



Προγραμματισμός Διαδικτύου

Δρ. Μηνάς Δασυγένης
mdasygenis@uowm.gr



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Προγραμματισμός Διαδικτύου

Ασύγχρονος Προγραμματισμός (AJAX)



Επεκτάσεις..

- Αφού κατανοήσετε τα
 - HTML
 - XHTML
 - CSS
 - PHP
 - DOM

... το επόμενο βήμα είναι να μάθετε AJAX
(Asynchronous JavaScript and XML)



Τι είναι το AJAX

- Δεν είναι μια νέα γλώσσα προγραμματισμού.
 - Είναι ένας νέος τρόπος να χρησιμοποιούνται τα υπάρχοντα στάνταρντ.
 - Το AJAX είναι “η τέχνη της ανταλλαγής δεδομένων με ένα διακομιστή και η ενημέρωση στοιχείων της σελίδας χωρίς να γίνει επαναφόρτιση (reload) ή υποβολή (submit) από το χρήστη”.
-



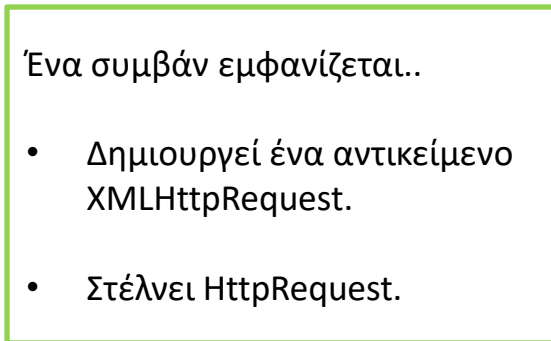
Τι επιτρέπει το AJAX

- Το AJAX είναι μια τεχνική για τη δημιουργία γρήγορων και δυναμικών σελίδων.
- Επιτρέπει την ασύγχρονη ενημέρωση δεδομένων της σελίδας με την ανταλλαγή μικρού μεγέθους δεδομένων με το διακομιστή στο παρασκήνιο (χωρίς να το καταλαβαίνει ο χρήστης).
- Ενημερώνεται η σελίδα χωρίς να γίνει επαναφόρτιση (reload).
- Οι κλασικές σελίδες πρέπει κάθε φορά να φορτώνουν μια ολόκληρη σελίδα για νέα δεδομένα.
- Χρησιμοποιείται ευρέως (facebook, google, youtube...).



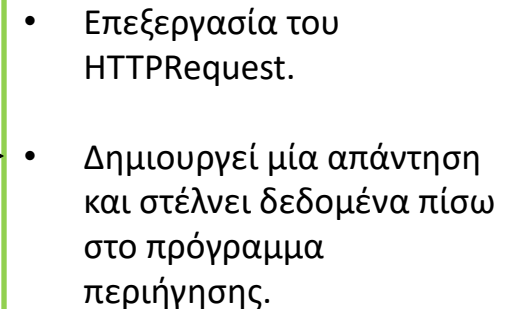
Σχεδιάγραμμα λειτουργίας AJAX

πρόγραμμα περιήγησης

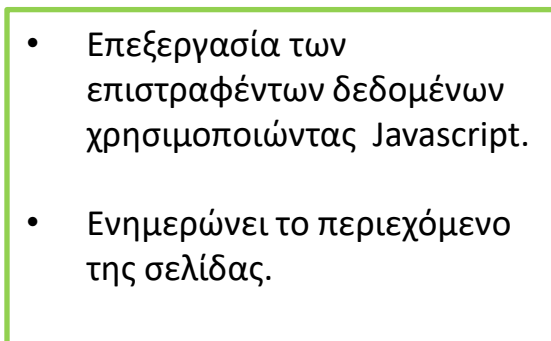


Διαδίκτυο

Διακομιστής



πρόγραμμα περιήγησης



Διαδίκτυο

Το AJAX βασίζεται σε πρότυπα

- **CSS** (για το style).
- **Javascript/DOM** (για την εμφάνιση και τη διαδραστικότητα της σελίδας).
- **XML** (χρησιμοποιείται για τη μορφή των μεταφερόμενων αρχείων).
- **XMLHttpRequest** αντικείμενο (για την ασύγχρονη επικοινωνία με το διακομιστή).

Έγινε ευρέως διαδεδομένο χάρη στο google το 2005 (google suggest).



Ένα παράδειγμα στο AJAX (1/2)

Έστω ότι έχουμε αυτή τη σελίδα:

```
<html>
  <body>

    <div id="myDiv"> <h2> Let AJAX change this
    text</h2></div>
    <button type="button" onclick="loadXMLDoc()">
    Change Content </button>

  </body>
</html>
```

Απαιτείται ένα <div> ή <tr> στο οποίο θα αποθηκεύεται η πληροφορία μας.

Let AJAX change this text

Change Content



Ένα παράδειγμα στο AJAX (2/2)

Προσθέτουμε στο HEAD τη συνάρτηση loadXMLDoc() η οποία θα υλοποιήσει το AJAX.

```
<head>
<script type="text/javascript">
function loadXMLDoc()
{
.... AJAX script goes here ...
}
</script>
</head>
```

Όταν πατηθεί λοιπόν το κουμπί εκτελείται ο κώδικας AJAX.



Ένα παράδειγμα στο AJAX

- Όλα τα σύγχρονα προγράμματα περιήγησης (browsers) υποστηρίζουν τη συνάρτηση **XMLHttpRequest()**;
- Επειδή στο παρελθόν ο IE5,IE6 χρησιμοποιούσε άλλο αντικείμενο για AJAX, χρησιμοποιούμε τον παρακάτω κώδικα μέσα στη συνάρτηση **loadXMLDoc()**:

```
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); }
```



Αποστολή αίτησης

- Για να στείλουμε την αίτηση ασύγχρονα:

```
xmlhttp.open("GET","ajax_info.txt",true);  
xmlhttp.send();
```

Μέθοδος	Περιγραφή
<code>open(method,url,async)</code>	<p>Καθορίζει τον τύπο της αίτησης, τη διεύθυνση URL, και αν η αίτηση πρέπει να παραδοθεί ασύγχρονα ή όχι.</p> <p>method: ο τύπος της αίτησης: GET ή POST. url: η τοποθεσία του αρχείου στο διακομιστή. async: true (ασύγχρονα) ή false (συγχρονισμένα)</p>
<code>send(string)</code>	<p>Αποστέλλει την αίτηση στο διακομιστή.</p> <p>String: Χρησιμοποιείται μόνο για αιτήσεις POST.</p>



GET ή POST

- GET
 - Απλούστερα
 - Γρηγορότερα
 - Χρησιμοποιείται συχνότερα

- POST
 - Ενημερώνει ένα αρχείο ή μια βάση δεδομένων στο διακομιστή.
 - Αποστολή μεγάλου όγκου δεδομένων.
 - Αποστολή της εισόδου του χρήστη (ενδέχεται να περιέχει άγνωστους χαρακτήρες).
 - Είναι πιο ασφαλές.



Παραδείγματα με GET / POST

```
xmlhttp.open("GET", "demo_get.asp?t=" +  
Math.random(), true);  
xmlhttp.send();
```

```
xmlhttp.open("GET", "demo_get2.asp?fname=Henry&lname=Ford", true);  
xmlhttp.send();
```

Παράδειγμα **POST**

```
xmlhttp.open("POST", "ajax_test.asp", true);  
xmlhttp.setRequestHeader("Content-  
type", "application/x-www-form-urlencoded");  
xmlhttp.send("fname=Henry&lname=Ford");
```



Επιπρόσθετες κεφαλίδες

- Αποστολή επιπρόσθετων κεφαλίδων με τη συνάρτηση `setRequestHeader(header, value)`

Μέθοδος	Περιγραφή
<code>setRequestHeader(header,value)</code>	<p>Προσθέτει επικεφαλίδες HTTP στην αίτηση.</p> <p>header: καθορίζει το όνομα της επικεφαλίδας.</p> <p>value: Καθορίζει την τιμή της επικεφαλίδας.</p>

Ασύγχρονη επεξεργασία (async=true) ΠΡΟΤΙΜΑΤΑΙ

- Μόλις έρθουν τα δεδομένα από το διακομιστή πρέπει να τα επεξεργαστούμε. Αυτό το επιτυγχάνουμε με τις παρακάτω γραμμές:

```
xmlhttp.onreadystatechange=function()  
{ if (xmlhttp.readyState==4 && xmlhttp.status==200)  
  {  
    document.getElementById("myDiv").innerHTML  
    =xmlhttp.responseText;    } }  
}
```

```
xmlhttp.open("GET", "ajax_info.txt", true);  
xmlhttp.send();
```



Σύγχρονη επεξεργασία (async=false)

- Αν στο **xmlhttp.open** δώσουμε την τρίτη παράμετρο ως false τότε το script εκτελείται σύγχρονα.
- Αυτό σημαίνει ότι έως να επιστρέψει τα αποτελέσματα ο διακομιστής **δε συνεχίζει η εκτέλεση παρακάτω**. Πιο απλή μορφή:

```
xmlhttp.open("GET", "ajax_info.txt", false);  
xmlhttp.send();  
document.getElementById("myDiv").innerHTML=  
    xmlhttp.responseText;
```



Η απάντηση (response) από το διακομιστή (server)

- Η απάντηση τοποθετείται σε div ή tr ως:
 - ResponseText
 - ResponseXML

Ιδιότητα	Περιγραφή
responseText	Παίρνει τα δεδομένα της απάντησης σαν μία συμβολοσειράς.
responseXML	Παίρνει τα δεδομένα της απάντησης σαν δεδομένα XML.

```
document.getElementById("myDiv").innerHTML+xmlhttp.responseText;
```

```
xmlDoc+xmlhttp.responseXML;  
var txt="";  
x+xmlDoc.getElementsByTagName("ARTIST");  
for (i=0;i<x.length;i++)  
{  
txt=txt + x[i].childNodes[0].nodeValue + "<br />";  
}  
document.getElementById("myDiv").innerHTML=txt;
```



Η συνάρτηση onreadystatechange

- Καλείται κάθε φορά που έρχονται δεδομένα από το διακομιστή.

Ιδιοκτησία	Περιγραφή
onreadystatechange	Αποθηκεύει μία συνάρτηση (ή ένα όνομα συνάρτησης) για να καλεστεί αυτόματα κάθε φορά που αλλάζει η ιδιοκτησία readyState.
readyState	Κρατάει την κατάσταση του XMLHttpRequest. Αλλαγές από 0 έως 4. 0: η αίτηση δεν έχει αρχικοποιηθεί 1: καθιερώθηκε η σύνδεση του διακομιστή 2: η αίτηση λήφθηκε 3: επεξεργασία αίτησης 4: η αίτηση ολοκληρώθηκε και η απάντηση είναι έτοιμη
status	200: "OK" 404: Page no found.

Σημειώστε ότι η συνάρτηση onreadystatechange καλείται 4 φορές για κάθε κατάσταση (1,2,3,4)



Παράδειγμα με την onreadystatechange()

```
xmlhttp.onreadystatechange=function()  
{  
  if (xmlhttp.readyState==4 &&  
      xmlhttp.status==200)  
  {  
    document.getElementById("myDiv").innerHTML=  
xmlhttp.responseText;  
  }  
}
```



AJAX

Προχωρημένα θέματα



Πολλαπλές διεργασίες AJAX στην ίδια σελίδα

- Αν υπάρχουν πολλαπλές AJAX διεργασίες τότε πρέπει να χρησιμοποιηθεί μια συνάρτηση επανάκλησης (callback) για κάθε έργο (task) και μια για το αντικείμενο XMLHttpRequest .
- Χρησιμοποιείται η τεχνική της συνάρτησης επανάκλησης (callback). Η συνάρτηση επανάκλησης (callback) ονομάζεται η συνάρτηση που περνάει ως παράμετρος σε μια άλλη συνάρτηση.



Παράδειγμα: Δημιουργώντας τη `showhint()` σε φόρμα PHP

- Μια καλή τεχνική στις φόρμες είναι η δημιουργία υποδείξεων (hints) καθώς πληκτρολογεί ο χρήστης:

Example

Start typing a name in the input field below:

First name:

Suggestions:

Example

Start typing a name in the input field below:

First name:

Suggestions: Anna , Amanda



Η φόρμα ενεργοποιεί τον χειριστή συμβάντων (event handler) για onkeyup (1/2)

```
<form action="">
```

```
First name: <input type="text"  
id="txt1"  
onkeyup="showHint(this.value)" />
```

```
</form>
```

```
<p>Suggestions: <span  
id="txtHint"></span></p>
```



Η συνάρτηση showhint(str) (2/2)

```
function showHint(str)
{var xmlhttp;
if (str.length==0) {
document.getElementById("txtHint").innerHTML=""; return;
}
if (window.XMLHttpRequest) { xmlhttp=new
XMLHttpRequest(); }
else { xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); }

xmlhttp.onreadystatechange=function() {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
document.getElementById("txtHint").innerHTML=xmlhttp.respon
seText;    } }

xmlhttp.open("GET","gethint.asp?q="+str,true);
xmlhttp.send();
}
```



Εξήγηση της συνάρτησης showhint()

- Αν είναι άδειο το πεδίο εισαγωγής (input field) τότε μη δείχνεις τίποτα.
- Αν δεν είναι άδειο:
 - Δημιούργησε το αντικείμενο XMLHttpRequest.
 - Δημιούργησε τη συνάρτηση όταν μας στείλει δεδομένα ο διακομιστής.
 - Στείλε την αίτηση στο διακομιστή.
- Παρατηρήστε ότι χρησιμοποιούμε τη μέθοδο GET.



Το script του διακομιστή (1/2)

```
//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;
```



Το script του διακομιστή (1/2)

```
// 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";
.....
```



Μπορούμε στο διακομιστή να χρησιμοποιήσουμε και ΒΔ

Select a customer: ▼

Customer info will be listed here...

Μόλις επιλεγθεί ένα όνομα, χωρίς να πατηθεί κάποιο άλλο πλήκτρο...

Alfreds Futterkiste ▼

CustomerID	ALFKI
CompanyName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany



Η σελίδα μας

- Χρησιμοποιούμε μια φόρμα στην οποία έχουμε ένα select box.
- Έχουμε συνδέσει το γεγονός onchange του select box με τη συνάρτηση showCustomer(str).

```
<select name="customers"  
  onchange="showCustomer(this.value)">
```

- Κάθε φορά που γίνεται λοιπόν η αλλαγή καλείται η showCustomer με παράμετρο το όνομα που έχει επιλεχθεί.



Η συνάρτηση showCustomer()

```
function showCustomer(str)
{
  var xmlhttp;
  if (str=="")
  {
    document.getElementById("txtHint").innerHTML="";
    return;
  }
  if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
  } else
  { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
  xmlhttp.onreadystatechange=function()
  {
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
      document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
    }
  }
  xmlhttp.open("GET","getcustomer.php?q="+str,true);
  xmlhttp.send();
}
```



Η συνάρτηση showCustomer()

- Η συνάρτηση λοιπόν:
 - Ελέγχει αν έχει κληθεί με κάποια μη μηδενική συμβολοσειρά (string) .
 - Δημιουργεί το αντικείμενο XMLHttpRequest .
 - Δημιουργεί τη συνάρτηση χειρισμού.
 - Στέλνει το αίτημα στο διακομιστή.
 - Παρατηρήστε ότι επιλέξαμε να χρησιμοποιήσουμε τη μέθοδο GET και την παράμετρο q με την επιλογή μας .



Μπορούμε στο διακομιστή να χρησιμοποιήσουμε και ΒΔ

```
<?php
$q=$_GET["q"];
$con = mysql_connect('localhost', 'peter', 'abc123');
if (!$con) { die('Could not connect: ' . mysql_error()); }
mysql_select_db("ajax_demo", $con);
$sql="SELECT * FROM user WHERE id = '". $q ."'";
$result = mysql_query($sql);
echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
<th>Hometown</th>
<th>Job</th>
</tr>";
while($row = mysql_fetch_array($result))
{ echo "<tr>";
echo "<td>" . $row['FirstName'] . "</td>";
echo "<td>" . $row['LastName'] . "</td>";
echo "<td>" . $row['Age'] . "</td>";
echo "<td>" . $row['Hometown'] . "</td>";
echo "<td>" . $row['Job'] . "</td>";
echo "</tr>"; }
echo "</table>";
mysql_close($con);?>
```



Περισσότερα παραδείγματα

Επισκεφτείτε τη σελίδα

http://www.w3schools.com/ajax/ajax_examples.asp

Για περισσότερα παραδείγματα

