



```
function AJAXCrearObjeto(){
    var obj;
    if(window.XMLHttpRequest) { // Vemos si es estándar
        obj = new XMLHttpRequest();
    }else { // No es estándar. Es IE o no tiene el objeto
        try {
            obj = new ActiveXObject("Microsoft.XMLHTTP");
        }catch (e){
            alert('El navegador utilizado no está soportado');
        }
    }
    return obj;
}
```

- Llamada a la función anterior:

```
<script>
    oXML = AJAXCrearObjeto();
</script>
```

Enviar la petición (II): Parámetros



- Habitualmente, en la **petición AJAX** pasaremos **parámetros al servidor** indicando qué es lo que necesitamos (como cuando lo hacíamos a través de un formulario):
 - **GET**: usaremos una URL con parámetros (?) en el método `open()` y llamaremos sin parámetros a `send()`
 - **POST**: usaremos el método `POST` en el método `open()` y pasaremos los parámetros desde el método `send()`

```
// Creamos la variable parámetro
provincia = 'Alicante';
// Creamos el objeto
oXML = AJAXCrearObjeto();
... //Alternativa 1

// Preparamos la petición con parámetros
oXML.open('GET', 'listaCiudades.asp?Provincia=' + provincia, true);
// Preparamos la recepción
oXML.onreadystatechange = leerDatos;
// Realizamos la petición
oXML.send('');

... //Alternativa 2
// Preparamos la petición con parámetros
oXML.open('POST', 'listaCiudades.asp', true);
// Preparamos la recepción
oXML.onreadystatechange = leerDatos;
// Realizamos la petición
oXML.send('Provincia=' + provincia);
```

AJAX

Enviar la petición



- Utilizaremos los métodos/propiedades `open()`, `onreadystatechange` y `send()` (**preparar** la petición, seleccionar la **función** que procesará la recepción y **enviar** la petición)
- El método `open()` requiere:
 - Método de petición (GET o POST)
 - URL que se enviará al servidor y mediante la cual se recuperará la respuesta generada por éste (asp, php, jsp...)
- Ejemplo: Solicitud de un archivo de texto empleando GET

```
<script>
    // Creamos el objeto
    oXML = AJAXCrearObjeto();
    // Preparamos la petición
    oXML.open('GET', 'archivo.txt');
    // Preparamos la recepción
    oXML.onreadystatechange = leerDatos; //se verá más adelante
    // Realizamos la petición
    oXML.send(''); // También se puede hacer sin comillas
</script>
```

La respuesta AJAX: XML



- Por definición, AJAX utiliza **XML** para organizar los datos transmitidos entre el servidor y el navegador
- Para que el navegador sea capaz de interpretarlos, tendrá que identificarlos como XML y su contenido tendrá que ser un XML válido (de lo contrario, no serán utilizables). Por tanto, el código del servidor deberá:
 - Especificar cabecera de contenido XML (text/xml)
 - Recomendable enviar también encabezamientos de **control de caché**
 - La respuesta deberá empezar con la declaración de versión:


```
<?xml version="1.0"?>
```
- Ejemplo asp:


```
<%
    response.ContentType="text/xml"
    response.CacheControl="no-cache, must-revalidate"
    response.Expires="Mon, 26 Jul 2008 05:00:00 GMT"
%>
```
- Ejemplo PHP


```
<?php
    header("Content-Type: text/xml");
    header("Cache-Control: no-cache, must-revalidate");
    header("Expires: Mon, 26 Jul 2008 05:00:00 GMT");
?>
```

Ejemplo completo



```
function leerDatos() {
    if (oXML.readyState == 4) {
        var xml = oXML.responseXML.documentElement;
        for (i = 0; i < xml.getElementsByTagName('mensaje').length; i++) {
            var item = xml.getElementsByTagName('mensaje')[i];
            var txt = item.getElementsByTagName('texto')[0].firstChild.data;
            alert(txt); //Cambiar por código que rellene controles de usuario
        }
    }
}

function AJAXCrearObjeto() {
    var obj;
    if(window.XMLHttpRequest) { // no es IE
        obj = new XMLHttpRequest();
    } else { // Es IE o no tiene el objeto
        try {
            obj = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e) {
            alert('El navegador utilizado no está soportado');
        }
    }
    return obj;
}

oXML = AJAXCrearObjeto(); // Recuperamos directamente el archivo XML
oXML.open('get', 'archivo.xml');// Normalmente, llamar a una página de servidor
oXML.onreadystatechange = leerDatos;
oXML.send('');
```