

G5c- (Inf trukea. XML, DOM, AJAX) jQuery eta Ajax



- <http://api.jquery.com/jquery.ajax/>
- http://www.w3schools.com/jquery/jquery_ajax_load.asp
- <http://api.jquery.com/jquery.get/>
- <http://api.jquery.com/jquery.post/>
- <http://api.jquery.com/>
- <http://www.w3schools.com/jquery>
- <http://api.jquery.com/category/ajax/shorthand-methods/>

- **`$.ajax()`**
 - Perform an asynchronous HTTP (Ajax) request.
 - ***Use this if you need to do something when XHR fails, or you need to specify ajax options (e.g. `cache: false`) on the fly.***



These methods perform the more common types of Ajax requests in less code:

- **`$.get()`**
 - Load data from the server using a HTTP GET request.
 - ***Use this if you want to make a GET call and play extensively with the response***
- **`$.getJSON()`**
 - Load JSON-encoded data from the server using a GET HTTP request.
- **`$.getScript()`**
 - Load a JavaScript file from the server using a GET HTTP request, then execute it.
- **`$.post()`**
 - Load data from the server using a HTTP POST request.
 - ***Use this if you want to make a POST call and don't want to load the response to some container DOM.***
- **`.load()`**
 - Load data from the server and place the returned HTML into the matched element.



jQuery eta Ajax

- AJAX funtzionalitatea lantzeko metodoak ditu jQuery-k.

Posible da:

- Urruneko zerbitzari bati testu soila, HTML, XML edo JSON eskatzea HTTP Get edo HTTP Post erabiliz
- Zuzenean kanpoko datuak kargatzea aukeratutako elementuen barnera
- Metodo orokorra:
 - `$.ajax({name:value, name:value, ... })`



jQuery eta Ajax: *load* metodoa

- `load()` metodoak zerbitzari bateko datuak kargatzen ditu, eta datu horiek aukeratutako elementu batean jartzen ditu
- Sintaxia: `$(hautatzailea).load(URLa,datuak,atzeradeia);`
 - `URLa` derrigorrezkoa da. Kargatu nahi den URLa adierazten du
 - `datuak` hautazkoa da. gako/balio bikoteen multzo bat adierazten du, eskaerarekin batera bidaltzeko
 - `atzeradeia` hautazkoa da. Funtzio baten izena da, eta `load()` metodoa burututakoan exekutatzen da funtzio hori
 - Funtzioak eduki ohi dituen parametroak:
 - `responseTxt` - karga ondo burutu bada, erantzunaren testua
 - `statusTxt` - eskaeraren egoera
 - `xhr` - XMLHttpRequest objektua



jQuery eta Ajax: *get* metodoa

- `$.get()` metodoak zerbitzariko datuak eskatzen ditu HTTP GET eskaera-metodoaz
- Sintaxia: `$.get(URLa, atzeradeia, mota);`
 - **Atzeradeia** funtzioak bi parametro ditu: lehenengoa datuak eta bigarrena estatusa.
 - **mota** hautazkoa da. "`xml`", "`html`", "`json`", "`text`"



jQuery eta Ajax: *post* metodoa

- `$.post()` metodoak zerbitzariko datuak eskatzen ditu HTTP POST eskaera-metodoaz
- Sintaxia: `$.post(URLa, datuak, atzeradeia, mota);`
 - **URLa** derrigorrezkoa da. Kargatu nahi den URLa adierazten du
 - **datuak** hautazkoa da. gako/balio bikoteen multzo bat adierazten du, eskaerarekin batera bidaltzeko
 - **atzeradeia** hautazkoa da. Funtzio baten izena da, eta post() metodoa ondo burutzen bada exekutatzen da funtzio hori
 - Funtzioak eduki ohi dituen parametroak:
 - erantzuna - karga ondo burutu bada, erantzunaren testua
 - egoera - eskaeraren egoera
 - **mota** hautazkoa da. "`xml`", "`html`", "`json`", "`text`"

.load()

Adib1 efektu
berdina

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Adibidea1</title>
5 <script type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
6 <script language = "javascript">
7
8 function datuakEskatu()
9 {
10 $("#emaitza").load("datuak.txt");
11 }
12 </script>
13 </head>
14 <body>
15 <H1>AJAX erabiliz datuak erakutsi</H1>
16 <form> <input type = "button" value = "Mezua azaldu" onclick = "datuakEskatu()">
17 </form>
18 <div id="emaitza" style="background-color:#99FF66;">
19 <p>Testua hemen agertuko da</p>
20 </div>
21 </body>
22 </html>
```

```
<!DOCTYPE html>
<html><head> <title>Adibidea4</title><script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script language = "javascript">
function showPeli(str){
  $('#content').html('<div></div>');
  var jqxhr=$.get("pelikulakoDatuak.php", {q:str},
    function(datuak){
      $('#content').fadeIn(1000).html(datuak);
    });
  jqxhr.fail(function(){
    $('#content').fadeIn().html('<p class="error">
      <strong> Zerbitzariak ez duela erantzuten dirudi</p>');
  })
}
</script> </head>
<body> <br>
<form> Titulu bat aukera ezazu:
<select name="pelis" onchange="showPeli(this.value)">
  <option value="">.....</option>
  <option value="La vida de los otros">La vida de los otros</option>
  <option value="Milenium I">Milenium I</option>
</select>
</form>
<div id="content"><b>Pelikulako datuen bistaratzea hemen ...</b></div>
</body></html>
```

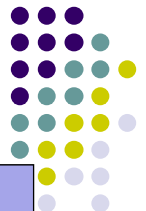
.get() Adib4-6
efektu antzekoa
(Adib6 ere)

```

<!DOCTYPE html>
<html><head> <title>Adibidea4</title><script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.js">
</script>
<script language = "javascript">
function adibAjaxMetodoaShowPeli(str){
$.ajax({
    url: 'pelikulakoDatuak.php?q='+ str,
    beforeSend:function(){$('#content').html('<div>  </div>')},
    success:function(datuak){$('#content').fadeIn().html(datuak);},
    error:function(){
        $('#content').fadeIn().html('<p class="error"><strong>
Zerbitzariak ez duela erantzuten dirudi</p>');
    }
});
}
</script> </head>
<body> <br>
<form> Titulu bat aukera ezazu:
<select name="pelis" onchange="adibAjaxMetodoaShowPeli(this.value)">
    <option value="">.....</option>
    <option value="La vida de los otros">La vida de los otros</option>
    <option value="Milenium I">Milenium I</option>
</select>
</form>
<div id="content"><b>Pelikulako datuen bistaratzea hemen ...</b></div>
</body></html>

```

.ajax erabiliz
funtzionalitatea
berdina



Uso del método **.getJSON**
para visualizar fotos
obtenidas del portal *flickr*.

```

<html lang="en">
<head> <meta charset="utf-8">
<title>jQuery.getJSON demo</title>
<style> img {height: 100px; float: left; } </style>
<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
</head>
<body>
<div id="images"></div>
<script>
(function()
{var flickerAPI = "http://api.flickr.com/services/feeds/photos_public.gne?jsoncallback=?";
$.getJSON( flickerAPI, {tags: "paisajes", tagmode: "any", format: "json" })
.done(function( data )
{
    $.each( data.items, function( i, item ) {
        if ( i == 10 ) {return false;}
        else
        {
            alert (item.title);
            $( "<img>" ).attr( "src", item.media.m ).appendTo( "#images" );
        }
    });
});})();
</script>
</body>
</html>

```



.getJSON (.....) bidez bezeroak jasotzen dituen datuak

```
{
  "title": "Talk On Travel Pool",
  "link": "http://www.flickr.com/groups/talkontravel/pool/",
  "description": "Travel and vacation photos from around the world, featured on the Talk On Travel blog. Join us
    in sharing your favourite destinations, your top hot spots, and the places to avoid. ",
  "modified": "2009-02-02T11:10:27Z",
  "generator": "http://www.flickr.com/",
  "items": [
    {
      "title": "View from the hotel",
      "link": "http://www.flickr.com/photos/33112458@N08/3081564649/in/pool-998875@N22",
      "media": {"m": "http://farm4.static.flickr.com/3037/3081564649_4a6569750c_m.jpg"},
      "date_taken": "2008-12-04T04:43:03-08:00",
      "description": "<p><a href='\"http://www.flickr.com/people/33112458@N08/\"> Talk On Travel</a>    has added a
        photo to the pool:</p><p><a href='\"http:// www.flickr.com/photos/33112458@N08/3081564649/\" title='\"View from the
        hotel\"> <img      src='\"http://farm4.static.flickr.com/3037/3081564649_4a6569750c_m.jpg\" width='\"240\"   height='\"180\"
        alt='\"View from the hotel\" /></a></p> ",
      "published": "2008-12-04T12:43:03Z",
      "author": "nobody@flickr.com (Talk On Travel)",
      "author_id": "33112458@N08",
      "tags": "spain dolphins tenerife canaries lagomera aqualand playadelasamericas junglepark losgigantos
        loscristines talkontravel"
    }
  ]
}
```



Beste adibide bat: berrien bistaratzea (RSS)

Adibide interesgarriak, besteak beste, RSSarena

- <http://kyleschaeffer.com/development/the-perfect-jquery-ajax-request/>



Cachea-rekin arazoak

- Nabigatzaileen portaera ez da berdina zerbitzariari eskaera bat egitean
- Portaera modifika liteke esplizituki eskatuaz gorde dezala/ez dezala gorde eskaera cachean
- PHPen esplizituki esan diezaiokegu emaitza cachean meta EZ dezala:

```
<?php
header("Control-cache: no-store, no-cache, must-revalidate");
... PHP aplikazioa...
?>
```

HTTP response-ren header/buruko bat itzuliko du cachean ez dela gorde azalduz



Cachea-rekin arazoak (II)

- Bezeroaren atalean, JavaScript erabiliz, php bat inbokatzen ez denean (adibidez, .xml bat ekarri nahi delako), cachearen arazoa egon ez dadin, beti balio desberdinak itzultzen dituen parametro fiktizio/estra bat gehitzen da:

```
XHR.open("GET", "contador.xml?q="+ new Date().getTime());
```

- jQuery-rekin aritzean, \$.ajax metodoa erabili liteke. Metodo honek cache parametro bat du non haren balioaren arabera (true, false), cachean metatzen da edo ez emaitza.

```
$.ajax({
  url : '/bidea/nire/fitxategira/fitxategia.json',
  dataType : 'json',
  data : data,
  cache : false,
  success : function(data) {
  }
});
```



url-encoding aspergarriekin zer?

- “bat=” + document.getElementById(“izena”.value + “&bi=” + document.getElementById(“abizena1”.value + “&hiru=” + document.getElementById(“abizena2”.value + Eta beste 20
- jQuery-k badu soluzioa: **.serialize()**
- <http://www.jose-aguilar.com/blog/obtener-los-datos-de-un-formulario-con-serialize/>



Garapena AJAXen oinarrituz egiteko erreminta lagungarriak

- Noski, JavaScript zuzenean erabiliz progama liteke, baina metodo hori ez da “eskalagarriak”*, beharrezkoa izanik bestelako erreminta, garapen ingurunetara,... jotzea.
 - <http://yuilibary.com/>
 - /connection
 - <http://www.dojotoolkit.org/>
 - <https://mochi.github.io/mochikit/>
 - <https://script.aculo.us/>
- MS produktuen munduan aldiz:
 - <http://www.asp.net/ajax>

* “Zerbaiten handitze azkarra eta gehienetan kezkarria ...”