

Beginning Ajax

سید کاوه احمدی

Ajax

- Asynchronous JavaScript and XML
 - توسط Jeese James Garrett معرفی شده است.
 - Ajax یک تکنولوژی جدید نیست.
 - بلکه یک اصطلاح است که به ترکیب چهار تکنولوژی مختلف با یکدیگر اشاره می کند.

عناصر اصلی Ajax

- جاوااسکریپت
- شیوه‌نامه‌های آبخاری (CSS)
- Document Object Model (DOM)
- شی XMLHttpRequest

استفاده از شی XMLHttpRequest

- استفاده از این شی سه مرحله دارد:
1. ساخت شی XMLHttpRequest
 2. ارسال درخواست به سرور
 3. دریافت پاسخ از سرور

ساخت شی XMLHttpRequest

- `new XMLHttpRequest();`

– در مرورگرهای جدید نظیر فایرفاکس و اینترنت اکسپلورر ۷

- `new ActiveXObject('Msxml2.XMLHTTP');`

– در اینترنت اکسپلورر ۶

ساخت شی XMLHttpRequest

```
var xml;  
  
try {  
    xml = new XMLHttpRequest();  
} catch (e) {  
    xml = new ActiveXObject('Msxml2.XMLHTTP');  
}
```

متدهای شی XMLHttpRequest

<code>abort</code>	درخواست را لغو می کند
<code>open</code>	یک درخواست به سرور را باز می کند
<code>send</code>	یک درخواست به سرور ارسال می کند
<code>setRequestHeader</code>	یک نام / مقدار را به هدر ارسالی اضافه می کند

خصوصیت های شی XMLHttpRequest

<code>onreadystatechange</code>	the event handler that fires at every state change
<code>readyState</code>	the state of the request
<code>responseText</code>	the response body as string
<code>responseXML</code>	the response body as xml
<code>status</code>	the HTTP status code returned by a request
<code>statusText</code>	the text version of the HTTP status code

```
// create the request object
var xml;

try {
    xml = new XMLHttpRequest();
} catch (e) {
    xml = new ActiveXObject('Msxml2.XMLHTTP');
}

// open the socket
xml.open('get', 'data.html', true);

// watch for when the state of the document gets updated
xml.onreadystatechange = function () {
    // wait until the data is fully loaded
    if (xml.readyState == 4) {
        // do something with responseXML, responseText
        // xml.responseXML contains the XML Document
        // xml.responseText contains the response Text
    }
};

// establish the connection
xml.send(null);
```

مقادیر خصوصیت readyState

0 (uninitialized)	open() has not been called yet
1 (loading)	open() has been called, but send() has not been called
2 (loaded)	send() has been called, but the server has not responded yet
3 (interactive)	data is being received from server
4 (complete)	the server's response is complete

statusText , status

status	statusText
200	OK
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error

دریافت پاسخ از سرور: استاتیک

```
xml.open('get', 'data.html', true);

xml.onreadystatechange = function () {
    if (xml.readyState == 4) {
        alert(xml.responseText);
        // You fetched me using Ajax
    }
};

xml.send(null);
```

data.html



You fetched me
using Ajax

دریافت پاسخ از سرور: دینامیک

```
xml.open('get', 'data.php', true);

xml.onreadystatechange = function () {
    if (xml.readyState == 4) {
        alert(xml.responseText);
        // welcome to new world
    }
};

xml.send(null);
```

data.php

```
<?php
print('welcome');
print(' ');
print('to');
print(' ');
print('new');
print(' ');
print('world');
?>
```

processed

```
welcome to new
world
```

ارسال دیتا به سرور از طریق متد GET

```
xml.open('get', 'sum.php?x=1&y=2', true);

xml.onreadystatechange = function () {
    if (xml.readyState == 4) {
        alert(xml.responseText);
        // x+y = 3
    }
};

xml.send(null);
```

sum.php

```
<?php
$x = $_GET['x'];
$y = $_GET['y'];
print('x+y = ');
print($x + $y);
?>
```

processed

```
x+y = 3
```

ارسال دیتا به سرور از طریق متد POST

```
xml.open('post', 'sum_post.php', true);
xml.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');

xml.onreadystatechange = function () {
    if (xml.readyState == 4) {
        alert(xml.responseText);
        // x+y = 3
    }
};

xml.send('x=1&y=2');
```

sum_post.php

```
<?php
    $x = $_POST['x'];
    $y = $_POST['y'];
    print('x+y = ');
    print($x + $y);
?>
```

processed

```
x+y = 3
```

لود ای جکس صفحات

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>JavaScript | Ajax | Example</title>
    </head>

    <body>
        <a href="#" id="page1">Page 1</a>
        <a href="#" id="page2">Page 2</a>
        <a href="#" id="page3">Page 3</a>

        <div id="result"><h1>
            Please Choose a Page
        </h1></div>
        <script type="text/javascript" src="script.js"></script>
    </body>

</html>
```


لود ای جکس صفحات

```
var m = document.getElementById('page1'),
    p = document.getElementById('page2'),
    k = document.getElementById('page3');

m.onclick = function () {
    sendRequest('page1');
};

p.onclick = function () {
    sendRequest('page2');
};

k.onclick = function () {
    sendRequest('page3');
};
```

لود ای جکس صفحات

```
function sendRequest(id) {
    var xml;

    try {
        xml = new XMLHttpRequest();
    } catch (e) {
        xml = new ActiveXObject('Msxml2.XMLHTTP');
    }

    xml.open('get', id + '.html', true);
    xml.onreadystatechange = function () {
        if (xml.readyState == 4) {
            document.getElementById('result').innerHTML = xml.responseText;
        }
    };
    xml.send(null);
}
```

لود ای جکس صفحات با jQuery

▪ جی کوئری، تابعی به اسم `$.ajax` برای کار با برنامه‌های ای جکس دارد.

– این تابع یک پارامتر می‌پذیرد که جنس این پارامتر، از نوع شی می‌باشد.

– به وسیله خاصیت‌ها و متدهایی این شی می‌توان به کنترل حرفه‌ای و ساده ای جکس رسید.

```
$('#[id ^= "page"]').click(function () {
    $.ajax({
        async: false,
        type: 'get',
        url: 'AjaxPages/' + $(this).attr('id') + '.html',
        dataType: 'text',
        success: function (h) {
            $('#result').html(h);
        }
    });
});
```

لود ای جکس صفحات با jQuery

▪ تابع `load`

```
$('#[id ^= "page"]').click(function () {
    $('#result').load($(this).attr('id') + '.html');
});
```

تنظیمات بیشتر

- جمع دو عدد ارسال شده و چاپ نتیجه

```
$('#calculate').click(function () {
    $.ajax({
        async: false,
        type: 'get',
        url: 'sum.php',
        data: "x=5&y=4",
        success: function (html) {
            $('#sum').html(html);
        },
        error: function() {
            alert('sorry, try back later');
        },
        beforeSend: function() {
            $('#sum').html('Loading...')
        },
    });
});
```

sum.php

```
<?php
$x = $_GET['x'];
$y = $_GET['y'];
print('x+y = ');
print($x + $y);
?>
```

تنظیمات بیشتر

- جمع دو عدد ارسال شده و چاپ نتیجه - روشی دیگر

```
$('#calculate').click(function () {
    $.ajax({
        async: false,
        type: 'get',
        url: 'sum.php?x=5&y=4',
        success: function (html) {
            $('#sum').html(html);
        },
        error: function() {
            alert('sorry, try back later');
        },
        beforeSend: function() {
            $('#sum').html('Loading...')
        },
    });
});
```

sum.php

```
<?php
$x = $_GET['x'];
$y = $_GET['y'];
print('x+y = ');
print($x + $y);
?>
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>JavaScript | Ajax | Timeout</title>
  </head>

  <body>
    <div id="c">click on me</div>
    <div id="r"></div>

    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      $.ajax ({
        url: 'data.php',
        timeout: 1000,
        error: function () {
          alert('sorry, try back later');
        }
      });
    </script>
  </body>
</html>

```

delay.php

```

<?php
sleep(5);
echo('After 5 Secends');
?>

```

تنظیم مقادیر پیش فرض برای تمام درخواست ها

■ جی کوئری، تابعی به اسم `$.ajaxSetup` برای این کار دارد.

```

$.ajaxSetup({
  async: false,
  type: 'get',
  beforeSend: function() {
    $('#loading').show()
  }
});

```

توابع سراسری رویداد ای جکس

```
.ajaxComplete()  
.ajaxError()  
.ajaxSend()  
.ajaxStart()  
.ajaxStop()  
.ajaxSuccess()
```

```
$('#log').ajaxComplete(function() {  
    $(this).text('Triggered ajaxComplete handler.');
```

```
});
```

دسترسی ساده به ای جکس

.load()

```
$('#result').load('ajax/test.html', function() {  
    alert('Load was performed.');
```

```
});
```

\$.get()

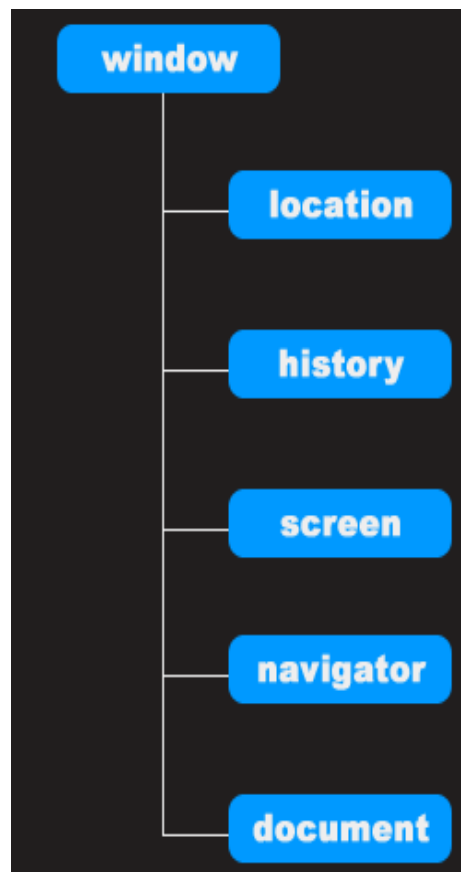
```
$.get("test.php", { name: "John", time: "2pm" } );
```

\$.post()

```
$.post("test.php", { name: "John", time: "2pm" } );
```

ای جکس و لینک‌های ثابت

- یکی از ایراداتی که در برنامه‌های ای جکس به وفور یافت می‌شود، از بین رفتن لینک‌های ثابت درون سایت است.
- مثال قبل را در نظر بگیرید. با کلیک بر روی اسم افراد، صفحه منحصر به فرد آن‌ها به نمایش درمی‌آید. اما اگر کسی بخواهد یگراست به آن صفحه برود، چه‌طور؟! راه حل بسیار ساده است: استفاده از Hash ها.



window.location

```
http://localhost/index.html?y=08#hidecalendar
```

window.location

```
alert(window.location.hash);
```

```
http://localhost/index.html#hidecalendar
```



hash

ای جکس و Hash ها

- همان طور که گفته شد، برای برطرف کردن مشکل لینک‌های ثابت در برنامه‌های ای جکس، از هش‌ها استفاده می‌کنیم.
- بدین ترتیب در اسکریپت‌مان، کدی را اضافه می‌کنیم که ابتدا هش را چک می‌کند. اگر شخصی آدرس سایت را با یک هش داشته باشد، یعنی می‌خواهد از لینک ثابت آن صفحه استفاده کند.
- کاری که ما انجام می‌دهیم این است:
- به جای آن که منتظر بمانیم کاربر درخواست بفرستد، خودمان درخواست را می‌فرستیم.

Old One (Bad)

```
<a href="#" id="page1">Page 1</a>
<a href="#" id="page2">Page 2</a>
<a href="#" id="page3">Page 3</a>
```

New One (Good)

```
<a href="#page1" id="page1">Page 1</a>
<a href="#page2" id="page2">Page 2</a>
<a href="#page3" id="page3">Page 3</a>
```