

Разработка плагина оглавления для статей на сайте

Круглик Роман Игоревич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье рассматривается поэтапно создание мини-плагина для автоматической генерации оглавления с помощью JavaScript. В качестве примера реализовано просто оглавления для статьи.

Ключевые слова: скрипт, JavaScript, оглавление на сайте.

Developing a table of contents plugin for articles on a site

Kruglik Roman Igorevich

Sholom-Aleichem Priamursky State University

Student

Abstract

In article discusses the phased creation of a mini-plugin for automatic generation of table of contents using JavaScript. As an example, just the table of contents for the article is implemented.

Keywords: script, JavaScript, table of contents on the site.

Интернет развивается довольно стремительно. Быстро растет количество изданий, посвященных Сети, что предвещает широкое ее распространение даже в далеких от техники областях. Интернет превращается из большой игрушки для интеллектуалов в полноценный источник разнообразной полезной информации для любой категории пользователей.

Люди заходят на сайты и блоги обычно с целью поиска какой-то конкретной информации, конкретных ответов на свои вопросы. Иногда статьи занимают огромное пространство на странице и человеку сложно разобраться и найти нужные ему ответы. Поэтому вначале статьи можно увидеть содержание, с помощью которого пользователь перемещается по статье и быстрее находит нужную ему информацию. Обычно содержания делают в виде активных ссылок по статье или обзору, это помогает, не тратя большого количества времени найти интересующую посетителям часть статьи и быстро перейти на неё.

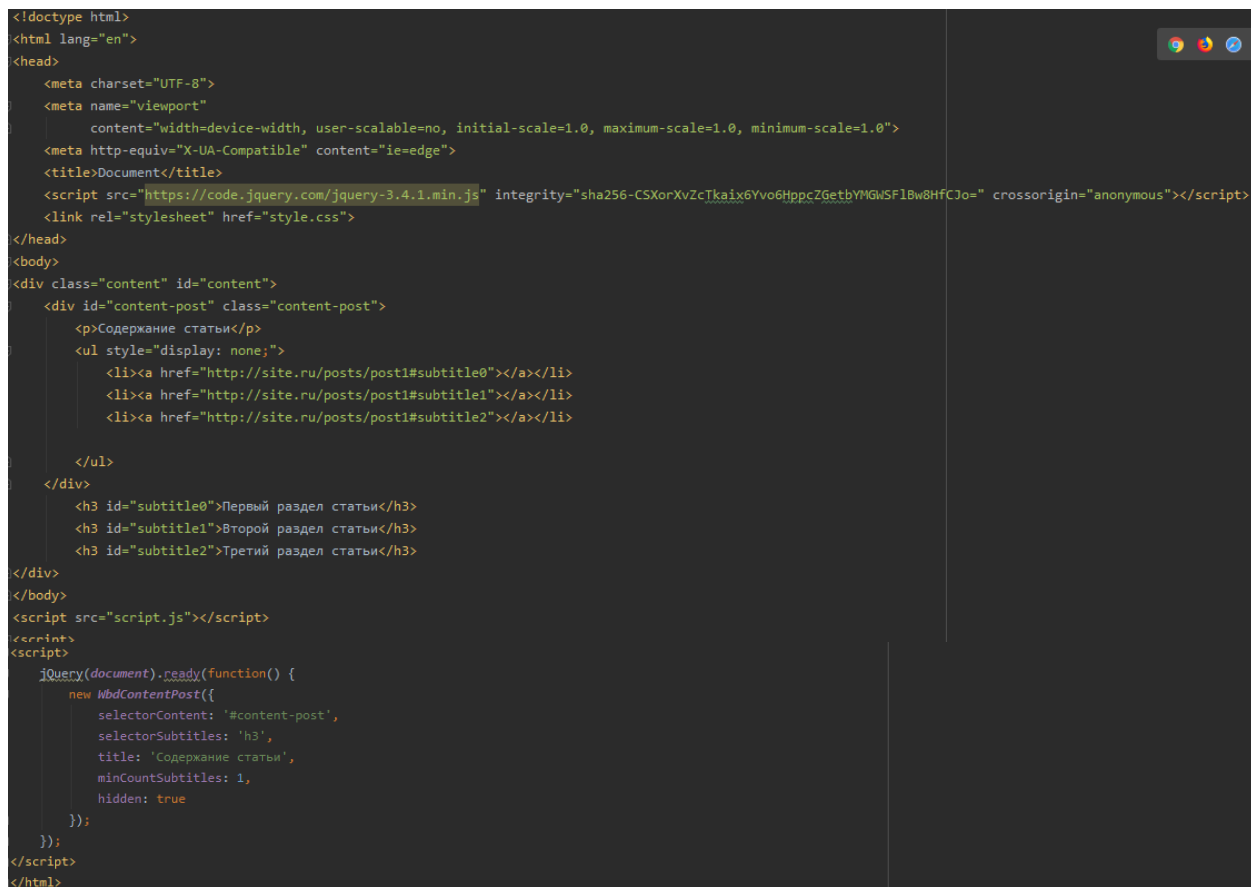
Активные ссылки содержания работают только внутри статьи (поста) блога. Место, куда должно перебросить пользователя определяется создателем.

Исследования в области создания плагина оглавления для статей актуальны и по сей день. На ресурсе [1] рассматривается один из способов создания оглавления (содержания) на сайте. В статье [2] на блоге можно увидеть реализацию содержания для сайта/блога с помощью только HTML5. На ресурсе [3] пошагово рассмотрено создание автоматической генерации содержания для статьи.

В данной статье будет рассмотрен плагин для автоматической генерации оглавления. Так же реализована:

- возможность скрывать оглавление, клик на который разворачивает/скрывает содержание.
- возможность не выводить оглавление, если в статье мало разделов
- полная кастомизация оглавления.

Для начала рассмотрим главный файл с разметкой, который включает в себя оглавление и сами разделы, на которые будем переходить (см. рис. 1).



```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="https://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGKfsSI5E7HC3o=" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="content" id="content">
  <div id="content-post" class="content-post">
    <p>Содержание статьи</p>
    <ul style="display: none;">
      <li><a href="http://site.ru/posts/post1#subtitle0"></a></li>
      <li><a href="http://site.ru/posts/post1#subtitle1"></a></li>
      <li><a href="http://site.ru/posts/post1#subtitle2"></a></li>
    </ul>
  </div>
  <h3 id="subtitle0">Первый раздел статьи</h3>
  <h3 id="subtitle1">Второй раздел статьи</h3>
  <h3 id="subtitle2">Третий раздел статьи</h3>
</div>
</body>
<script src="script.js"></script>
<script>
  jQuery(document).ready(function() {
    new WbdContentPost({
      selectorContent: '#content-post',
      selectorSubtitles: 'h3',
      title: 'Содержание статьи',
      minCountSubtitles: 1,
      hidden: true
    });
  });
</script>
</html>
```

Рисунок 1. Файл index.html

Помимо разметки в конце файла подключаем сам скрипт, который является динамичным и не изменяется при подключении к другим веб-ресурсам. Для корректной работы скрипта необходимо обозначить дополнительные настройки. Такая настройка выставляется всего 1 раз и распространяется на всю систему. Разберём каждый атрибут отдельно и зачем он нужен:

1. selectorContent – блок, в котором находится оглавление

2. selectorSubtitles – заголовки вне оглавлениям
3. title – название оглавления
4. minCountSubtitles - минимальное количество заголовков, при котором строится оглавление. Если меньше, оглавление показываться не будет.
5. hidden - показывать содержание раскрытым или скрывать под спойлер.

Далее рассмотрим скрипт, который подключается и настраивается в главном файле. (см. рис. 2).

```
var WbdContentPost = (function($) {  
    var Content = function(options) {  
        this.ui = {  
            $content: $(options.selectorContent),  
            $subtitles: $(options.selectorSubtitles)  
        };  
        this.title = options.title || 'Содержание статьи';  
        this.minCount = options.minCountSubtitles || 1000;  
        this.hidden = options.hidden || false;  
  
        this.init();  
    }  
    Content.prototype.init = function() {  
        if (!this.ui.$content.length || this.ui.$subtitles.length < this.minCount) {  
            this.ui.$content.remove();  
            return false;  
        }  
        var contentHtml = '',  
            url = document.location.href.split( separator: '#' )[0],  
            display = this.hidden ? 'none' : 'block';  
        this.ui.$subtitles.each(function(n, item) {  
            var anchor = '<a id="subtitle" + n + "></a>',  
                $item = $(item),  
                subtitle = '<li><a href="' + url + '#subtitle' + n + '>' + $item.text() + '</a></li>';  
  
            $item.before(anchor);  
            contentHtml += subtitle;  
        });  
        this.ui.$content.html('<ul style="display:' + display + '>' + contentHtml + '</ul>');  
        this.ui.$content.prepend( nodes: '<p>' + this.title + '</p>');  
        if (this.hidden) {  
            this.ui.$content.find('p').on('click', function() {  
                $(this).next().slideToggle(300);  
            });  
        }  
    }  
    return Content;  
})(jQuery);
```

Рисунок 2. Плагин

Так же нужны стили для самого оглавления (см. рис. 3).

```
.content-post {  
  padding: 20px 0;  
}  
.content-post p {  
  display: inline;  
  font-weight: 700;  
  font-size: 1.1em;  
  color: steelblue;  
  margin-bottom: 5px;  
  border-bottom: dotted 1px;  
  cursor: pointer;  
}  
.content-post ul {  
  margin-top: 20px;  
}  
.content-post li {  
  list-style: none;  
  padding: 1px 0;  
}  
.content-post li a {  
  border-bottom: dotted 1px;  
  text-decoration: none;  
}
```

Рисунок 3. Файл стилей

Теперь можно посмотреть на результат работы плагина (см. рис. 4).

Содержание статьи

Первый раздел статьи

Второй раздел статьи

Третий раздел статьи

Первый раздел статьи

Второй раздел статьи

Третий раздел статьи

Рисунок 4. Файл стилей

При нажатии на оглавление пользователя будет отправлять туда, куда ему нужно. Так же при нажатии на «Содержание статьи» блок будет скрывается и может быть открыт при повторном нажатии.

В результате представленный в статье мини-плагин автоматически генерирует оглавление и вешает обработчики в нужные места на сайте, тем самым в разы уменьшая время на написание статьи и упрощается навигацию для пользователей.

Библиографический список

1. Простые способы создания оглавления (содержания) на сайте // webkato URL: <https://webkato.ru/prostye-sposoby-sozdanija-oglavlenija-soderzhanija-na-sajte/> (дата обращения: 15.01.2020).
2. Как сделать содержание (оглавление) в статье сайта/блога HTML5 // Шпаргалка блоггера URL: <http://shpargalkablog.ru/2014/03/table-of-contents-html5.html> (дата обращения: 15.01.2020).
3. Автоматическая генерация содержания для статьи // Шпаргалка блоггера URL: <https://ruseller.com/lessons.php?rub=32&id=767> (дата обращения: 15.01.2020).