

Несколько меньшие цифры как кликов, так и показов страниц были получены в статистике Google. Ниже в таблице приводятся сравнительные данные статистики двух поисковых систем, взятые в произвольный отрезок времени продолжительностью 1 год с 10.04.2016 г. по 10.04.2017 г.

*Таблица 1. Данные статистики поисковых систем.*

|  |  |  |
| --- | --- | --- |
|  | Google | Yandex |
| Всего кликов | 325 | 511 |
| Всего показов | 4544 | 4693 |
| CTR | 7.16% | 10.89% |

4.2. Описание программы для анализа сайта

Визуальный анализ сайта требует больших временных затрат, и кроме того, есть вероятность упустить некоторые проблемы. Чтобы исключить человеческий фактор и ускорить процесс, на языке Python была написана программа, позволяющая автоматизировать процесс выявления проблемных мест на сайте и предлагающая способ устранения проблем там, где это возможно.

Для работы с HTTP-запросами была использована библиотека requests [33], а также lxml.html для обработки HTML и возможности получения элементов страницы посредством XPath-запросов [34].

Так как сайт представляет из себя совокупность связанных между собой ссылками HTML-страниц, необходимо собрать все имеющиеся на данном сайте ссылки путем перехода по каждой из них, после чего выделить уникальные, а также те, которые ведут именно на HTML-страницы, которые и требуется проанализировать. Также это в определенной мере позволяет увидеть сайт глазами поискового робота, так как если на сайте имеется страница, не связанная ссылками ни с какой другой страницей на сайте, робот и эта программа не смогут её обнаружить.

Для начала проверяется наличие переадресаций (редиректов) с главной страницы на другой адрес, и если они имеются, в качестве адреса главной страницы принимается тот, от которого последовал код ответа 200, который далее записывается в список ссылок, по которым ранее был осуществлен переход. С данной страницы собираются все имеющиеся ссылки и формируется набор уникальных ссылок, откуда исключаются пустые ссылки и ссылки с параметрами, в которых присутствует символ '?', так как по данным ссылкам находится идентичный контент. Ссылкам из данного набора посредством словаря (dict) присваивается уровень вложенности 1. Затем по данному набору в цикле осуществляется переход по каждой ссылке из набора с последующим ее добавлением в список ранее использованных для сбора ссылок с этих страниц. В список новых ссылок добавляются только те, которые не пересекаются с набором, по которому осуществляется цикл. По окончанию перехода по всем ссылкам из первого набора, начинается переход по ссылкам из нового набора, которым присваивается уровень вложенности на 1 выше, чем у ссылок из предыдущего набора. Если набор новых ссылок оказался пуст, цикл завершается. Из полученного списка ссылок удаляются те, которые не ведут на HTML-страницы. Проверяется наличие всех ссылок из данного списка на какой-либо одной странице сайта, что позволяет сделать вывод о наличии HTML-карты сайта или ее отсутствии. Далее страницы, на которые ведут ссылки из полученного списка, анализируются на предмет проблем.

В первую очередь проверяется соответствие уровня вложенности страницы со структурой URL, и при наличии расхождений появляется соответствующее уведомление. Далее анализируется текстовое содержание страницы. Для этой выполнения этой задачи было решено воспользоваться библиотекой pymorphy2, которая представляет из себя морфорогический анализатор для русского и украинского языков, составленный авторами статьи [35]. Текст избавляется от пунктуации, приводится к строчным буквам, делится на список слов и далее осуществляется лемматизация для каждого слова из списка посредством возможностей упомянутой ранее библиотеки. Полученный список затем избавляется от стоп-слов, которые были взяты из набора, предоставляемый библиотекой nltk [36]. Далее считается количество употреблений каждого слова и считается их отношение к количеству слов во всём тексте. Таким образом можно получить плотность вхождения слов в тексте. Для упрощения выделяется 10 самых употребляемых слов, плотность вхождения которых и рассматривается. Пользователь будет предупрежден при наличии слов со слишком большим процентом употребления, что может служить сигналом о переоптимизации, а также о том, что плотность вхождения самого употребляемого слова ниже минимума в 1.5%.

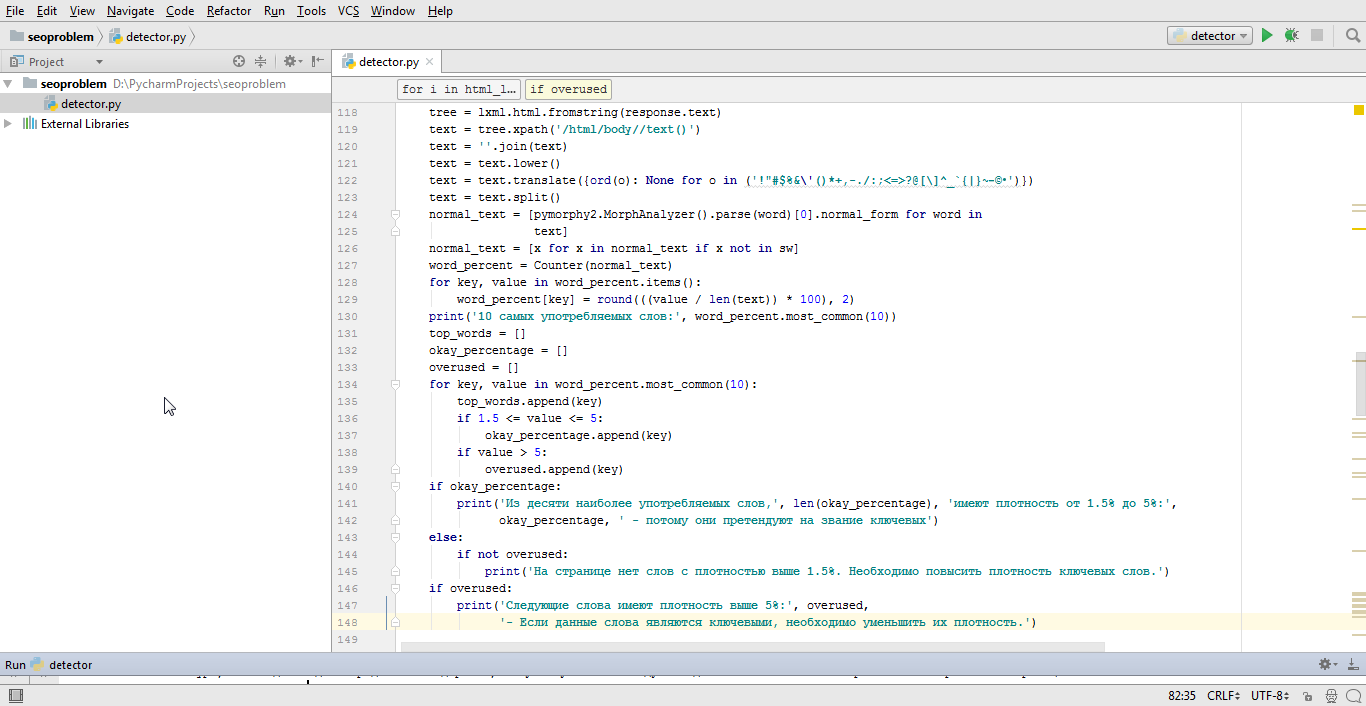


Рис. 7. Пример части программы, отвечающей за анализ текста.

Следующим этапом проверяются заголовки страниц (содержимое тегов <title> и <h1>). Как и в случае с текстом страницы, они избавляются от стоп-слов, пунктуации, проходят лемматизацию, после чего смотрится на пересечение слов заголовка с десятком наиболее употребляемых слов текста страницы, что позволяет сделать вывод о наличии ключевых слов в заголовке, как и о его релевантности содержимому страницы. Выдаются предупреждения при отсутствии пересечения, а также рекомендации по увеличению или уменьшению количества ключевых слов в заголовке в зависимости от их количества. Проверяется и наличие пересечения содержимого тегов <title> и <h1>.

Далее проверяется наличие атрибутов Schema.org на странице и разметка какого вида данных используется.

Проверяется наличие и корректное заполнение мета-тегов content-type, description, keywords и viewport. В случае с keywords, проверяется количество слов в данном теге, и при большом их количестве следует предупреждение. Также проверяется и заполнение атрибутов alt у изображений, но предупреждение вызывается только при наличии незаполненных атрибутов.

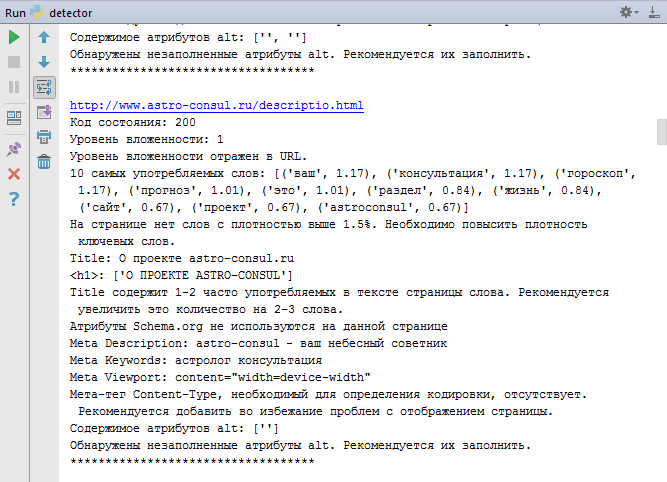


Рис. 8. Пример результата анализа страницы данной программой.

4.3. Обнаруженные проблемы и их устранение

В процессе обработки страниц сайта данной программой было выяснено следующее:

Cодержимое тегов <title> является недостаточным для определения содержания многих страниц, так как отсутствуют важные ключевые фразы. Более того, содержимое тегов <h1> в большинстве случаев полностью совпадает с <title>, представляя из себя литературно-художественный текст, неадаптированный под задачи поиска. URL не отражает уровень вложенности страниц для всех страниц уровня 2.

Кроме того, на сайте отсутствует разметка структурированных данных, хотя присутствует раздел «Контакты», оригинальные статьи и цены на услуги, которые можно разметить согласно Schema.org.

Изображения в подавляющем большинстве не имеют описания в виде значения атрибута alt, что является упущением использования дополнительной возможности употребления необходимых ключевых слов.

Отсутствует мета-тег content-type, объявляющий тип содержимого и кодировку, который следует добавить во избежание проблем с некорректным отображением содержимого страницы в браузерах. Мета-тег keywords на важных страницах наполнен большим количеством слов, что может послужить плохим сигналом для поисковых систем, и поэтому требует сокращения количества слов в нем.

Несмотря на то, что уровень вложенности страниц не превышает двух кликов (три клика для ссылок на скачивание файлов), а количество страниц весьма небольшое, отсутствует карта сайта, как отдельная страница для улучшения перелинковки.

Что касается текстового содержания, то, несмотря на наличие тематических терминов на страницах, плотность необходимых ключевых слов, составляющих целевые запросы, на главной и других страницах недостаточна и редко превышает 1.5%.

По замыслу автора сайт должен опознаваться поисковой системой и выдаваться по запросу "астрологические консультации". Однако проверка текста показала, что плотность вхождения слов "астрология" и "консультации" в тексте главной страницы равнялась 0.94% и 0.78%, что меньше упомянутого ранее минимума в 1.5%, что требовало внесения изменений в текст страницы.

При изучении похожих по тематике запросов выяснено, что наибольшей популярностью из всех слов по астрологической тематике пользуется запрос по слову "гороскоп". К примеру по данным Yandex за последний месяц сделано 7 005 714 показов.

Было решено воспользоваться этим ключевым словом в разделе, посвящённом астрологическому анализу жизни известных личностей, назвав его "Гороскопы известных людей". Согласно статистике сервисов для вебмастеров от Google и Yandex, по данному запросу сайт сделал существенную динамику, улучшив позиции в поиске с последних страниц до второй.

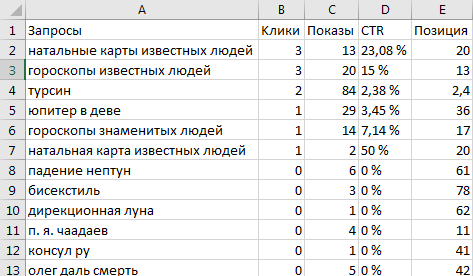


Рис. 9. Статистика по запросам из Google Search Console после внесенных изменений.

Вместе с тем, это слово не противоречит основной цели сайта, так как расчет гороскопов является одной из оказываемых услуг. Было решено добавить больше информации о данном виде услуг, попутно увеличивая плотность вхождения слов «расчет» и «гороскоп».

Ради эксперимента было сделано несколько вариантов запросов на статьи сайта. Выяснилось, что на популярный запрос "гороскоп + имя какой-либо знаменитости" (например, "гороскоп Шукшина") поисковик показывает страницу сайта только в том случае, если эта фраза была употреблена в первой сотне слов текста.

Так на запрос "гороскоп Гурченко" поисковая система Yandex поставила соответствующую статью сайта astro-consul.ru на первую позицию только потому, что эта фраза была употреблена в начале текста – в первой сотне слов, в отличие от других статей данного раздела, в которых подобное сочетание находилось в дальнейшем тексте, за пределами начального отрывка в сто слов.

Отсюда следует вывод, что наличие ключевых слов в начале текста способствует лучшему ранжированию страницы.

Согласно рекомендациям программы, содержимое тегов <title> было изменено там, где ключевых слов было недостаточно, а также приведено к виду «первичное ключевое слово - вторичное ключевое слово» на тех страницах, где это являлось уместным. Путем изменения содержимого данного тега было также обеспечено неполное пересечение с содержимым тегов <h1>. Так страницу с заголовком «Астрологические консультации - расчет гороскопа рождения», который пересекается с содержимым <h1> по словам "расчет гороскопа рождения", поможет странице быть найденной по этому сочетанию слов.

Программа показала, что URL страниц уровня вложенности 2 подразумевает уровень вложенности 1, но изменение URL требует установки редиректа 301 со старого адреса на новый во избежание ошибок при повторном обходе сайта поисковыми роботами, а также потери ссылочного веса. В связи с этим в файл .htaccess, находящийся в корневой папке сайта, были добавлены строки следующего типа для каждого URL, которому требовалось изменение:

RewriteRule ^persona012.html /horoscopes/zodiak-245.html [R=301,L]

От данного изменения не ожидается серьёзных повышений в выдаче ввиду незначительного влияния URL на ранжирование, однако в сочетании с изменением заголовков может закрепить результат.

Результат программы показал, что ни на одной странице не используется разметка Schema.org. В разделе контактов указан телефон, адреса электронной почты и Skype. Эти данные можно разметить при помощи атрибутов telephone и email, относящихся к виду данных Person. К сожалению, Skype не поддерживается таким атрибутом как sameAs, который может использоваться для профилей в социальных сетях. Также, в связи с наличием оригинальных статей, где указан и автор, и дата публикации, эти статьи также можно подвергнуть разметке при помощи атрибутов вида данных Article, такими как articleBody, author, datePublished. С использованием атрибутов name и price вида данных Offer можно разметить данные на странице, посвященной расценкам на оказываемые услуги.

Открыв сайт с мобильного устройства, он был отображен так, как показано на Рис. 10:

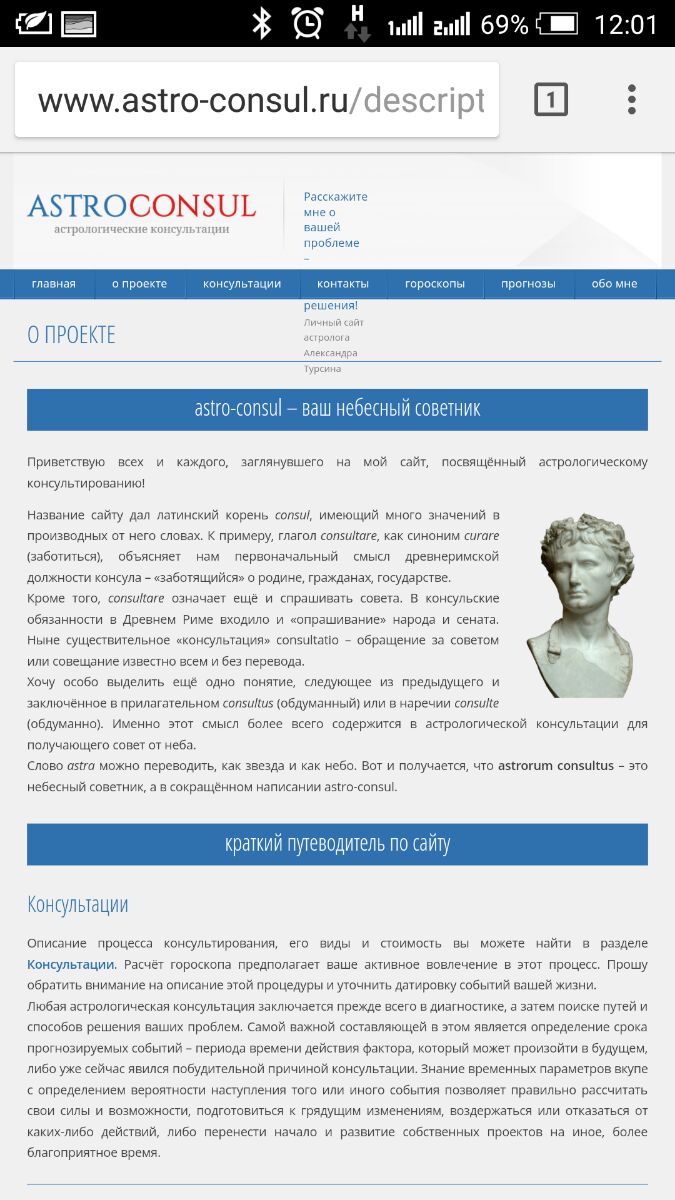


Рис. 10. Некорректное отображение текста верхней секции сайта на мобильном устройстве.

Несмотря на наличие корректно прописанного мета-тега viewport, текст верхней секции отображался некорректно. Были предприняты временные меры путём ручной настройки ширины проблемных секций, что исправило проблему (Рис. 11), однако верстка по-прежнему не является адаптивной к любому экрану.

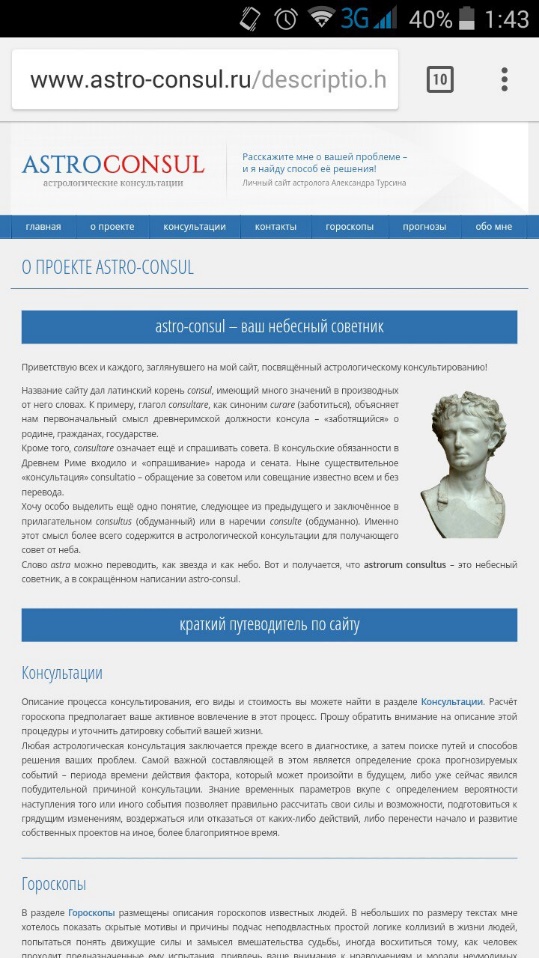


Рис. 11. Отображение сайта на мобильном устройстве после настройки ширины проблемной секции.

Ввиду большого количества слов в мета-тегах keywords на некоторых страницах, было решено сократить их содержимое до трёх наиболее важных слов или фраз, относящихся к содержимому страницы. От этого изменения не ожидается никаких результатов, однако этим самым можно обезопасить страницы от подозрений со стороны поисковых систем.

Со слов автора сайта, некоторые изображения носят дизайнерский характер, однако они находятся в HTML-коде и соответствуют теме сайта, в связи с чем необходимо заполнить атрибут alt. Были добавлены тексты для атрибута у тех изображений, где, по показаниям программы, данный атрибут был пустым. Таким образом к содержимому сайта было добавлено несколько новых тематических терминов, что позволит выдать сайт по несколько большему кругу запросов, пусть и не относящихся к основной цели сайта.

4.4. Результаты

В сравнении с результатами, отображенными на Рис. 9, после проведенной работы за короткий промежуток времени позиции сайта по релевантным ему запросам возросли незначительно. Более того, ввиду изменений, связанных с добавлением информации об оказании услуги по расчету гороскопа рождения, в статистике появился соответствующий запрос («расчет гороскопа»).

Столь незначительные изменения в статистике связаны с тем, что на данный момент у сайта пока невысокая посещаемость, но уже наметилась тенденция к ее увеличению. Так как работа с сайтом активно продолжается, то следует ожидать дальнейших улучшений в позициях.

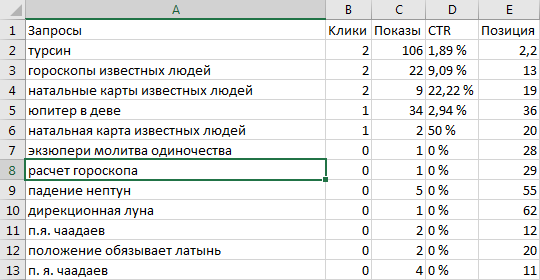


Рис. 12. Статистика по запросам из Google Search Console после проведенных работ.

Выводы

Программа выполняет поставленную задачу по анализу сайта и выдаче рекомендаций по исправлению найденных проблем. Основываясь на результатах ее работы, владелец сайта может внести изменения во внутреннюю составляющую сайта, причем на свое усмотрение, причем результаты могут быть как выше ожидаемого, так и ниже. В связи с этим работа над сайтом должна вестись постоянно.

Полученные результаты не в полной мере оправдали ожиданий ввиду того, что изменение факторов не оказывает моментального влияния и требуют времени, особенно при невысокой посещаемости.

Был выработан план по дальнейшему развитию сайта и улучшению его структуры, куда входит создание HTML-карты сайта и страницы ошибки 404, продолжение мер по мобильной оптимизации сайта, внешняя оптимизация, а также наполнение его контентом посредством публикации разнообразных циклов статей, связанных с основной темой сайта. Также выяснилось, что система управления содержимым данного сайта не поддерживает атрибуты, необходимые для внедрения микроразметки Schema.org, поэтому потребуется использование сторонних программных продуктов для обеспечения данных функций.

Заключение

В процессе работы с данным, ранее остановившимся в развитии, сайтом, у владельца сайта проявилась заинтересованность в дальнейшем его продвижении, появились планы по насыщению контентом и извлечению коммерческой выгоды, что в конечном итоге должно привести к появлению более информативного и привлекательного для различных категорий пользователей сети ресурса.

В связи с этим дальнейшее развитие получит программа по анализу сайта. В планах увеличение количества рассматриваемых факторов, улучшение предлагаемых рекомендаций, написание удобного пользовательского интерфейса, а также оптимизация кода с целью ускорения работы и возможности применения ее на более крупных сайтах с различной структурой. Одной из конечных целей развития программы может стать использование ее владельцами других сайтов с целью улучшения позиций в поисковой выдаче, что является положительным моментом для коммерческих сайтов, и позволит владельцу сайта увеличить прибыль путем привлечения большего количества клиентов, при этом минимизируя расходы на оптимизацию сайта, внося изменения во внутреннюю составляющую сайта самостоятельно, следуя рекомендациям, либо привлекая стороннего специалиста, но давая ему конкретное техническое задание, не переплачивая за лишнюю работу.

В перспективе создание продукта, который будет помогать мне при работе над другими сайтами, а также другим специалистам в повседневной работе, что может стать основой для магистерской работы.

Список литературы

1. How Search Engine Works - The Beginners Guide to SEO. https://moz.com/beginners-guide-to-seo/how-search-engines-operate

2. Робот Googlebot. https://support.google.com/webmasters/answer/182072?hl=ru

3. Принцип работы поисковой системы. https://devaka.ru/articles/how-search-engines-work

4. Robots.txt and Meta Robots. https://moz.com/learn/seo/robotstxt

5. Использование robots.txt. https://yandex.ru/support/webmaster/controlling-robot/robots-txt.xml

6. Инвертированный файл. http://wiki.liveinternet.ru/IR/InvertirovannyjjFajjl

7. Ранжирование сайта. Внешние и внутренние факторы. http://pr-cy.ru/lib/seo/Ranzhirovanie-sayta-Vneshnie-i-vnutrennie-faktory

8. Search Engine Ranking Factors 2015. https://moz.com/search-ranking-factors

9. On-Page Ranking Factors - SEO Best Practices. https://moz.com/learn/seo/on-page-factors

10. On-Site SEO. https://moz.com/learn/seo/on-site-seo

11. More than Keywords: 7 Concepts of Advanced On-Page SEO. https://moz.com/blog/7-advanced-seo-concepts

12. Классификация поисковых запросов. https://seo-akademiya.com/baza-znanij/podbor-zaprosov/klassifikacija-poiskovyh-zaprosov/

13. Оптимальная плотность ключевых слов в тексте. https://seo-akademiya.com/baza-znanij/kontent/optimalnaya-plotnost-klyuchevyix-slov-v-tekste/

14. Title Tag. https://moz.com/learn/seo/title-tag

15. Заголовки H1 и TITLE. Делать ли их одинаковыми или разными? https://devaka.ru/articles/title-and-h1

16. SEO Best Practices for URLs. https://moz.com/learn/seo/url

17. Что такое "вес ссылки"? http://webjunk.ru/2009/06/chto-takoe-ves-ssilki/

18. 301 редирект - самое полное руководство. https://seoprofy.ua/blog/optimizaciya-sajtov/301-redirekt

19. Internal Links - SEO Best Practices. https://moz.com/learn/seo/internal-link

20. Что такое внутренняя перелинковка. https://seo-akademiya.com/seo-wiki/vnutrennjaja-perelinkovka/

21. Что такое карта сайта. https://seo-akademiya.com/seo-wiki/karta-sajta/

22. Schema.org Structured Data. https://moz.com/learn/seo/schema-structured-data

23. Schema.org. https://yandex.ru/support/webmaster/schema-org/what-is-schema-org.xml

24. Person - schema.org. https://schema.org/Person

25. Meta tags that Google understands. https://support.google.com/webmasters/answer/79812?hl=en

26. The Wonderful World of SEO Meta Tags. https://moz.com/blog/seo-meta-tags

27. Использование HTML-элементов. https://yandex.ru/support/webmaster/controlling-robot/html.xml?lang=ru

28. SEO Meta Tags Best Practices. https://www.hobo-web.co.uk/definitive-guide-to-using-important-meta-tags/

29. Alt Text. https://moz.com/learn/seo/alt-text

30. HTTP Status Codes SEO Best Practices. https://moz.com/learn/seo/http-status-codes

31. Optimizing for Mobile - SEO Best Practices. https://moz.com/learn/seo/mobile-optimization

32. Off-Site SEO. https://moz.com/learn/seo/off-site-seo

33. Requests: HTTP for Humans. http://docs.python-requests.org/en/master/#

34. lxml.html. http://lxml.de/lxmlhtml.html

35. Korobov M.: Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, P. 320-332 (2015).

36. Natural Language Toolkit. <http://www.nltk.org/index.html>

Приложение

Приложение 1. Код программы.

**import** requests  
**import** lxml.html  
**import** re  
**from** nltk.corpus **import** stopwords  
**import** pymorphy2  
**from** collections **import** Counter  
  
sw = stopwords.words(**'russian'**) + [**'неё'**, **'её'**, **'всё'**]  
site = **'http://astro-consul.ru/'**link\_history = [site]  
print(**'Сайт:'**, site)  
response = requests.get(site)  
**if** response.history:  
 **for** i **in** response.history:  
 print(**'Обнаружен редирект с кодом'**, i.status\_code)  
 print(**'Конечный URL:'**, response.url)  
**if** response.url **not in** link\_history:  
 link\_history.append(response.url)  
 site = response.url  
print(**'Сбор ссылок на HTML-страницы...'**)  
tree = lxml.html.fromstring(response.text)  
links = tree.xpath(**'/html//a/@href'**)  
links\_set = list(set(links))  
links\_set = [x **for** x **in** links\_set **if** x != **''**]  
links\_set = [x **for** x **in** links\_set **if '?' not in** x]  
**for** k **in** range(len(links\_set)):  
 **if** links\_set[k] **not in** link\_history:  
 links\_set[k] = site + links\_set[k]  
depth = 1  
linkdepth = dict.fromkeys(links\_set, depth)  
newlinks = []  
**while** (links\_set):  
 depth += 1  
 **for** i **in** links\_set:  
 **if** (i **not in** link\_history):  
 response = requests.get(i)  
 content\_type = response.headers.get(**'Content-Type'**)  
 **if 'text/html' not in** content\_type:  
 **continue** tree = lxml.html.fromstring(response.text)  
 **if** response.url **not in** link\_history:  
 link\_history.append(response.url)  
 links2 = tree.xpath(**'/html//a/@href'**)  
 links\_set2 = list(set(links2))  
 links\_set2 = [x **for** x **in** links\_set2 **if** x != **''**]  
 links\_set2 = [x **for** x **in** links\_set2 **if '?' not in** x]  
 **for** k **in** range(len(links\_set2)):  
 **if** links\_set2[k] **not in** link\_history:  
 links\_set2[k] = site + links\_set2[k]  
 linksect = list(set(links\_set).intersection(set(links\_set2)))  
 linkdif = list(set(links\_set2) - set(linksect))  
 linkdif = [x **for** x **in** linkdif **if** x **not in** link\_history]  
 **if** linkdif:  
 newlinks = newlinks + linkdif  
 newdepth = dict.fromkeys(newlinks, depth)  
 linkdepth.update(newdepth)  
 links\_set = newlinks  
 newlinks = []  
html\_links = []  
**for** i **in** link\_history:  
 response = requests.get(i)  
 content\_type = response.headers.get(**'Content-Type'**)  
 **if** (**'text/html' in** content\_type) **and** (response.history == []):  
 html\_links.append(i)  
print(**'Найдено'**, len(html\_links), **'HTML-страниц:'**)  
print(html\_links)  
print(**'Максимальный уровень вложенности страниц (количество кликов от главной страницы):'**, max(linkdepth.values()))  
htmlmap = **False  
for** i **in** html\_links:  
 response = requests.get(i)  
 tree = lxml.html.fromstring(response.text)  
 links = tree.xpath(**'/html//a/@href'**)  
 links\_set = list(set(links))  
 links\_set = [x **for** x **in** links\_set **if** x != **''**]  
 links\_set = [x **for** x **in** links\_set **if '?' not in** x]  
 **for** k **in** range(len(links\_set)):  
 **if** links\_set[k] **not in** link\_history:  
 links\_set[k] = site + links\_set[k]  
 **if** links\_set == html\_links:  
 print(**'Имеется HTML-карта сайта:'**, i)  
 htmlmap = **True  
 break  
if** htmlmap == **False**:  
 print(**'На сайте отсутствует HTML-карта сайта'**)  
print(**'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**)  
**for** i **in** html\_links:  
 response = requests.get(i)  
 print(i)  
 print(**'Код состояния:'**, response.status\_code)  
 print(**'Уровень вложенности:'**, linkdepth.get(i))  
 struct\_test = re.sub(**'\.html$'**, **''**, i)  
 struct\_test = re.sub(site, **''**, struct\_test)  
 struct\_test = struct\_test.split(**'/'**)  
 struct\_test = [x **for** x **in** struct\_test **if** x != **''**]  
 **if** len(struct\_test) == linkdepth.get(i):  
 print(**'Уровень вложенности отражен в URL.'**)  
 **else**:  
 print(**'Уровень вложенности некорректно отражен в URL. Подразумевается:'**, len(struct\_test), **'- Фактически:'**,  
 linkdepth.get(i))  
 tree = lxml.html.fromstring(response.text)  
 text = tree.xpath(**'/html/body//text()'**)  
 text = **''**.join(text)  
 text = text.lower()  
 text = text.translate({ord(o): **None for** o **in** (**'!"#$%&\'()\*+,-./:;<=>?@[\]^\_`{|}~–©•'**)})  
 text = text.split()  
 normal\_text = [pymorphy2.MorphAnalyzer().parse(word)[0].normal\_form **for** word **in** text]  
 normal\_text = [x **for** x **in** normal\_text **if** x **not in** sw]  
 word\_percent = Counter(normal\_text)  
 **for** key, value **in** word\_percent.items():  
 word\_percent[key] = round(((value / len(text)) \* 100), 2)  
 print(**'10 самых употребляемых слов:'**, word\_percent.most\_common(10))  
 top\_words = []  
 okay\_percentage = []  
 overused = []  
 **for** key, value **in** word\_percent.most\_common(10):  
 top\_words.append(key)  
 **if** 1.5 <= value <= 5:  
 okay\_percentage.append(key)  
 **if** value > 5:  
 overused.append(key)  
 **if** okay\_percentage:  
 print(**'Из десяти наиболее употребляемых слов,'**, len(okay\_percentage), **'имеют плотность от 1.5% до 5%:'**,  
 okay\_percentage, **' - потому они претендуют на звание ключевых'**)  
 **else**:  
 **if not** overused:  
 print(**'На странице нет слов с плотностью выше 1.5%. Необходимо повысить плотность ключевых слов.'**)  
 **if** overused:  
 print(**'Следующие слова имеют плотность выше 5%:'**, overused,  
 **'- Если данные слова являются ключевыми, необходимо уменьшить их плотность.'**)  
 titletags = tree.xpath(**'/html//title/text()'**)  
 h1s = tree.xpath(**'/html//h1/text()'**)  
 **if** len(titletags) > 1:  
 print(**'Используется более одного тега <title>. Поисковыми системами и браузерами будет использоваться первый.'**)  
 print(**'Title:'**, titletags[0])  
 print(**'<h1>:'**, h1s)  
 titletag = titletags[0]  
 titletag = titletag.translate({ord(o): **None for** o **in** (**'!"#$%&\'()\*+,-./:;<=>?@[\]^\_`{|}~–©•'**)})  
 **for** h **in** range(len(h1s)):  
 h1s[h] = h1s[h].translate({ord(o): **None for** o **in** (**'!"#$%&\'()\*+,-./:;<=>?@[\]^\_`{|}~–©•'**)})  
 h1s[h] = [pymorphy2.MorphAnalyzer().parse(word)[0].normal\_form **for** word **in** h1s[h].lower().split()]  
 h1s[h] = [x **for** x **in** h1s[h] **if** x **not in** sw]  
 norm\_title = [pymorphy2.MorphAnalyzer().parse(word)[0].normal\_form **for** word **in** titletag.lower().split()]  
 norm\_title = [x **for** x **in** norm\_title **if** x **not in** sw]  
 norm\_h1s = h1s  
 **if** len(set(norm\_title).intersection(top\_words)) > 0:  
 **if** 3 <= len(set(norm\_title).intersection(top\_words)) <= 5:  
 print(**'Title содержит 3-5 часто употребляемых в тексте страницы слов.'**)  
 **if** 1 <= len(set(norm\_title).intersection(top\_words)) <= 2:  
 print(  
 **'Title содержит 1-2 часто употребляемых в тексте страницы слова. Рекомендуется увеличить это количество на 2-3 слова.'**)  
 **if** len(set(norm\_title).intersection(top\_words)) > 5:  
 print(**'Title содержит более 5 ключевых слов. Рекомендуется уменьшить это количество.'**)  
 **else**:  
 print(**'Заголовок не содержит часто употребляемых в тексте страницы слов.'**)  
 **for** j **in** range(len(norm\_h1s)):  
 **if** len(set(norm\_title).intersection(norm\_h1s[j])) == 0:  
 print(**'Title и '**+str(j+1)+**'-й H1 не пересекаются. Следует обеспечить пересечение по важным ключевым словам.'**)  
 **if** norm\_title == norm\_h1s[j]:  
 print(**'Title и '**+str(j+1)+**'-й H1 совпадают. Следует сделать их различными, обеспечив пересечение по важным ключевым словам.'**)  
 schemas = tree.xpath(**'/html//@itemtype'**)  
 schema\_usage = **False  
 if** schemas:  
 **for** k **in** range(len(schemas)):  
 schemas[k].lower()  
 **if 'http://schema.org/' in** schemas[k]:  
 schema\_usage = **True** schema\_type = re.sub(**'http://schema.org/'**, **''**, schemas[k])  
 print(**'На данной странице есть использование Schema.org. Вид данных:'**, schema\_type)  
 **if** schema\_usage == **False**:  
 print(**'Атрибуты Schema.org не используются на данной странице'**)  
 metatags = tree.xpath(**'/html/head/meta'**)  
 ctype\_exists = **False** desc\_exists = **False** keyw\_exists = **False** vport\_exists = **False  
 for** m **in** range(len(metatags)):  
 metahttp = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@http-equiv'**)  
 **if 'content-type' in** metahttp:  
 ctype\_exists = **True** metactype = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@content'**)  
 metactype = **''**.join(metactype)  
 **if 'charset' in** metactype:  
 print(**'Мета Content-Type: content="'** + metactype + **'"'**)  
 **if** metactype == **''**:  
 print(**'Мета-тег Content-Type не заполнен.'**)  
 **for** m **in** range(len(metatags)):  
 metaname = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@name'**)  
 **if 'viewport' in** metaname:  
 vport\_exists = **True** metaview = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@content'**)  
 metaview = **''**.join(metaview)  
 **if** metaview == **''**:  
 print(  
 **'Мета-тег viewport не заполнен. Необходимо заполнить его по типу content="width=device-width" с целью улучшения страницы отображения на мобильных устройствах.'**)  
 **if 'width=' in** metaview:  
 print(**'Meta Viewport: content="'** + metaview + **'"'**)  
 **if 'description' in** metaname:  
 desc\_exists = **True** metadesc = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@content'**)  
 metadesc = **''**.join(metadesc)  
 **if** metadesc == **''**:  
 print(**'Незаполненный мета-тег description'**)  
 **else**:  
 print(**'Meta Description:'**, metadesc)  
 **if 'keywords' in** metaname:  
 keyw\_exists = **True** metakeys = tree.xpath(**'/html/head/meta['** + str(m + 1) + **']/@content'**)  
 metakeys = **''**.join(metakeys)  
 **if** metakeys == **''**:  
 print(**'Незаполненный мета-тег keywords'**)  
 **else**:  
 print(**'Meta Keywords:'**, metakeys)  
 **if** len(metakeys.replace(**','**, **''**).split()) > 6:  
 print(  
 **'Рекомендуется уменьшить количество слов в мета-теге keywords во избежание потенциальных санкций'**)  
 **if** desc\_exists == **False**:  
 print(  
 **'Мета-тег description отсутствует. Рекомендуется добавить описание страницы для отображения его на страницах поисковой выдачи.'**)  
 **if** keyw\_exists == **False**:  
 print(  
 **'Мета-тег keywords отсутствует. Рекомендуется добавить ввиду использования его некоторыми поисковыми системами.'**)  
 **if** vport\_exists == **False**:  
 print(  
 **'Мета-тег viewport отсутствует. Рекомендуется добавить для корректного отображения страницы на мобильных устройствах.'**)  
 **if** ctype\_exists == **False**:  
 print(  
 **'Мета-тег Content-Type, необходимый для определения кодировки, отсутствует. Рекомендуется добавить во избежание проблем с отображением страницы.'**)  
 alts = tree.xpath(**'/html//@alt'**)  
 empty\_alts = **False  
 if** alts:  
 print(**'Содержимое атрибутов alt:'**, alts)  
 **for** l **in** range(len(alts)):  
 **if** alts[l] == **''**:  
 empty\_alts = **True  
 if** empty\_alts == **True**:  
 print(**'Обнаружены незаполненные атрибуты alt. Рекомендуется их заполнить.'**)  
 print(**'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n'**)