

» Статьи » Разработка » Делаем подгрузку контента при прокрутке страницы

Инструменты

Заработок

Раскрутка

+ 2 Рубрики

16 ноября 2014 · Антон Кулешов

## Делаем подгрузку контента при прокрутке страницы

В настоящее время на многих сайтах можно увидеть динамически подгружаемый контент при прокрутке страницы. Этот эффект стал одной из хороших альтернатив обычной постраничной навигации. Хотя и у него есть свои недостатки, но удобство этого способа их компенсирует. Из этой статьи вы узнаете, как реализовать такой функционал у себя в проекте используя jQuery плагин jScroll.

Подключаем jQuery библиотеку и скрипт:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="jquery.jscroll.min.js" type="text/javascript"></script>
```

Теперь нам необходимо добавить HTML разметку:

```
<div id="content">
  <div class="next">
    <a href="путь_до_обработчика">next</a>
  </div>
</div>
```

И наконец, применяем плагин к блоку с индикатором content:

```
$("#content").jscroll();
```

У jScroll есть свои настройки:

- debug – включает возможность отслеживать работу скрипта через консоль, принимает значения true или false (по умолчанию false);
- autoTrigger – отвечает за автоподгрузку контента, принимает значения true или false. По умолчанию установлено true, но если указать false автоподгрузка будет отключена, а вместо неё появится кнопка;
- autoTriggerUntil – указывает сколько раз подгружать контент автоматически (по умолчанию false);
- loadingHtml – в этом параметре мы можем поместить HTML код, который будет в место «Loading...» (по умолчанию «Loading...»);
- padding – отступ до подгружаемого контента. Данный параметр работает, когда в значение autoTrigger установлено true (по умолчанию 0);
- nextSelector – указывается селектор поиска пути обработчика (по умолчанию «a:last»);
- contentSelector – указывается селектор для загрузки части контента (по умолчанию «»);
- callback – функция вызывается в конце каждой загрузки данных. Внутри её передаётся jQuery объект HTML

элемента, в котором находится загруженный контент (по умолчанию false).

Завершая рассматривать подключение плагина и его настройки, важно упомянуть, что передача данных на сервер осуществляется при помощи метода GET.

## Пример работы с плагином

После того, как мы рассмотрели основные моменты, связанные с плагином, можно приступить к реализации примера. Мы сделаем скрипт, который при первом открытии страницы будет грузить определенное число записей, а при скроллинге уже догружать остальные.

HTML:

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8" />
  <title>jQuery jScroll</title>
  <link href="main.css" type="text/css" rel="stylesheet" />
  <script src="jquery.lib.js" type="text/javascript"></script>
  <script src="jquery.jscroll.min.js" type="text/javascript"></script>
</head>
</head>
<body>
  <div id="content">
  </div>
</body>
</html>
```

CSS:

```
*{
  padding: 0;
  margin: 0;
  font-family: Tahoma;
}
#content{
  font-size: 14px;
  width: 60%;
  margin: 0 auto;
  border: 1px solid #ccc;
  padding: 0 7px;
}
  #content p{
    margin: 10px;
  }
  #content p span:first-child{
    font-size: 20px;
    font-weight: bold;
    padding: 10px;
  }
  .jscroll-loading{
    text-align: center;
    padding: 7px 0;
  }
  .next a{
    padding: 7px 5px;
    color: #000;
    border-radius: 5px;
    display: block;
    border: 1px solid #ccc;
    margin: 0 auto 8px;
    text-decoration: none;
    width: 30px;
  }
  .next a:hover{
    border: 1px solid #54A7EB;
    color: #54A7EB;
  }
}
```

jQuery код:

```

$(function(){

    var content = $("#content"),
        loading = "<img src='loading.gif' alt='Loading' />";

    // Подгрузка первых записей
    $.ajax({
        url: "jscroll.php",
        dataType: "json",
        type: "GET",
        data: { type: "start" },
        success: function(data){

            if(data){

                content.append(data);
                content.find(".jscroll-loading:first").slideUp(700, function(){

                    $(this).remove();
                });

                // Вызываем плагин
                $("#content").jscroll({
                    autoTriggerUntil: 2,
                    loadingHtml: loading
                });
            }
        },
        beforeSend: function(){

            content.append("<div class='jscroll-loading'>" + loading + "</div>");
        }
    });
});

```

- Помещаем весь скрипт в jQuery обёртку;
- Определяем две переменных: впервой будет находиться результат выборки, во второй HTML код loading;
- После загрузки страницы обращаемся к файлу jscroll.php с параметром type равным start и добавляем на страницу loading;
- Если пришёл ответ, то добавляем его на страницу и удаляем loading;
- Вызываем плагин.

Переходим к серверу, тут у нас будет чуть больше кода. Я не стал выносить функции в отдельный файл, чтобы всё было максимально компактно.

Вверху файла описываем настройки сервера: \$start\_count\_posts (изначальное число подгружаемых записей), \$loading\_count\_posts (число записей подгружаемых при прокрутке), \$path (путь до обработчика) и \$text (записи у нас будут отличаться только своими номерами, так как это пример).

```

$start_count_posts = 7;
$loading_count_posts = 10;
$path = "jscroll.php";

$text = "Равным образом ... прогрессивного развития.";

```

Далее мы добавим вспомогательные функции: is\_ajax(), is\_get() и template\_post().

```

function is_ajax(){

    if(isset($_SERVER["HTTP_X_REQUESTED_WITH"]) &&
        !empty($_SERVER["HTTP_X_REQUESTED_WITH"]) &&
        strtolower($_SERVER["HTTP_X_REQUESTED_WITH"]) == "xmlhttprequest"){

        return true;
    }

    return false;
}

function is_get(){

    if($_SERVER["REQUEST_METHOD"] == "GET"){

        return true;
    }

    return false;
}

function template_post($num, $text){

    if($num && $text){

        $html .= "

        <p>
            <span>".$num."</span>
            <span>
                ".$text."
            </span>
        </p>
        ";

        return $html;
    }

    return false;
}

```

Теперь нам остаётся написать обработчик запросов, который будет анализировать пришедшие данные на сервер и возвращать результат его работы.

```

// Обработка запросов
if(is_get() && is_ajax()){

    // Подгрузка первых записей
    if(isset($_GET["type"]) && $_GET["type"] === "start"){
        sleep(1);

        for($i = 0; $i <= $start_count_posts; $i++){

            $posts .= template_post($i, $text);
        }

        $posts .= "

        <div class='next'>
            <a href='\".$path.\"?count=\".$loading_count_posts.\">next</a>
        </div>
    ";

        echo(json_encode($posts));
    }

    // Подгрузка записей
    if(isset($_GET["count"])){

        $count = (int) $_GET["count"];

        if($count == 50){

            exit;
        }

        sleep(1);

        for($i = $count + 1; $i <= $count + $loading_count_posts; $i++){

            $posts .= template_post($i, $text);
        }

        $count += $loading_count_posts;

        $posts .= "

        <div class='next'>
            <a href='\".$path.\"?count=\".$count.\">next</a>
        </div>
    ";

        echo($posts);
    }
}

exit;

```

Обработка данных начинается с проверки способа, которым отправляются они на сервер. Если всё удачно, то заходим внутрь и обрабатываем пришедшие данные, иначе вызываем функцию exit. Так как в данном примере мы выводим одну и ту же запись, я умышленно поставил задержки в каждом из случаев по одной секунде. Далее формируем HTML код для ответа и отправляем обратно к пользователю.

Вот и всё, теперь вы сможете легко добавить к себе на сайт jScroll плагин для подгрузки контента.

#AJAX

49 025

4

Самые популярные способы создания HTTP-запросов в JavaScript

1С-Битрикс компонент – AJAX навигация

Ajax-загрузка данных на сервер



**Андрей Чернышов**

29 марта 2015

Спасибо за подробный пример, мне как раз было нужно добавить ленту на сайт и этот плагин меня выручил.



**Gennady Shabalovsky**

10 августа 2016

Я обычный "читатель" таких лент и попал сюда когда искал как с этим бороться. У меня 8 гига памяти и я не всегда могу дойти до конца такой странички. Особенно если это с gifками, видео или соvb-ми.



**Антон Кулешов**

14 августа 2016

С этим бороться нет смысла. Будет лента на сайте или нет решает только владелец сайта.



**Олег Берговин**

29 июня 2018

У нормальных владельцев сайтов есть меню постраничного перехода.  
Не нравится крутить ленту - загружай отдельные страницы через меню.

**Плавная кнопка прокрутки «Наверх»**

**Делаем форму обратной связи на PHP**