

ლონლაქე გორგი
თარგმანი ვებ გვერდიდან
www.w3schools.com

asynchronous javascript and html

AJAX

Programing

ვიკიწიგნების ბიბლიოთეკიდან



იფიქრე თავისუფლად
ისწავლე თავისუფლად

შესავალი

რა უნდა იცოდეთ?

სანამ ამ ენის შესწავლას დაიწყებდეთ თქვენ უნდა იცოდეთ :

- [HTML](#) / XHTML
- JavaScript

AJAX - Asynchronous JavaScript and XML(ასინქრონიზებული JavaScript და XML)

Ajax არ არის ახალი პროგრამირების ენა, მაგრამ ის არის უკეთესი, უფრო სწრაფი და ინტერაქტული ვებ პროგრამების შექმნის ტექნოლოგია.

Ajax-ის დახმარებით თქვენს JavaScript-ს შეუძლია დაუკავშირდეს სერვერს XMLHttpRequest ობიექტის გამოყენებით. ამ ობიექტის დახმარებით, თქვენს JavaScript-ს შეუძლია ივაჭროს ვებ სერვერით, გვერდის განახლების გარეშე.

Ajax იყენებს ასინქრონიზებულ მონაცემთა გადაცემას ბრაუზერსა და ვებ სერვერს შორის და საშუალებას აძლევს ვებ გვერდს სერვერიდან მოითხოვოს ინფორმაციის მხოლოდ ის ბიტები, რომლის მიმოხილვაც სურს მომხმარებელს.

Ajax ტექნოლოგია ინტერნეტ პროგრამებს აქცევს უფრო პატარა, სწრაფ და მომხმარებელზე მორგებულ პროგრამებად.

ვებ სტანდარტები

Ajax ბაზირებულია შემდეგ ვებ სტანდარტებზე:

- JavaScript
- XML
- HTML
- CSS

Ajax-ში გამოყენებული ვებ სტანდარტები კარგადაა განსაზღვრული და უზრუნველყოფილია ყველა ბრაუზერის მიერ. Ajax-ის პროგრამები არიან ბრაუზერისგან და პლათფორმისგან დამოუკიდებელი.

Ajax, როგორც უკეთესი ინტერნეტ პროგრამა

ბევრი ვებ პროგრამა სხვა სისტემურ პროგრამებზე სასარგებლოა, რადგან ეს პროგრამები არის ადვილად დასაყენებელი, უზრუნველყოფილი და ადვილად შესაქმნელი.

თუმცა ინტერნეტ პროგრამები არ არის ყოველთვის „მდიდარი“ და მომხმარებელზე მორგებული, ისე, როგორც სისტემური პროგრამები.

Ajax-ით ინტერნეტ პროგრამები შესაძლოა გახადოს უფრო მდიდარი და მომხმარებელზე მორგებული.

HTTP მოთხოვნები

Ajax იყენებს HTTP მოთხოვნებს

ტრადიციულ JavaScript კოდირებაში, თუ საჭიროა მიიღო ნებისმიერი ინფორმაცია მონაცემთა ბაზებიდან, ან ფაილი სერვერიდან, ან გააგზავნო მომხმარებლის ინფორმაცია სერვერზე, სერვერზე უნდა შეიქმნას HTML, GET და POST მონაცემებიდან. მომხმარებელი იმოქმედებს „გაგზავნა“ ღილაკზე ინფორმაციის გასაგზავნად, დაელოდება სერვერის პასუხს და შემდეგ ახალი გვერდი ჩამოტვირთავს შედეგებს.

რადგან სერვერი ტვირთავს ახალ გვერდს რათა გადაგზავნოს მომხმარებლის მიერ შეყვანილი ინფორმაცია, ამიტომ საერთოდ პროგრამა მუშაობს ნელა და არ არის მორგებული მომხმარებელზე.

Ajax-ის დახმარებით, JavaScript უკავშირდება უშუალოდ სერვერს XMLHttpRequest ობიექტის დახმარებით.

HTTP-ის მოთხოვნით ვებ გვერდს შეუძლია გააგზავნოს მოთხოვნა და მიიღოს პასუხი სერვერიდან გვერდის განახლების გარეშე. მომხმარებელი რჩება არსებულ გვერდზე.

XMLHttpRequest ობიექტი

XMLHttpRequest ობიექტის გამოყენებით, ვებ პროგრამისტი გვერდის შემცველობას ანახლებს გვერდის ახლიდან ჩტვირთვის გარეშე!

Ajax პოპულარული გახდა კომპანია Google-მა 2005 წელს.

პირველი Ajax როგრამა

Ajax-ის მუშაობაში გასარკვევად შექმენით მცირე Ajax პროგრამა.

პირველ რიგში შექმენით სტანდარტული HTML ფორმა ორი ტექსტური ველით: მომხმარებლის სახელი და დრო. მომხმარებლის სახელის ველი შეივსება მომხმარებლის მიერ, ხოლო დროის ველი კი Ajax-ის მიერ.

```
<html>
<body>
<form name="myForm">
Name: <input type="text" name="username" />
Time: <input type="text" name="time" />
</form>
</body>
</html>
```

გაითვალისწინეთ რომ ამ HTML ფორმას არ გააჩნია „გაგზავნა“ ლილაკი.

ბრაუზერის მხარდაჭერა

Ajax-ის ძირითადი პრინციპი არის XMLHttpRequest ობიექტი.

სხვადასხვა ბრაუზერები იყენებენ XMLHttpRequest ობიექტის შექმნის სხვადასხვა მეთოდებს.

Internet Explorer იყენებს ActiveXObject-ს, როცა სხვა ბრაუზერები იყენებენ ჩაშენებულ JavaScript ობიექტს XMLHttpRequest.

ამ ობიექტის შექმნისათვის და შემდგომ ბრაუზერში გამოყენებისათვის, გამოიყენეთ ოპერატორი "try and catch".

მაგალითი:

```
<html>
<body>
<script type="text/javascript">
function ajaxFunction()
{
var xmlhttp;
try
{
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Your browser does not support AJAX!");
return false;
}
}
}
}
}
</script>
<form name="myForm">
Name: <input type="text" name="username" />
Time: <input type="text" name="time" />
</form>
</body>
</html>
```

მაგალითის ახსნა:

პირველად შეიქმნება ცვლადი `xmlHttp`. შემდგომ უნდა შეიქმნას ობიექტი `XMLHttpRequest=new XMLHttpRequest()`. ეს იმუშავებს Firefox, Opera და Safari ბრაუზერებზე. თუ ეს არ იმუშავებს შეეცადეთ შექმნათ ობიექტი `xmlHttp=new ActiveXObject("Msxml2.XMLHTTP")`, რომელიც იმუშავებს Internet Explorer 6.0+ ბრაუზერზე, და თუ ესეც არ იმუშავებს შეეცადეთ შექმნათ `xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")` ობიექტი, რომელიც იმუშავებს Internet Explorer 5.5+ ბრაუზერზე.

თუ ამ სამი ობიექტიდან არცერთი არ იმუშავებს, მაშინ მომხმარებელს აქვს ძველი ბრაუზერი და ის მიიღებს შეტყობინებას რომ მისი ბრაუზერი არ უზრუნველყოფს Ajax.

AJAX - XMLHttpRequest ობიექტი

მონაცემების სერვერზე გაგზავნამდე, უნდა შეისწავლოთ XMLHttpRequest ობიექტის სამი მთავარი თვისება.

onreadystatechange თვისება

სერვერზე მოთხოვნის გაგზავნის შემდეგ, საჭიროა ფუნქცია, რომელიც მიიღებს სერვერის მიერ გამოგზავნილ მონაცემებს.

onreadystatechange თვისებას გააჩნია ფუნქცია, რომელიც უზრუნველყოფს პასუხის მიღებას სერვერიდან. ქვემოთ მოყვანილი კოდი საზღვრავს ცარიელ ფუნქციას და ამავდროულად აყენებს onreadystatechange თვისებას.

```
xmlHttpRequest.onreadystatechange=function()
{
// We are going to write some code here
}
```

readyState თვისება

readyState თვისება შეიცავს სერვერის პასუხის სტატუსს. readyState-ის ყოველ ცვლილებაზე გაემგზავნება onreadystatechange ფუნქცია.

ქვემოთ მოყვანილია readyState თვისების შესაძლო ვარიანტები:

მდგომარეობა	აღწერა
0	მოთხოვნა არ არის ინიციალიზებული
1	მოთხოვნა დაყენდა
2	მოთხოვნა გაიგზავნა
3	მოთხოვნა დამუშავების პროცესშია
4	მოთხოვნა შესრულდა

onreadystatechange ფუნქციაში ჩაამატეთ If ოპერატორი, რათა დატესტოთ სერვერის პასუხის მიღების შედეგები:

```
xmlHttpRequest.onreadystatechange=function()
{
if(xmlHttpRequest.readyState==4)
{
// Get the data from the server's response
}
}
```

responseText თვისება

სერვერიდან გამოგზავნილი პასუხი შესაძლებელია მოიძებნოს responseText თვისების დახმარებით.

მაგალითი:

```
xmlHttp.onreadystatechange=function ()
{
if (xmlHttp.readyState==4)
{
document.myForm.time.value=xmlHttp.responseText;
}
}
```

AJAX - მოთხოვნის გაგზავნა Server-ზე

სერვერზე მოთხოვნის გასაგზავნად გამოიყენება Open() და Send() მეთოდები.

Open() მეთოდი შეიცავს სამ არგუმენტს. პირველი არგუმენტი განსაზღვრავს სერვერზე გასაგზავნი მოთხოვნის მეთოდს (GET, ან POST). მეორე არგუმენტი განსაზღვრავს სერვერული სკრიპტის ბმულს. მესამე არგუმენტი განსაზღვრავს მოთხოვნის ასინქრონიზაციას. Send() მეთოდი აგზავნის მოთხოვნას სერვერზე. თუ მიიღებთ HTML და ASP ფაილებს, რომლებიც არიან იგივე დირექტორიაში, მაშინ კოდი იქნება შემდეგნაირად ჩამოყალიბებული:

```
xmlHttp.open ("GET", "time.asp", true);
xmlHttp.send (null);
```

ახლა უნდა გადაწყვიტოთ, თუ როდის გაემგზავნება Ajax ფუნქცია. მიეცით საშუალება ფუნქციას გაემგზავს, როგორც "behind the scenes", როცა მომხმარებელი შეიყვანს მონაცემებს მომხმარებლის სახელის ველში:

```
<form name="myForm">
Name: <input type="text"
onkeyup="ajaxFunction();" name="username" />
Time: <input type="text" name="time" />
</form>
```

მაგალითი:


```
<html>
<body>
<script type="text/javascript">
function ajaxFunction()
{
var xmlHttp;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Your browser does not support AJAX!");
return false;
}
}
}
xmlHttp.onreadystatechange=function()
{
if(xmlHttp.readyState==4)
{
document.myForm.time.value+xmlHttp.responseText;
}
}
xmlHttp.open("GET","time.asp",true);
xmlHttp.send(null);
}
</script>
<form name="myForm">
Name: <input type="text"
onkeyup="ajaxFunction();" name="username" />
Time: <input type="text" name="time" />
</form>
</body>
</html>
```

Ajax - სერვერული სკრიპტი

ახლა შექმენით სკრიპტი, რომელიც გამოსახავს სერვერის მიმდინარე დროს.

responseText თვისება მოაგროვებს სერვერიდან გამოგზავნილ მონაცემებს. აქ საჭირო იქნება დროის უკან გაგზავნა. კოდი ფაილში "time.asp" გამოიყურება ასე:

```
<%  
response.expires=-1  
response.write(time)  
%>
```

შენიშვნა: Expires თვისება განსაზღვრავს, თუ რამდენ ხანს შეინახება გვერდი ბრაუზერზე. თუ მომხმარებელი დააბრუნებს იგივე გვერდს, მანამ სანამ მას დრო გაუვა, მაშინ შენახული ვერსია გამოისახება ეკრანზე. Response.Expires=-1 მიუთითებს იმაზე, რომ გვერდს დრო არ გაუვა.

Ajax შეთავაზება

AJAX HTML გვერდი

ეს არის HTML გვერდი. ის შეიცავს უბრალო HTML კოდს და ბმულს JavaScript-ზე.

```
<html>
<head>
<script src="clienthint.js"></script>
</head>
<body>
<form>
First Name:
<input type="text" id="txt1"
onkeyup="showHint (this.value) ">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

AJAX JavaScript

ეს არის JavaScript კოდი, ჩაწერილი ფაილში "clienthint.js":

```
var xmlHttp

function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
xmlHttp=GetXmlHttpRequest();
if (xmlHttp==null)
{
alert ("Your browser does not support AJAX!");
return;
}
var url="gethint.asp";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlHttp.onreadystatechange=stateChanged;
xmlHttp.open ("GET", url, true);
xmlHttp.send (null);
}
{
if (xmlHttp.readyState==4)
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText;
}
}
```

```

function GetXmlHttpRequestObject()
{
var xmlHttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlHttp;
}

```

AJAX ASP მაგალითი

კოდი ფაილში "gethint.asp" დაწერილია VBScript-ით Internet Information Server (IIS)-სათვის. ეს უბრალოდ ამოწმებს სახელების მასივს და კლიენტთან აბრუნებს კორესპონდენციის სახელებს.

```

<%
response.expires=-1
dim a(30)
'Fill up array with names
a(1)="Anna"
a(2)="Brittany"
a(3)="Cinderella"
a(4)="Diana"
a(5)="Eva"
a(6)="Fiona"
a(7)="Gunda"
a(8)="Hege"
a(9)="Inga"
a(10)="Johanna"
a(11)="Kitty"
a(12)="Linda"
a(13)="Nina"
a(14)="Ophelia"
a(15)="Petunia"
a(16)="Amanda"
a(17)="Raquel"
a(18)="Cindy"
a(19)="Doris"
a(20)="Eve"
a(21)="Evita"
a(22)="Sunniva"
a(23)="Tove"
a(24)="Unni"
a(25)="Violet"

```

```

a(26)="Liza"
a(27)="Elizabeth"
a(28)="Ellen"
a(29)="Wenche"
a(30)="Vicky"
'get the q parameter from URL
q=ucase(request.querystring("q"))
'lookup all hints from array if length of q>0
if len(q)>0 then
    hint=""
    for i=1 to 30
        if q=ucase(mid(a(i),1,len(q))) then
            if hint="" then
                hint=a(i)
            else
                hint=hint & " , " & a(i)
            end if
        end if
    next
end if
'Output "no suggestion" if no hint were found
'or output the correct values
if hint="" then
    response.write("no suggestion")
else
    response.write(hint)
end if
%>

```

Ajax PHP მაგალითი

```

<?php
header("Cache-Control: no-cache, must-revalidate");
// Date in the past
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");

// Fill up array with names
$a[]="Anna";
$a[]="Brittany";
$a[]="Cinderella";
$a[]="Diana";
$a[]="Eva";
$a[]="Fiona";
$a[]="Gunda";
$a[]="Hege";
$a[]="Inga";
$a[]="Johanna";
$a[]="Kitty";
$a[]="Linda";
$a[]="Nina";
$a[]="Ophelia";
$a[]="Petunia";
$a[]="Amanda";
$a[]="Raquel";
$a[]="Cindy";
$a[]="Doris";
$a[]="Eve";
$a[]="Evita";

```

```
$a[]="Ellen";
$a[]="Wenche";
$a[]="Vicky";
//get the q parameter from URL
$q=$_GET["q"];
//lookup all hints from array if length of q>0
if (strlen($q) > 0)
{
    $hint="";
    for($i=0; $i<count($a); $i++)
    {
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
        {
            if ($hint=="")
            {
                $hint=$a[$i];
            }
            else
            {
                $hint=$hint." , ".$a[$i];
            }
        }
    }
}

// Set output to "no suggestion" if no hint were found
// or to the correct values
if ($hint == "")
{
    $response="no suggestion";
}
else
{
    $response=$hint;
}

//output the response
echo $response;
?>
```

Ajax კოდი შეთავაზების მაგალითისათვის

Ajax HTML გვერდი

ეს არის HTML გვერდი. ის შეიცავს უბრალო HTML ფორმას და ბმულს JavaScript-ზე.

```
<html>
<head>
<script src="clienthint.js"></script>
</head>
<body>
<form>
First Name:
<input type="text" id="txt1"
onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

Ajax JavaScript

ეს არის ფაილში - "clienthint.js", მოთავსებული JavaScript კოდი.

```
var xmlHttp

function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
xmlHttp=GetXmlHttpRequest();
if (xmlHttp==null)
{
alert ("Your browser does not support AJAX!");
return;
}
var url="gethint.asp";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlHttp.onreadystatechange=stateChanged;
xmlHttp.open ("GET", url, true);
xmlHttp.send(null);
}
function stateChanged()
{
if (xmlHttp.readyState==4)
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText;
}
}

function GetXmlHttpRequest()
```

```

{
var xmlhttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlhttp;
}

```

Ajax სერვერული გვერდი - ASP და PHP

ASP

წინა პარაგრაფში მოყვანილი მაგალითის გვერდი დაწერილია VBScript-ით, ინტერნეტის ინფორმაციის სერვერისათვის(IIS). ის უბრალოდ ამოწმებს სახელების მასივს.

```

<%
response.expires=-1
dim a(30)
'Fill up array with names
a(1)="Anna"
a(2)="Brittany"
a(3)="Cinderella"
a(4)="Diana"
a(5)="Eva"
a(6)="Fiona"
a(7)="Gunda"
a(8)="Hege"
a(9)="Inga"
a(10)="Johanna"
a(11)="Kitty"
a(12)="Linda"
a(13)="Nina"
a(14)="Ophelia"
a(15)="Petunia"
a(16)="Amanda"
a(17)="Raquel"
a(18)="Cindy"
a(19)="Doris"
a(20)="Eve"
a(21)="Evita"
a(22)="Sunniva"
a(23)="Tove"
a(24)="Unni"
a(25)="Violet"

```



```

a(26)="Liza"
a(27)="Elizabeth"
a(28)="Ellen"
a(29)="Wenche"
a(30)="Vicky"
'get the q parameter from URL
q=ucase(request.querystring("q"))
'lookup all hints from array if length of q>0
if len(q)>0 then
  hint=""
  for i=1 to 30
    if q=ucase(mid(a(i),1,len(q))) then
      if hint="" then
        hint=a(i)
      else
        hint=hint & " , " & a(i)
      end if
    end if
  next
end if
'Output "no suggestion" if no hint were found
'or output the correct values
if hint="" then
  response.write("no suggestion")
else
  response.write(hint)
end if
%>

```

PHP

```

<?php
header("Cache-Control: no-cache, must-revalidate");
// Date in the past
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");

// Fill up array with names
$a[]="Anna";
$a[]="Brittany";
$a[]="Cinderella";
$a[]="Diana";
$a[]="Eva";
$a[]="Fiona";
$a[]="Gunda";
$a[]="Hege";
$a[]="Inga";
$a[]="Johanna";
$a[]="Kitty";
$a[]="Linda";
$a[]="Nina";
$a[]="Ophelia";
$a[]="Petunia";
$a[]="Amanda";
$a[]="Raquel";
$a[]="Cindy";
$a[]="Doris";

```

```
$a[]="Eve";
$a[]="Evita";
$a[]="Sunniva";
$a[]="Tove";
$a[]="Unni";
$a[]="Violet";
$a[]="Liza";
$a[]="Elizabeth";
$a[]="Ellen";
$a[]="Wenche";
$a[]="Vicky";
//get the q parameter from URL
$q=$_GET["q"];
//lookup all hints from array if length of q>0
if (strlen($q) > 0)
{
    $hint="";
    for($i=0; $i<count($a); $i++)
    {
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
        {
            if ($hint=="")
            {
                $hint=$a[$i];
            }
            else
            {
                $hint=$hint." , ".$a[$i];
            }
        }
    }
}

// Set output to "no suggestion" if no hint were found
// or to the correct values
if ($hint == "")
{
    $response="no suggestion";
}
else
{
    $response=$hint;
}

//output the response
echo $response;
?>
```

Ajax მონაცემთა ბაზა

მაგალითი:

ეს არის HTML გვერდი. ის შეიცავს უბრალო HTML ფორმას და ბმულს JavaScript-ზე.

```
<html>
<head>
<script src="selectcustomer.js"></script>
</head>
<body>
<form>
Select a Customer:
<select name="customers" onchange="showCustomer(this.value)">
<option value="ALFKI">Alfreds Futterkiste
<option value="NORTS ">North/South
<option value="WOLZA">Wolski Zajazd
</select>
</form>
<p>
<div id="txtHint"><b>Customer info will be listed here.</b></div>
</p>
</body>
</html>
```

როგორც ხედავთ, ეს არის უბრალო HTML ჩამოსაშლელი მენიუ, სახელად "customers"

ზემოთ მოყვანილ მაგალითში პარაგრაფი შეიცავს განყოფილებას "txtHint". ეს განყოფილება გვერდზე ინახავს სივრცეს, ვებ სერვერიდან გამოგზავნილი მონაცემების გამოსასახავად.

როდესაც მომხმარებელი მონისნავს მონაცემს, ფუნქცია სახელად "showCustomer()", გაეშვება. ამ ფუნქციის გაშვებას იწვევს "onchange" მოვლენა. სხვა სიტყვებით: ყოველთვის, როდესაც მომხმარებელი შეცვლის მნიშვნელობას ჩამოსაშლელ მენიუში, გამოძახებულ იქნება ფუნქცია showCustomer.

Ajax JavaScript

ეს არის ფაილში "selectcustomer.js" მოთავსებული JavaScript.

```
var xmlHttp

function showCustomer(str)
{
xmlHttp=GetXmlHttpRequest();
if (xmlHttp==null)
{
alert ("Your browser does not support AJAX!");
return;
}
var url="getcustomer.asp";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlHttp.onreadystatechange=stateChanged;
xmlHttp.open ("GET", url, true);
xmlHttp.send (null);
}
```

```

function stateChanged()
{
if (xmlHttp.readyState==4)
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText;
}
}

function GetXmlHttpRequestObject()
{
var xmlHttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlHttp;
}

```

Ajax სერვერული გვერდი

სერვერული გვერდი, რომელსაც იძახებს JavaScript, არის უბრალო ASP გვერდი, სახელად "getcustomer.asp".

კოდი უშვებს SQL მონაცემთა ბაზებს და შედეგებს აბრუნებს HTML გვერდზე.

```

<%
response.expires=-1
sql="SELECT * FROM CUSTOMERS WHERE CUSTOMERID="
sql=sql & "'" & request.querystring("q") & "'"

set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open(Server.MapPath("/db/northwind.mdb"))
set rs = Server.CreateObject("ADODB.recordset")
rs.Open sql, conn

response.write("<table>")
do until rs.EOF
for each x in rs.Fields
response.write("<tr><td><b>" & x.name & "</b></td>")
response.write("<td>" & x.value & "</td></tr>")
next
rs.MoveNext
loop

response.write("</table>")
%>

```

Ajax XML

მაგალითი:

ეს არის HTML გვერდი. ის შეიცავს უბრალო HTML ფორმას და ბმულს JavaScript-ზე.

```
<html>
<head>
<script src="selectcd.js"></script>
</head>
<body>
<form>
Select a CD:
<select name="cds" onchange="showCD(this.value)">
<option value="Bob Dylan">Bob Dylan</option>
<option value="Bonnie Tyler">Bonnie Tyler</option>
<option value="Dolly Parton">Dolly Parton</option>
</select>
</form>
<p>
<div id="txtHint"><b>CD info will be listed here.</b></div>
</p>
</body>
</html>
```

როგორც ხედავთ, ეს არის უბრალო HTML ჩამოსაშლელი მენიუ, სახელად "cds".

ზემოთ მოყვანილ მაგალითში პარაგრაფი შეიცავს განყოფილებას "txtHint". ეს განყოფილება გვერდზე ინახავს სივრცეს, ვებ სერვერიდან გამოგზავნილი მონაცემების გამოსასახავად.

როდესაც მომხმარებელი მონისნავს მონაცემს, ფუნქცია სახელად "showCD", გაეშვება. ამ ფუნქციის გაშვებას იწვევს "onchange" მოვლენა. სხვა სიტყვებით: ყოველთვის, როდესაც მომხმარებელი შეცვლის მნიშვნელობას ჩამოსაშლელ მენიუში, გამოძახებულ იქნება ფუნქცია showCD.

Ajax JavaScript

ეს არის ფაილში "selectcd.js" მოთავსებული JavaScript.

```
var xmlHttp
function showCD(str)
{
xmlHttp=GetXmlHttpRequest();
if (xmlHttp==null)
{
alert ("Your browser does not support AJAX!");
return;
}
var url="getcd.asp";
url=url+"?q="+str;
url=url+"&sid="+Math.random();
xmlHttp.onreadystatechange=stateChanged;
xmlHttp.open ("GET", url, true);
xmlHttp.send (null);
}
```

```

function stateChanged()
{
if (xmlHttp.readyState==4)
{
document.getElementById("txtHint").innerHTML=xmlHttp.responseText;
}
}
function GetXmlHttpRequestObject()
{
var xmlHttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlHttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlHttp;
}

```

Ajax სერვერული გვერდი

სერვერული გვერდი, რომელსაც იძახებს JavaScript, არის უბრალო ASP გვერდი, სახელად "getcd.asp".

```

<%
response.expires=-1
q=request.querystring("q")

set xmlDoc=Server.CreateObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load(Server.MapPath("cd_catalog.xml"))

set nodes=xmlDoc.selectNodes("CATALOG/CD[ARTIST='" & q & "' ]")

for each x in nodes
  for each y in x.childnodes
    response.write("<b>" & y.nodename & ":</b> ")
    response.write(y.text)
    response.write("<br />")
  next
next
%>

```