**Ajax.**

**Asynchronous Javascript and XML** — асинхронный **JavaScript** и **XML**.

При обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

**Ajax** – это способ взаимодействия с **HTML, JavaScript, CSS.**

Запросы **AJAX** осуществляются с помощью объекта **XMLHttpRequest**, его методов и свойств.

Пример создания объекта: **var per = new XMLHttpRequest();**

**Для работы некоторых примеров должна быть скачана библиотека jQuery.**

У объекта **XMLHttpRequest** есть нужные нам для работы свойства:

**onreadystatechange** – присваиваем функцию, код которой исполнится при получении ответа от сервера.

**readyState** - Узнает состояние готовности сервера.

**responseText** - Ответ сервера в виде строки.

**responseXML** - Ответ сервера в виде **XML**.

**status** - код ответа сервера.

Запросы **GET** и **POST** в **Ajax**.

Запрос типа **GET** в Ajax можно отправить с помощью двух методов объекта **XMLHttpRequest**.

**open(‘GET’, file, true)** – Создаем **Ajax** запрос типа **GET**, указываем путь к файлу, **true** – указывает сделать асинхронный запрос.

**send(‘data’)** – передает запрос на сервер, data используется для **POST** запросов.

В **POST** обязателен заголовок **Content-Type**, содержащий кодировку - **setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')**

**Пособие написано специально для snakeproject.ru, критику, замечания и предложения Вы можете оставить через страничку «контакты» на сайте.**

Давайте познакомимся с примерами:

1. **GET – запрос**

**Без передачи данных файлу.(В файле 'test.txt' напишите несколько любых строк)**

<html>

<head>

<script type='text/javascript'>

function give(){

xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200)

document.getElementById('res').innerHTML=xmlhttp.responseText;

}

xmlhttp.open('GET', 'test.txt', true);

xmlhttp.send();

}

</script>

</head>

<body>

<b><p id='res'>Асинхронно получить данные из файла</p></b>

<input type='button' value='Получить' onclick='give()'/>

</body>

</html>

**С передачей данных файлу.**

<html>

<head>

<script type='text/javascript'>

function run(){

var x=document.getElementById('num1').value;

var y=document.getElementById('num2').value;

var z=document.getElementById('num3').value;

xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200)

document.getElementById('res').innerHTML=xmlhttp.responseText;

}

xmlhttp.open('GET','add.php?x='+x+'&y='+y+'&z='+z,true);

xmlhttp.send();

}

</script>

</head>

<body>

<b><p id='res'>Пример (x+y)\*z. Введите три числа</p></b>

x<input type='number' id='num1' />

y<input type='number' id='num2' />

z<input type='number' id='num3' />

<br /><br />

<input type='button' value='Решить пример' onclick='run()'/>

</body>

</html>

**В add.php напишите следующий код:**

<?php

$x = $\_GET[x];

$y = $\_GET[y];

$z = $\_GET[z];

echo ($x + $y)\*$z;

?>

1. **POST – запрос**

**Без передачи данных файлу.(В файле 'test.txt' напишите несколько любых строк)**

<html>

<head>

<script type='text/javascript'>

function give(){

xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200)

document.getElementById('res').innerHTML=xmlhttp.responseText;

}

xmlhttp.open('POST','test.txt',true);

xmlhttp.send();

}

</script>

</head>

<body>

<b><p id='res'>Асинхронно запросить данные из файла test. </p></b>

<input type='button' value='Получить данные' onclick='give()'/>

</body>

</html>

**С передачей данных файлу.**

<html>

<head>

<script type='text/javascript'>

function start(){

var x=document.getElementById('num1').value;

var y=document.getElementById('num2').value;

var z=document.getElementById('num3').value;

xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200)

document.getElementById('res').innerHTML=xmlhttp.responseText;

}

xmlhttp.open('POST','addpost.php',true);

xmlhttp.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

var str='x='+x+'&y='+y+'&z='+z;

xmlhttp.send(str);

}

</script>

</head>

<body>

<b><p id='res'>Пример (x+y)\*z. Введите три числа:</p></b>

<input type='number1' id='num1' />

<input type='number2' id='num2' />

<input type='number3' id='num3' />

<br />

<input type='button' value='Решить пример' onclick='start()'/>

</body>

</html>

**В addpost.php напишите следующий код:**

<?php

$x = $\_POST[x];

$y = $\_POST[y];

$z = $\_POST[z];

echo ($x + $y)\*$z;

?>

1. **Запрос к XML**

**Рассмотрим пример получения данных из XML файла:**

<html>

<head>

<script type='text/javascript'>

function xml(){

xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200){

**//Данные по первой точке**

var per=xmlhttp.responseXML;

document.getElementById('res1').innerHTML='Адрес точки: '+per.getElementsByTagName('ip-from')[0].childNodes[0].nodeValue;

var dep1=per.getElementsByTagName('response1');

var cont='Траффик: <br />';

for (var i=0;i<dep1.length;i++){

cont+=(i+1)+'. '+dep1[i].childNodes[0].nodeValue+' скачано.<br />';

}

document.getElementById('fin1').innerHTML=cont;

**//Данные по второй точке**

var per=xmlhttp.responseXML;

document.getElementById('res2').innerHTML='Адрес точки: '+per.getElementsByTagName('ip-from')[1].childNodes[0].nodeValue;

var dep2=per.getElementsByTagName('response2');

var cont='Траффик: <br />';

for (var i=0;i<dep2.length;i++){

cont+=(i+1)+'. '+dep2[i].childNodes[0].nodeValue+' скачано.<br />';

}

document.getElementById('fin2').innerHTML=cont;

}

}

**//MathRandom() нужен для избежания кэширования метода GET, если данные в файле** изменятся.

xmlhttp.open('GET','test.xml?x="+Math.random()',true);

xmlhttp.send();

}

</script>

</head>

<body>

<b><p id='res1'>Получить данные первой точки.</p>

<p id='fin1'></p></b>

<b><p id='res2'>Получить данные второй точки.</p>

<p id='fin2'></p></b>

<input type='button' value='Получить данные XML' onclick='xml()'/>

</body>

</html>

**Текст в файле test.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<event date=" 13/Feb/2014:01:01:01 " result="success">

<ip-from>192.168.1.1</ip-from>

<response1>200мб</response1>

<response1>3500мб</response1>

<ip-from>172.16.0.1</ip-from>

<response2>110мб</response2>

<response2>300мб</response2>

</event>

1. **Запросы AJAX с помощью jQuery**

**Синтаксис запроса: $(“Селектор”).load(url, data, function) – где url – обработчик, data – данные(необязательный параметр), function – функция, которая будет вызвана после окончания запроса(необязательный параметр)**

**Обратите внимание, тут мы подключаем библиотеку jQuery.**

**Вызов без параметров:**

<html>

<head>

<script type="text/javascript" src="jquery.js">

</script>

<script type="text/javascript">

$(document).ready(function(){

$("#but").click(function(){

$("#res").load("test.txt");

})

});

</script>

</head>

<body>

<p id="res">Получить данные из файла test.txt.</p>

<input id="but" type="button" value="Метод load" />

</body>

</html>

**Вызов с параметрами(add.php был представлен ранее в примерах)**

<html>

<head>

<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript">

$(document).ready(function(){

$("#but").click(function(){

$("#res").load("add.php","x=4&y=5&z=3");

})

});

</script>

</head>

<body>

<p id="res">(x+y)\*z.</p>

<input id="but" type="button" value="Решить пример" />

</body>

</html>

1. **Реализация поиска**

**Рассмотрим пример интерактивного поиска с помощью PHP и AJAX**

<html>

<head>

<style type='text/css'>

#article

{

list-style-type:none;

padding:0px;

position:absolute;

top:53px;

width:155px;

border-style:solid;

border-width:1px;

cursor: pointer;

}

</style>

<script type='text/javascript'>

function run(value){

**//Передаем данные для поиска**

var xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200)

document.getElementById('res').innerHTML=xmlhttp.responseText;

}

**//Метод encodeURI заменяет символы, соответствующими им UTF-8 escape-последовательностями.**

xmlhttp.open('GET','article.php?search='+encodeURI(value),true);

xmlhttp.send();

}

**//По щелчку на полученном результате добавим его в поле поиска**

function add(content){

document.getElementById('search').value=content;

document.getElementById('res').style.display='none';

}

</script>

</head>

<body>

<p>Поиск статьи:</p>

**<!-- Событие onkeyup возникает в момент отпускания нажатой клавиши -->**

<input id='search' type='text' onkeyup='run(value)'/>

<div id='res'> </div>

</body>

</html>

**Article.php:**

<?php

**//Список статей**

$article = array("SQL: DDL", "SQL: DML", "T-SQL: DDL", "T-SQL: DML");

**//Данные из поисковой строки**

$search=urldecode($\_GET["search"]);

**//Если пользователь что-то ввел**

if(strlen($search) > 0){

$part="";

**//Ищем совпадения**

for ($i=0;$i<count($article);$i++){

**/\***

**Символы переводятся в нижний регистр.**

**Сравниваем символы из поисковой строки с символами в списке статей, совпадения записываем в переменную part.**

**substr учитывает количество переданных из строки поиска символов.**

**Если мы передали например три символа, сравним их с тремя первыми символами списка статей.**

**\*/**

if(strtolower($search)==strtolower(substr($article[$i],0,strlen($search)))){

if($part==""){

**//Выводим первое найденное совпадение**

$part="<ul id='article'><li onmouseover='this.style.backgroundColor=

\"silver\";'

onmouseout='this.style.backgroundColor=\"white\";'

onclick='add(this.innerHTML)'>".$article[$i]."</li>";

}

else{

**//Выводим следующее найденное совпадение**

$part=$part."<li onmouseover='this.style.backgroundColor=\"silver\";'

onmouseout='this.style.backgroundColor=\"white\";'

onclick='add(this.innerHTML)'>".$article[$i]."</li>";

}

}

}

**//Покажем совпадения или отладочное сообщение.**

if($part==""){

echo "Статьи нет в базе!";

}

else{

$part=$part."</ul>";

echo $part;

}

}

?>

1. **JSON**

**JSON в отличии от XML занимает меньше места, и более прост в синтаксисе.**

**Рассмотрим пример работы с файлом JSON:**

<html>

<head>

<script type='text/javascript'>

**//Счетчик позиций для файла**

var i=0;

function give(){

xhttp=new XMLHttpRequest();

**//Math.random() для предотвращения проблем GET с кэшированием**

xhttp.open('GET','json.json?x='+Math.random(),true);

xhttp.send();

xhttp.onreadystatechange=function(){

if (xhttp.readyState==4){

**//Содержимое файла json нужно обработать с помощью функции - eval**

var json=eval( '('+xhttp.responseText+')' );

**//После получения данных отобразим кнопки навигации**

document.getElementById('nav1').style.display='block';

document.getElementById('nav2').style.display='block';

**//В файле у нас четыре записи с индексами позиций от 0 до 3**

if ((i>=0) && (i<=3)){

document.getElementById('ip').innerHTML=json.list[i].ip;

document.getElementById('mask').innerHTML=json.list[i].mask;

document.getElementById('traf').innerHTML=json.list[i].traf;

}

**//Если достигли последней позиции, идем по кругу**

else{

i = 0;

document.getElementById('ip').innerHTML=json.list[i].ip;

document.getElementById('mask').innerHTML=json.list[i].mask;

document.getElementById('traf').innerHTML=json.list[i].traf;

}

}

}

}

</script>

</head>

<body>

<input type='button' value='Получить данные' onclick='give()'/>

<p>Адрес: <span id='ip'>Нет данных</span></p>

<p>Маска: <span id='mask'>Нет данных</span></p>

<p>Траффик: <span id='traf'>Нет данных</span></p>

<input id='nav1' style='display:none;' type='button' value='Предыдущая' onclick='if((i!=0))i--;give()'/>

<input id='nav2' style='display:none;' type='button' value='Следующая' onclick='if((i!=13))i++;give()'/>

</body>

</html>

**Сам файл json.json:**

{

list:

[

{

ip:"1.1.1.1",

mask:"/24",

traf:834

},

{

ip:"2.2.2.2",

mask:"/24",

traf:123

},

{

ip:"3.3.3.3",

mask:"/24",

traf:234

},

{

ip:"4.4.4.4",

mask:"/24",

traf:134453

}

]

}

**На этом наше краткое знакомство с технологией AJAX мы – закончим, с этими знаниями без труда можно осуществить легкий старт в освоении более сложных примеров.**