



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

---

## Προγραμματισμός στο Διαδίκτυο

### AJAX – jQuery – GoogleMaps - Google Charts – Javascript API

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

---



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## 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.



## Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Προγραμματισμός στο Διαδίκτυο

Ενότητα 4β – AJAX-jQuery-  
GoogleMaps-Google Charts  
Javascript API

Μανώλης Μαραγκουδάκης

Πανεπιστήμιο Αιγαίου  
Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

# Εισαγωγή



AJAX=Asynchronous JavaScript and XML

συνδυασμός και χρήση της JavaScript και της XML

- χωρίς να πρόκειται για μία νέα 'γλώσσα' ή τεχνολογία

Σκοπός είναι η αύξηση της δυναμικότητας της αλληλεπίδρασης (interactivity) στην πλευρά του πελάτη μεταξύ χρήστη και browser

Μέχρι στιγμής ξέραμε ότι και για την ελάχιστη αλλαγή στην πλευρά του χρήστη πρέπει ο server να δημιουργήσει και στείλει μία **ολόκληρη** νέα σελίδα

Με την μεθοδολογία AJAX αυτό θα γίνεται τώρα επιλεκτικά σε διάφορες περιοχές της σελίδας με το υπόλοιπο μέρος της να μένει σταθερό και να μην χρειάζεται ανανέωση από τον server

# Εισαγωγή



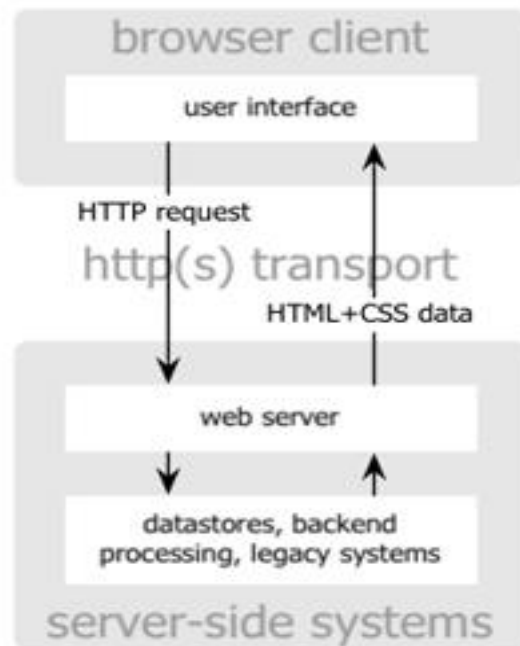
Για να γίνει αυτό, πρέπει να έχουμε πρόσβαση σε κάποιο δομημένο τρόπο που χρησιμοποιεί ο browser για να 'ζωγραφίσει' την σελίδα.

Έχουμε ήδη δει πως οι μοντέρνοι browsers κατασκευάζουν πρώτα το 'DOM της σελίδας'.

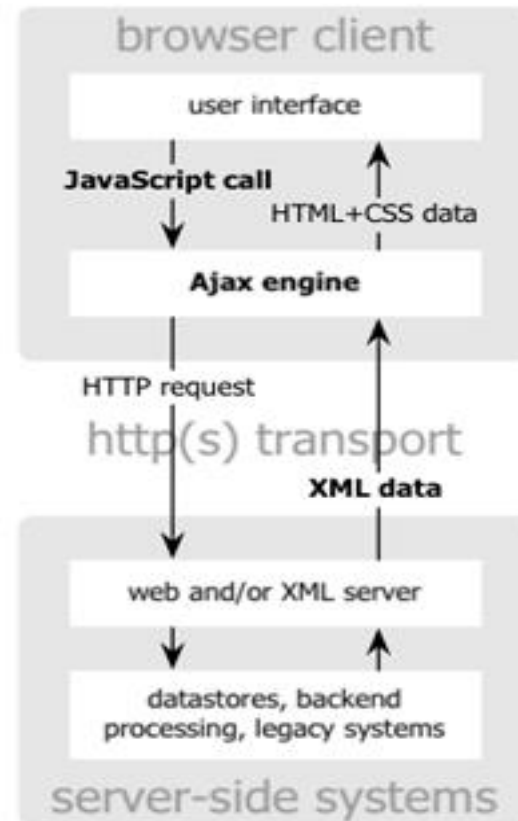
- Ο browser λαμβάνει κείμενο html, το εκλαμβάνει σαν xml και δημιουργεί στην μνήμη του το σχετικό XHTML DOM.
- Γι' αυτό και επιμένουμε στη αυστηρή χρήση της XHTML και όχι απλής HTML.

Μπορούμε όμως τώρα να επεμβαίνουμε επιλεκτικά σε αυτό το DOM και να επηρεάζουμε την τελική εμφάνιση και λειτουργικότητά της σελίδας, όπου και όπως θέλουμε, χωρίς να πειράζουμε το υπόλοιπό της.

# Μοντέλο web εφαρμογών



classic  
web application model

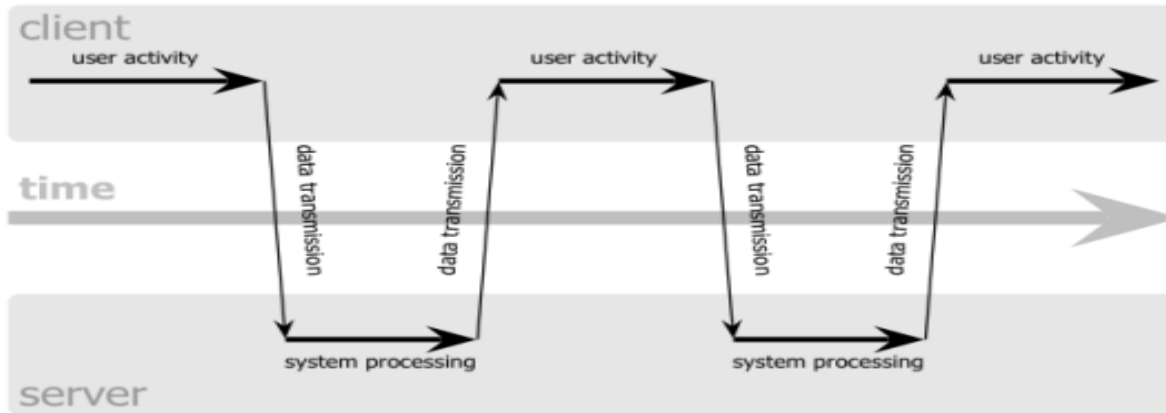


Ajax  
web application model

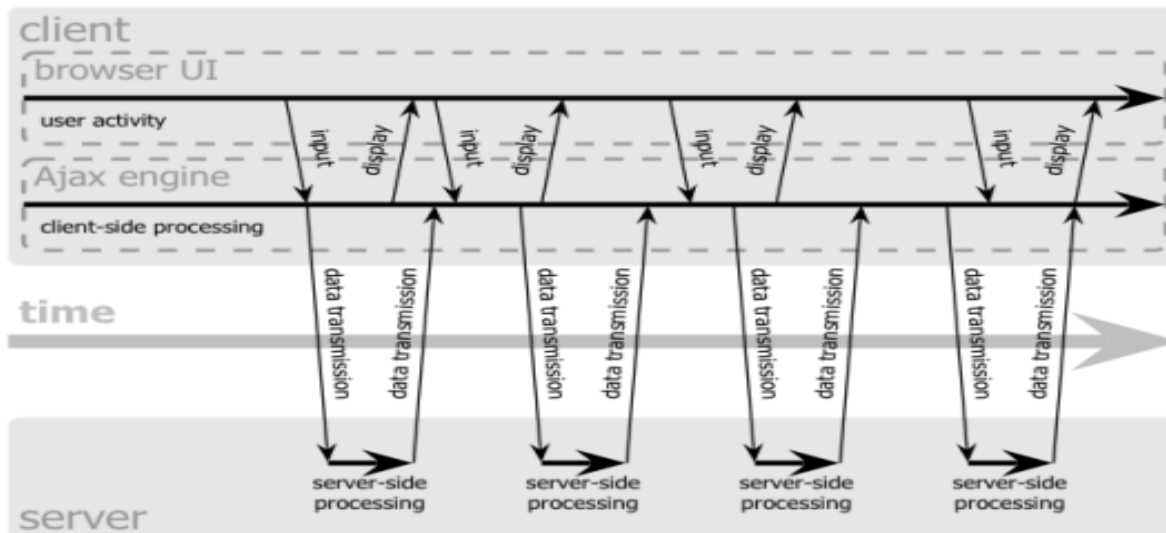
# Μοντέλο επικοινωνίας



## classic web application model (synchronous)



## Ajax web application model (asynchronous)





## Το αντικείμενο XMLHttpRequest

- Όλοι οι νέοι Browsers υποστηρίζουν το Javascript αντικείμενο XMLHttpRequest
  - Εξαίρεση αποτελούν οι εκδόσεις του IE πριν την 7
  - Οι εκδόσεις αυτές παρέχουν υποστήριξη μέσω αντικειμένου ActiveX
- Το αντικείμενο XMLHttpRequest διαθέτει ένα σύνολο ιδιοτήτων, μεθόδων και γεγονότων τα οποία χρησιμοποιεί ο προγραμματιστής για την υλοποίηση της ασύγχρονης επικοινωνίας



# Αρχικοποίηση



```
var xmlhttp;  
if (window.XMLHttpRequest){  
    // code for IE7+, Firefox, Chrome, Opera, Safari  
    xmlhttp=new XMLHttpRequest();  
}  
else if (window.ActiveXObject){  
    // code for IE6, IE5  
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
}  
else {  
    alert("Your browser does not support XMLHttpRequest!");  
}
```

# Παράδειγμα



```
1 <html>
2 <head>
3 <script type="text/javascript">
4   function loadXMLDoc()
5   {
6     var xmlhttp;
7     if (window.XMLHttpRequest)
8     { // code for IE7+, Firefox, Chrome, Opera, Safari
9       xmlhttp=new XMLHttpRequest();
10    }
11    else
12    { // code for IE6, IE5
13      xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
14    }
15
16
17    //GET or POST specifies the request, url to server, page therein, true/false is for synchronous/asynchronous
18    //Three cases: plain text, text produced by servlet, xml - Comment out one out of three
19    xmlhttp.open("GET","SomeText.txt",true); //plain text
20    //xmlhttp.open("GET","http://localhost:8080/myServletDir/firstServletDemo",true); //servlet text
21    //xmlhttp.open("GET","http://localhost:8080/myServletDir/someXML.xml",true); //xml
22
23    xmlhttp.onreadystatechange=function()
24    {
25      if (xmlhttp.readyState==4 && xmlhttp.status==200)
26      {
27        document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
28      }
29    }
30
31    xmlhttp.send(); //in the case of post: xmlhttp.send(string_to_be_sent_as_body_of_POST)
32  }
33 </script>
34 </head>
35 <body>
36 <div id="myDiv" style="border: 1px solid black"><h2>Let AJAX change this section only!</h2></div>
37 <button type="button" onclick="loadXMLDoc()">Change Content</button>
38 </body>
```

**Let AJAX change this section only!**

Change Content

12345...Done

Change Content

# Ιδιότητες του XMLHttpRequest



- **readyState**: Επιστρέφει την τρέχουσα κατάσταση της λειτουργίας του αιτήματος
- **responseText**: Επιστρέφει το σώμα της απόκρισης σαν αλφαριθμητικό.
- **responseXML**: Επιστρέφει το σώμα της απόκρισης σαν ένα αντικείμενο XML Document.
- **status**: Επιστρέφει τον HTTP κωδικό κατάστασης του αιτήματος.
- **statusText**: Επιστρέφει την φιλική μορφή (κείμενο) της HTTP κατάστασης του αιτήματος.

# Το προηγούμενο παράδειγμα με XML



- Χρήση της responseXML

Let AJAX change this section only!

Change Content

Toyota  
Opel  
VW

Change Content

```
<vehicles>
<car>Toyota</car>
<car>Opel</car>
<car>VW</car>
</vehicles>
```

# Μέθοδοι και Γεγονότα του XMLHttpRequest



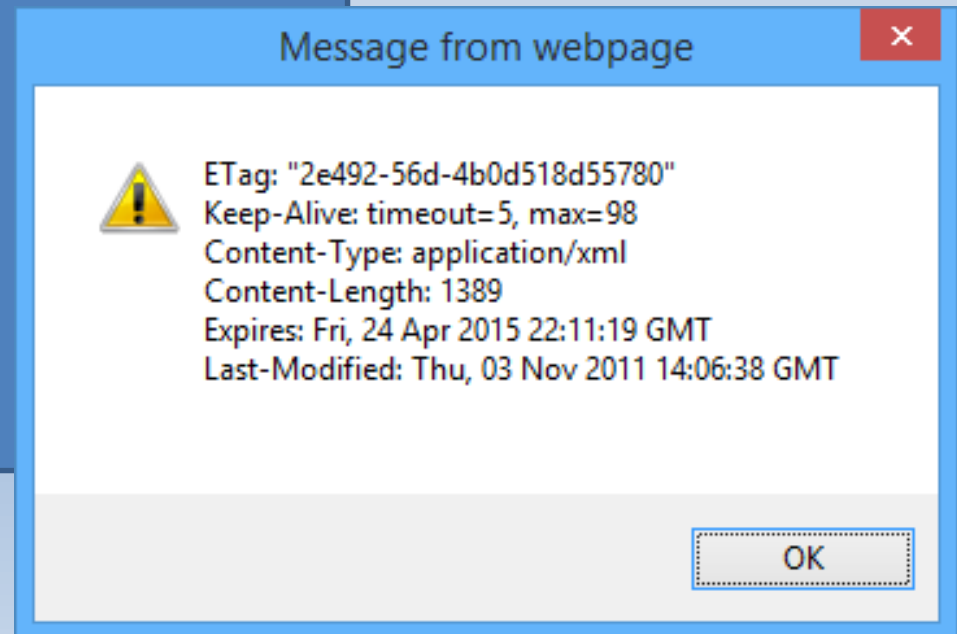
```
xmlhttp.open("GET","ajax_info.txt",false);  
xmlhttp.send();  
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
```

- **abort():** Απορρίπτει το xmlhttp αίτημα.
- **getAllResponseHeaders():** Επιστρέφει τις τιμές όλων των επικεφαλίδων σαν αλφαριθμητικό.
- **getResponseHeader(header):** Επιστρέφει την τιμή της συγκεκριμένης επικεφαλίδας.
- **open(Method,URL,Async):** Δίνει τιμές σε παραμέτρους τις επικοινωνίας όπου θα χρησιμοποιηθεί το αντικείμενο XMLHttpRequest.
  - Method: GET, POST
  - URL: τοποθεσία ενός αρχείου στο server (txt, XML, Servlet)
  - Async: True (ασύγχρονη), False (σύγχρονη)
- **send(postData):** Στέλνει το HTTP αίτημα στον server.
- **setRequestHeader(header,value):** Προσθέτει επιπλέον HTTP επικεφαλίδες στο αίτημα.
- **onreadystatechange:** καθορίζει τον χειριστή γεγονότων που θα κληθεί όταν αλλάξει η κατάσταση ready state.

# Παράδειγμα getAllResponseHeaders



```
function Test_ResponseHeaders ()
{
if (httpRequest.getAllResponseHeaders)
{
var headers = httpRequest.getAllResponseHeaders
(); alert (headers);
}
else
{
alert ("Your browser doesn't support
getAllResponseHeaders method");
}
}
```



# onreadystatechange



- Αποθηκεύει μια συνάρτηση ή το όνομα μιας συνάρτησης που καλείται αυτόματα κάθε φορά που το readyState αλλάζει
- Καταστάσεις readyState
  - 0: request not initialized
  - 1: server connection established
  - 2: request received
  - 3: processing request
  - 4: request finished and response is ready
- Status
  - 200: “OK”
  - 404: Page not found

# Παράδειγμα με auto-complete



Start typing a name in the input field below:

First name:  x

Suggestions: Anna , Amanda

```
1 <html>
2 <head>
3 <script>
4 function showHint(str) {
5     if (str.length == 0) {
6         document.getElementById("txtHint").innerHTML = "";
7         return;
8     } else {
9         var xmlhttp = new XMLHttpRequest();
10        xmlhttp.onreadystatechange = function() {
11            if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
12                document.getElementById("txtHint").innerHTML = xmlhttp.responseText;
13            }
14        }
15        xmlhttp.open("GET", "gethint.asp?q=" + str, true);
16        xmlhttp.send();
17    }
18 }
19 </script>
20 </head>
21 <body>
22
23 <p><b>Start typing a name in the input field below:</b></p>
24 <form>
25 First name: <input type="text" onkeyup="showHint(this.value)">
26 </form>
27 <p>Suggestions: <span id="txtHint"></span></p>
28 </body>
29 </html>
```



# Παράδειγμα με auto-complete



```
1 <?
2 response.expires=-1
3 dim a(30)
4 'Fill up array with names
5 a(1)="Anna"
6 a(2)="Brittany"
7 a(3)="Cinderella"
8 a(4)="Diana"
9 a(5)="Amanda"
10 'get the q parameter from URL
11 q=ucase(request.querystring("q"))
12
13 'lookup all hints from array if length of q>0
14 if len(q)>0 then
15     hint=""
16     for i=1 to 30
17         if q=ucase(mid(a(i),1,len(q))) then
18             if hint="" then
19                 hint=a(i)
20             else
21                 hint=hint & " , " & a(i)
22             end if
23         end if
24     next
25 end if
26
27 if hint="" then
28     response.write("no suggestion")
29 else
30     response.write(hint)
31 end if
32 >
```

Start typing a name in the input field below:

First name:  ×

Suggestions: Anna , Amanda

```
1 <?php
2 // Array with names
3 $a[] = "Anna";
4 $a[] = "Brittany";
5 $a[] = "Cinderella";
6 $a[] = "Diana";
7 $a[] = "Amanda";
8 // get the q parameter from URL
9 $q = $_REQUEST["q"];
10 $hint = "";
11
12 // lookup all hints from array if $q is different from ""
13 if ($q !== "") {
14     $q = strtolower($q);
15     $len=strlen($q);
16     foreach($a as $name) {
17         if (strstr($q, substr($name, 0, $len))) {
18             if ($hint === "") {
19                 $hint = $name;
20             } else {
21                 $hint .= ", $name";
22             }
23         }
24     }
25 }
26
27 echo $hint === "" ? "no suggestion" : $hint;
28 ?>
```

Το ίδιο με PHP,  
πριν θα  
αλλάζαμε μόνο  
το .asp σε .php



- Βιβλιοθήκη Javascript
  - Ελαφριά, γρήγορη, με πλούσια χαρακτηριστικά
  - Σκοπός: απλούστευση της χρήσης της JavaScript
  - Χαρακτηριστικά που προσφέρει
    - Χειρισμός HTML/DOM, CSS, γεγονότων
    - Effects and animations
    - Ajax
    - JSON parsing
    - Utilities
    - Browser compatibility
    - Επεκτασιμότητα

# Προσθήκοντας jQuery



- Τοπικά από το site

```
<head>  
<script src="jquery-1.11.1.min.js"></script>  
</head>
```

Το "type="text/javascript"" δεν χρειάζεται στην HTML5

- Include από CDN (Content Delivery Network)

```
<head>  
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">  
</script>  
</head>
```

# Συντακτικό jQuery



- Το βασικό συντακτικό είναι
  - **`$(selector).action()`**
    - Με το `$` αποκτούμε πρόσβαση στο jQuery
    - Με το `(selector)` εντοπίζουμε τα επιθυμητά HTML elements
    - `action()` η ενέργεια που εκτελείται στα στοιχεία
    - Παράδειγμα  
`$("p").hide()` – κρύβει όλα τα `<p>` στοιχεία.

- Που τοποθετούμε τον κώδικα

```
<script>  
$(document).ready(function(){  
  // jQuery methods go here...  
});  
</script>
```

Σιγουρεύει ότι η συνάρτηση θα τρέξει αφού φορτωθεί το DOM του εγγράφου

# Selectors



- element Selector (Επιλογή με βάση το όνομα)  
`$("p")`
- #id Selector (Επιλογή με βάση το id)  
`$("#myId")`
- .class Selector (Επιλογή με βάση την κλάση)  
`$(".myClass");`
- Attribute Selector (Επιλογή με βάση attribute)  
`$("input[name='first_name']");`
- Pseudo-Selectors  
`$("tr:odd"); //επιλέγει όλα τα περριτά <tr> elements`

# Events ...



- Το jQuery προσφέρει μεθόδους για να αντιστοιχίζουμε συναρτήσεις χειρισμού γεγονότων σε ενέργειες (π.χ. `.click()`, `.focus()` κλπ)
- Παραδείγματα
  - // Event setup using a convenience method

```
$( "p" ).click(function() {  
    console.log( "You clicked a paragraph!" );  
});
```
  - // Equivalent event setup using the `.on()` method

```
$( "p" ).on( "click", function() {  
    console.log( "click" );  
});
```

## ... Events



- Μέσα στην συνάρτηση χειρισμού γεγονότων
  - Η συνάρτηση λαμβάνει ένα αντικείμενο, το οποίο περιέχει πληροφορίες και μεθόδους σχετικά με το γεγονός
    - .preventDefault() method
    - pageX, pageY, type, data, target

- Αντιστοίχιση περισσότερων handlers

```
$( "p" ).on({  
    "click": function() { console.log( "clicked!" ); },  
    "mouseover": function() { console.log( "hovered!" ); }  
});
```

# Ajax και jQuery



- \$.ajax() method

`$.ajax({name:value, name:value, ... })`

- Options

- async
    - data
    - type
    - url

```
$("#button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
        $("#div1").html(result);
    }});
});
```



# jQuery παραδείγματα



- Chaining

- [http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery\\_chaining](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_chaining)

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color",
"red").slideUp(2000).slideDown(2000);
  });
});
</script>
```

# jQuery παραδείγματα



- Get Content

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>
```

# jQuery παραδείγματα



- Add elements

```
<script>
function afterText() {
  var txt1 = "<b>I </b>";           // Create element with HTML
  var txt2 = $("<i></i>").text("love "); // Create with jQuery
  var txt3 = document.createElement("b"); // Create with DOM
  txt3.innerHTML = "jQuery!";
  $("img").after(txt1, txt2, txt3); // Insert new elements after img
}
</script>
```



# jQuery παραδείγματα

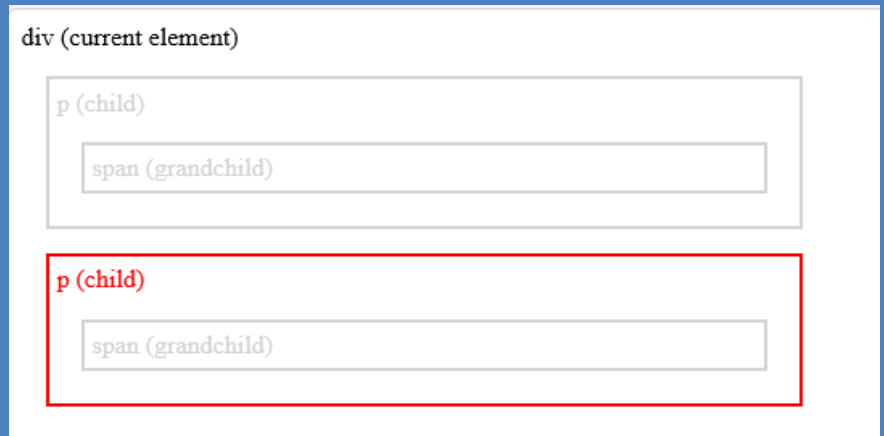


- Traversing DOM

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#div").children("p.2").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p class="1">p (child)
    <span>span (grandchild)</span>
  </p>
  <p class="2">p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
```



# Google Charts



```
<html>
<head>
  <!--Load the AJAX API-->
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">

    // Load the Visualization API and the piechart package.
    google.load('visualization', '1.0', {'packages':['corechart']});

    // Set a callback to run when the Google Visualization API is loaded.
    google.setOnLoadCallback(drawChart);

    // Callback that creates and populates a data table,
    // instantiates the pie chart, passes in the data and
    // draws it.
    function drawChart() {

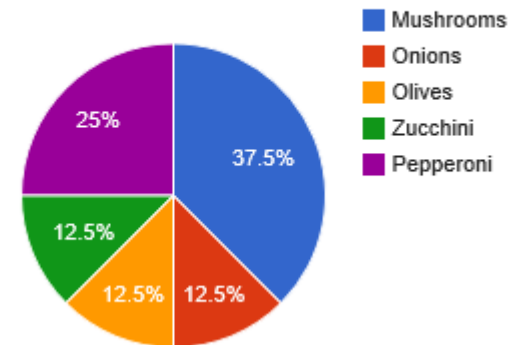
      // Create the data table.
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Topping');
      data.addColumn('number', 'Slices');
      data.addRows([
        ['Mushrooms', 3],
        ['Onions', 1],
        ['Olives', 1],
        ['Zucchini', 1],
        ['Pepperoni', 2]
      ]);

      // Set chart options
      var options = {'title':'How Much Pizza I Ate Last Night',
                    'width':400,
                    'height':300};

      // Instantiate and draw our chart, passing in some options.
      var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>

<body>
  <!--Div that will hold the pie chart-->
  <div id="chart_div"></div>
</body>
</html>
```

How Much Pizza I Ate Last Night



# Βασικά στάδια



- Χρειάζονται 3 βιβλιοθήκες:
  - Google JSAPI API
  - Google Visualization library
  - Τη βιβλιοθήκη που χρειάζεται το κάθε chart

```
<!--Load the AJAX API-->
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">

  // Load the Visualization API library and the piechart library.
  google.load('visualization', '1.0', {'packages':['corechart']});
  google.setOnLoadCallback(drawChart);
  // ... draw the chart...
</script>
```

# Προετοιμάζοντας τα δεδομένα



```
// ----- Version 1-----
// Add rows + data at the same time
// -----
var data = new google.visualization.DataTable();

// Declare columns
data.addColumn('string', 'Employee Name');
data.addColumn('datetime', 'Hire Date');

// Add data.
data.addRows([
  ['Mike', {v:new Date(2008,1,28), f:'February 28, 2008'}], // Example of specifying actual and formatted values.
  ['Bob', new Date(2007,5,1)], // More typically this would be done using a
  ['Alice', new Date(2006,7,16)], // formatter.
  ['Frank', new Date(2007,11,28)],
  ['Floyd', new Date(2005,3,13)],
  ['Fritz', new Date(2011,6,1)]
]);

// ----- Version 2-----
// Add empty rows, then populate
// -----
var data = new google.visualization.DataTable();
// Add columns
data.addColumn('string', 'Employee Name');
data.addColumn('date', 'Start Date');

// Add empty rows
data.addRows(6);
data.setCell(0, 0, 'Mike');
data.setCell(0, 1, {v:new Date(2008,1,28), f:'February 28, 2008'});
data.setCell(1, 0, 'Bob');
data.setCell(1, 1, new Date(2007, 5, 1));
data.setCell(2, 0, 'Alice');
data.setCell(2, 1, new Date(2006, 7, 16));
data.setCell(3, 0, 'Frank');
data.setCell(3, 1, new Date(2007, 11, 28));
data.setCell(4, 0, 'Floyd');
data.setCell(4, 1, new Date(2005, 3, 13));
data.setCell(5, 0, 'Fritz');
data.setCell(5, 1, new Date(2007, 9, 2));
```

# Παραμετροποιώντας τα charts



```
<html>
  <head>
    <!--Load the AJAX API-->
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript">

      // Load the Visualization API and the piechart package.
      google.load('visualization', '1.0', {'packages':['corechart']});

      // Set a callback to run when the Google Visualization API is loaded.
      google.setOnLoadCallback(drawChart);

      // Callback that creates and populates a data table,
      // instantiates the pie chart, passes in the data and
      // draws it.
      function drawChart() {

        // Create the data table.
        var data = new google.visualization.DataTable();
        data.addColumn('string', 'Topping');
        data.addColumn('number', 'Slices');
        data.addRows([
          ['Mushrooms', 3],
          ['Onions', 1],
          ['Olives', 1],
          ['Zucchini', 1],
          ['Pepperoni', 2]
        ]);

        // Set chart options
        var options = {'title':'How Much Pizza I Ate Last Night',
                      'width':400,
                      'height':300};

        // Instantiate and draw our chart, passing in some options.
        var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    </script>
  </head>

  <body>
    <!--Div that will hold the pie chart-->
    <div id="chart_div" style="width:400; height:300"></div>
  </body>
</html>
```



# Σχεδιάζοντας το γράφημα



```
<html>
<head>
  <!--Load the AJAX API-->
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">

    // Load the Visualization API and the piechart package.
    google.load('visualization', '1.0', {'packages':['corechart']});

    // Set a callback to run when the Google Visualization API is loaded.
    google.setOnLoadCallback(drawChart);

    // Callback that creates and populates a data table,
    // instantiates the pie chart, passes in the data and
    // draws it.
    function drawChart() {

      // Create the data table.
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Topping');
      data.addColumn('number', 'Slices');
      data.addRows([
        ['Mushrooms', 3],
        ['Onions', 1],
        ['Olives', 1],
        ['Zucchini', 1],
        ['Pepperoni', 2]
      ]);

      // Set chart options
      var options = {'title':'How Much Pizza I Ate Last Night',
                    'width':400,
                    'height':300};

      // Instantiate and draw our chart, passing in some options.
      var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
      chart.draw(data, options);

    }
  </script>
</head>

<body>
<!--Div that will hold the pie chart-->
  <div id="chart_div" style="width:400; height:300"></div>

</body>
</html>
```

# Προαιρετικά: βάζοντας event handlers



```
load libraries...

function drawChart() {

  prepare data...

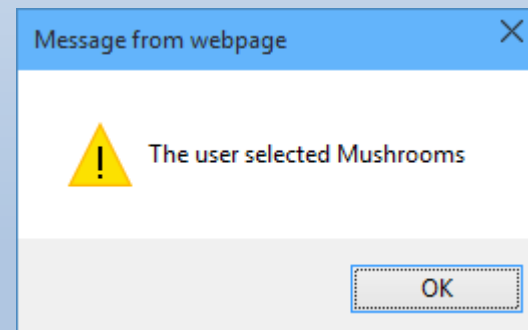
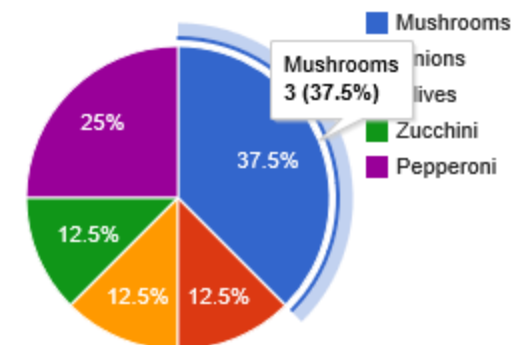
  var chart = new google.visualization.PieChart(document.getElementById('chart_div'))

  // The select handler. Call the chart's getSelection() method
  function selectHandler() {
    var selectedItem = chart.getSelection()[0];
    if (selectedItem) {
      var value = data.getValue(selectedItem.row, selectedItem.column);
      alert('The user selected ' + value);
    }
  }

  // Listen for the 'select' event, and call my function selectHandler() when
  // the user selects something on the chart.
  google.visualization.events.addListener(chart, 'select', selectHandler);

  draw the chart...
}
```

How Much Pizza I Ate Last Night



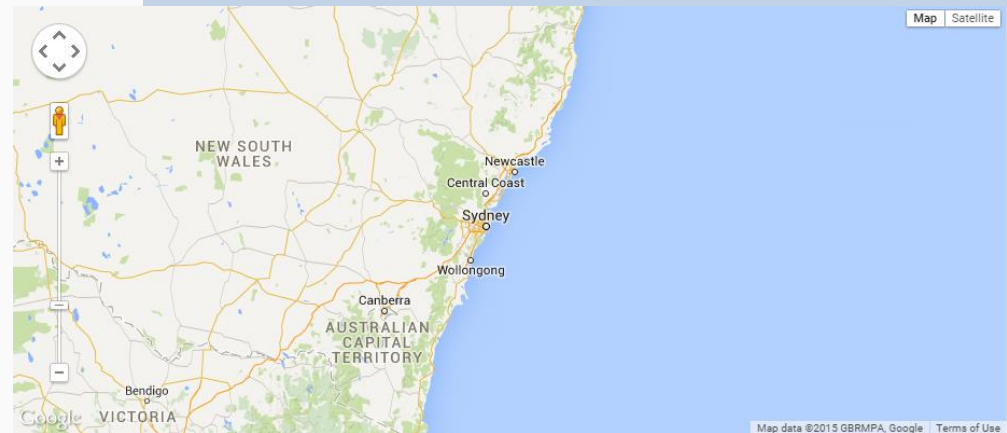
# Google MAPS API



```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <style>
      html, body, #map-canvas {
        height: 100%;
        margin: 0px;
        padding: 0px
      }
    </style>
    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
    <script>
var map;
function initialize() {
  var mapOptions = {
    zoom: 8,
    center: new google.maps.LatLng(-34.397, 150.644)
  };
  map = new google.maps.Map(document.getElementById('map-canvas'),
    mapOptions);
}

google.maps.event.addDomListener(window, 'load', initialize);

    </script>
  </head>
  <body>
    <div id="map-canvas"></div>
  </body>
</html>
```



# Map Types



- MapTypeId.ROADMAP
  - displays the default road map view. This is the default map type.
- MapTypeId.SATELLITE
  - displays Google Earth satellite images
- MapTypeId.HYBRID
  - displays a mixture of normal and satellite views
- MapTypeId.TERRAIN
  - displays a physical map based on terrain information.

```
var myLatLng = new google.maps.LatLng(-34.397, 150.644);
var mapOptions = {
  zoom: 8,
  center: myLatLng,
  mapTypeId: google.maps.MapTypeId.SATELLITE
};
var map = new google.maps.Map(document.getElementById("map-canvas"),
  mapOptions);
```

# Simple Markers



```
function initialize() {  
  var myLatLng = new google.maps.LatLng(-25.363882,131.044922);  
  var mapOptions = {  
    zoom: 4,  
    center: myLatLng  
  }  
  var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);  
  
  var marker = new google.maps.Marker({  
    position: myLatLng,  
    map: map,  
    title: 'Hello World!'  
  });  
}  
  
google.maps.event.addDomListener(window, 'load', initialize);
```

# GeoCoding



```
var geocoder;
var map;
function initialize() {
  geocoder = new google.maps.Geocoder();
  var latlng = new google.maps.LatLng(-34.397, 150.644);
  var mapOptions = {
    zoom: 8,
    center: latlng
  }
  map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
}
```

```
function codeAddress() {
  var address = document.getElementById('address').value;
  geocoder.geocode( { 'address': address}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      map.setCenter(results[0].geometry.location);
      var marker = new google.maps.Marker({
        map: map,
        position: results[0].geometry.location
      });
    } else {
      alert('Geocode was not successful for the following reason: ' + status);
    }
  });
}
```

```
google.maps.event.addDomListener(window, 'load', initialize);
```

