



Ajax: Fundamentos Parte I

Traducción de Dr. Roberto Solís Robles

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Temas

- **Motivación de Ajax**
- **El proceso básico de Ajax**
- **Necesidad de funciones anónimas**
- **Usando contenido dinámico y PHP**
- **Desplegado de resultados en XHTML**



Motivación

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

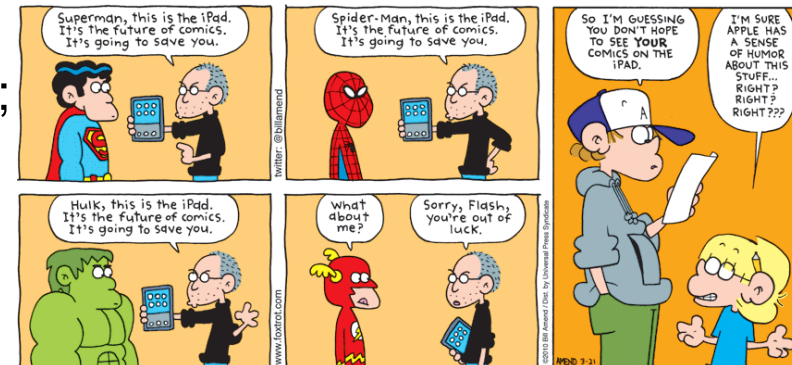
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Por qué Aplicaciones Web?

- **Desventajas de aplicaciones basadas en navegador**
 - GUI es pobre
 - XHTML esta bien para documentos estáticos, pero muy mal para programas
 - Comunicación es ineficiente
 - HTTP es un protocolo pobre para la forma en que usamos ahora las aplicaciones web
- **Entonces por qué todos quieren aplicaciones web?**
 - Acceso universal
 - Todos ya tienen un navegador instalado
 - Cualquier computadora en la red puede acceder al contenido
 - “Actualizaciones” automáticas
 - Contenido viene del servidor, así que nunca se hace obsoleto

Por qué Ajax?

- **Soluciona tres problemas clave de las aplicaciones web**
 - Actualizaciones de grano grueso
 - Síncrono: se congela mientras espera por el resultado
 - Opciones extremadamente limitadas para los widgets (elementos de la GUI)
- **Aun está basado en el navegador**
 - Ajax es algo así como “qué es lo mejor que se puede hacer con lo que todos *ya* tienen en su navegador?”
- **Contenido activo basado en el navegador**
 - Fallaron: Java Applets
 - No son soportados universalmente; no pueden interactuar con XHTML
 - Alternativa seria: Flash/Flex
 - No preinstalado en todas las PC; no disponible en iPhone/iPad
 - Mas nuevos y menos probados:
 - Microsoft Silverlight
 - JavaFX



De Bill Amend and foxtrot.com. © Universal Press Syndicate.

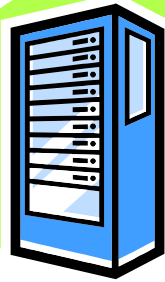
Apps Web Tradicionales vs. Apps Ajax

- **Apps Web Tradicionales : Actualizaciones Grandes No Frecuentes**

Página Web 1.

Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.
Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.

Presione Enviar



Servidor

Respuesta
es una
nueva
página

Página Web 2.

Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.
Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.

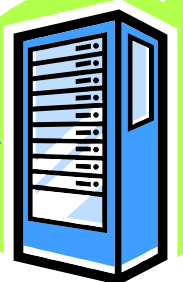
- **Apps Ajax:**
Actualizaciones Pequeñas
Frecuentes

Página Web.

Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.
Blah, blah, blah, blah. Yadda,
yadda, yadda. Blah, blah, blah, blah.
Yadda, yadda, yadda. Blah, blah,
blah, blah. Yadda, yadda, yadda.

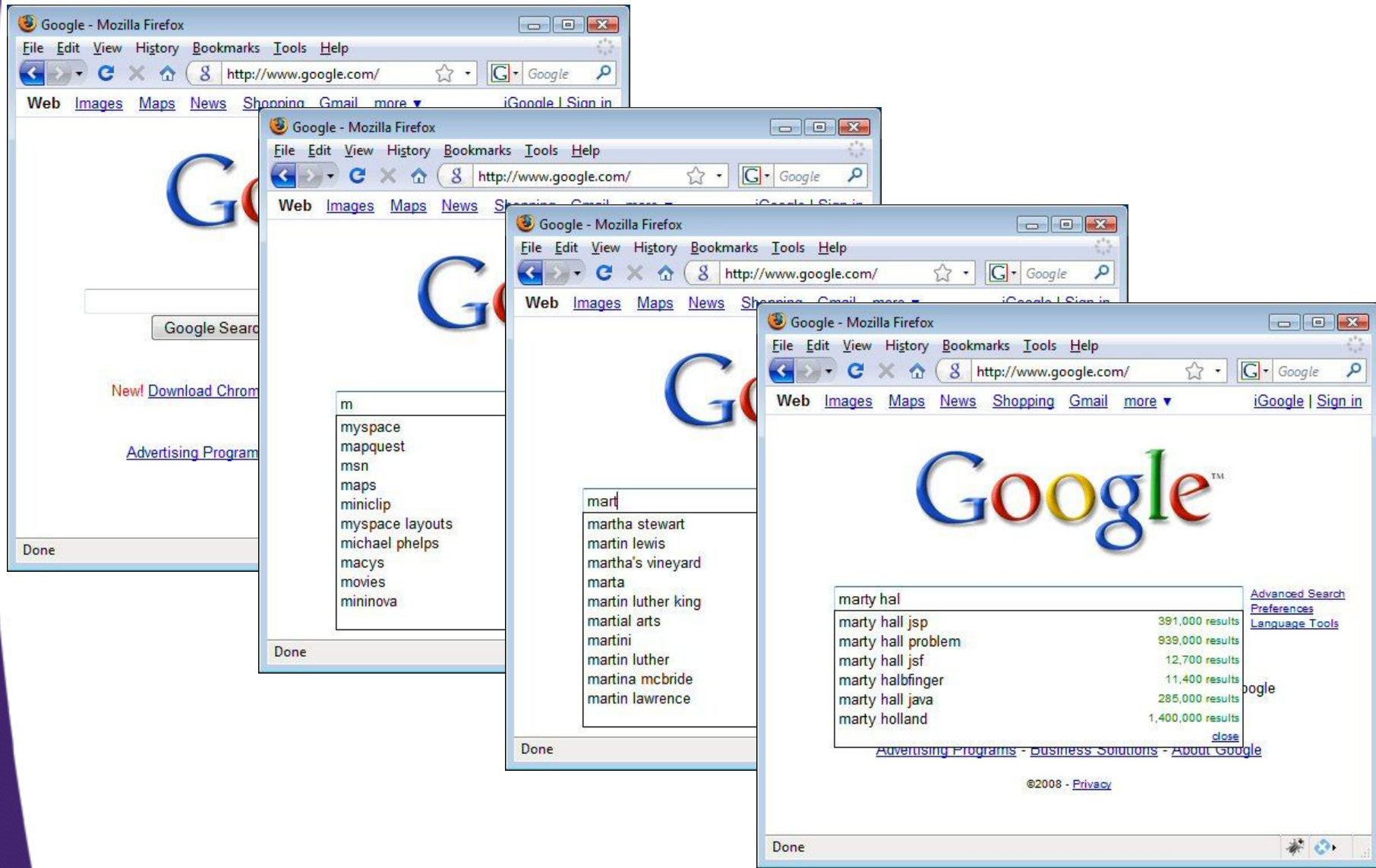
Solicitud HTTP desde JavaScript

Respuesta

**Servidor**

Resposta es una pequena parte de datos que es insertada en la pagina actual

Página de Google

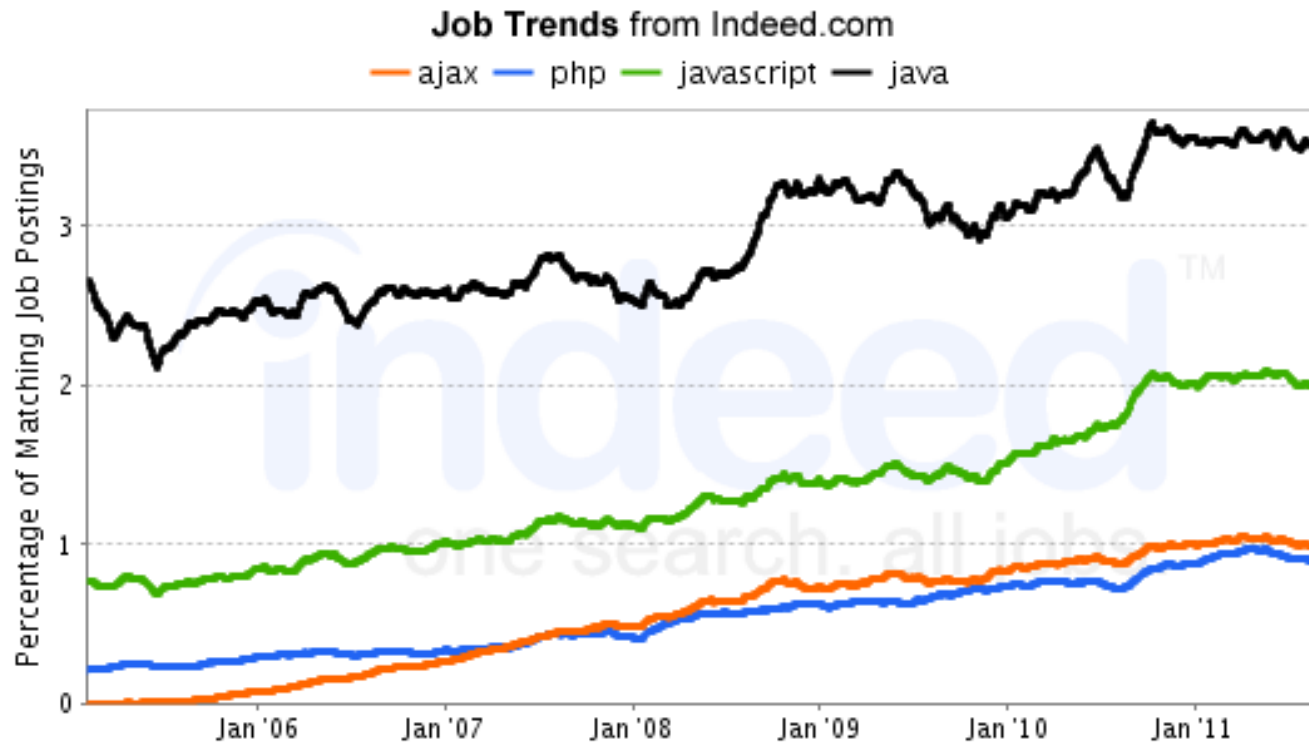


Más Ejemplos de Ajax

- <http://maps.google.com/>
- <http://demo.backbase.com/explorer/>
- <http://www.laszlosystems.com/demos/>
- http://www.smartclient.com/index.jsp#_Welcome

Trabajos en Ajax

Indeed.com compila datos de varios sitios de ofertas laborales





Proceso Básico

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Proceso Básico de Ajax

- **JavaScript**

- Defina un objeto para enviar solicitudes HTTP
- Inicie solicitud
 - Obtenga un objeto de solicitud
 - Diseñe una función anónima de manejo de respuesta
 - Proporcionado como atributo `onreadystatechange` de la solicitud
 - Inicie una solicitud GET o POST
 - Envíe datos
- Maneje respuesta
 - Espere por un valor de `readyState` de 4 y `status` HTTP de 200
 - Extraiga texto de retorno con `responseText` o `responseXML`
 - Haga algo con el resultado

- **XHTML**

- Cargue JavaScript
- Diseñe control que inicia la solicitud
- De ids a los elementos de entrada y la región de salida

Defina un Objeto de Solicitud

```
function getRequestObject() {  
    if (window.XMLHttpRequest) {  
        return(new XMLHttpRequest());  
    } else if (window.ActiveXObject) {  
        return(new ActiveXObject("Microsoft.XMLHTTP"));  
    } else {  
        return(null);  
    }  
}
```

Versión para Firefox, Netscape
5+, Opera, Safari, Mozilla,
Chrome, Internet Explorer 7 e IE
8.

Version para Internet Explorer 5.5 y 6

Falla en navegadores anteriores no estandar. No quiere hacer
un "throw new Error(...)" aqui porque esto sucede en
navegadores muy antiguos y Error apareció en JavaScript 1.5.

Inicie Solicitud

```
function sendRequest() {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { handleResponse(request) };  
    request.open("GET", "datos-mensaje.html", true);  
    request.send(null);  
}
```

Codigo a llamar cuando el servidor responda

Datos POST
(siempre null para solicitud GET)

URL del recurso en el servidor. Debe estar en el mismo servidor del cual la página fue cargada (*).

No espere por respuesta
(Envia solicitud asíncronamente)

(*) La regla de la Misma Fuente (o Mismo Origen) se ha relajado algo a partir de Firefox 3.5. Sin embargo, solo puede hacer solicitudes intersitio si el URL al que se conecta coloca un encabezado que específicamente lo permite. Para detalles, vea <http://www.petefreitag.com/item/703.cfm> y https://developer.mozilla.org/En/HTTP_access_control

Maneje Respuesta

```
function handleResponse(request) {  
    if (request.readyState == 4) {  
        alert(request.responseText);  
    }  
}
```

4 significa que la respuesta del servidor esta completa
(el manejador es invocado multiples veces –
ignore las primeras)

Texto de la respuesta del servidor

Cuadro de diálogo en popup

Funciones en JavaScript

- **JavaScript le deja pasar funciones a funciones**

```
function hazAlgoConRespuesta () { codigo }  
request.onreadystatechange = hazAlgoConRespuesta;
```

- Algo similar a los apuntadores de función en C/C++
 - Java no permite esto

- **JavaScript permite funciones anónimas**

```
var request = getRequestObject();  
request.onreadystatechange =  
    function() { codigo-que-usa-variable-request };
```

- Java tienen clases anónimas, pero no funciones anónimas
- C y C++ no tienen nada parecido a funciones anónimas.
- Las funciones anónimas son ampliamente usadas en Lisp, Ruby, Scheme, C# (as of 2.0), Python, Visual Basic, ML, PHP (a partir de versión 5.3), Clojure, Go, & otros.

Funciones: Ejemplos

```
function cuadrado(x) { return(x * x); }  
function triple(x) { return(x * 3); }  
function hazOperacion(f, x) { return(f(x)); }
```

```
hazOperacion(cuadrado, 5); → 25
```

```
hazOperacion(triple, 10); → 30
```

```
var funciones = [cuadrado, triple];
```

```
funciones[0](10); → 100
```

```
funciones[1](20); → 60
```

Funciones Anónimas: Ejemplos

```
function cuadrado(x) { return(x * x); }
```

```
cuadrado(10); → 100
```

```
(function(x) { return(x * x); })(10); → 100
```

```
function hazMultiplicador(n) {  
    return(function(x) { return(x * n); });  
}
```

```
var factor = 5;
```

```
var f = hazMultiplicador(factor);
```

```
f(3); → 15
```

```
factor = 500;
```

```
f(3); → 15
```

Enfoque Común pero Incorrecto (Variable de Solicitud Global)

```
var request;
```

```
function getRequestObject() { ... }
```

```
function sendRequest() {  
    request = getRequestObject();  
    request.onreadystatechange = handleResponse;  
    request.open("GET", "...", true);  
    request.send(null);  
}
```

```
function handleResponse() {  
    if (request.readyState == 4) {  
        alert(request.responseText);  
    }  
}
```

- Este es el enfoque mostrado en *Foundations of Ajax*, *Ajax in Practice*, *Ajax in Action*, *JavaScript the Definitive Guide*, *Pro JavaScript Techniques*, y *jQuery in Action*.

Problema con el Enfoque Común: Condiciones de Competencia!

- **Escenario**

- Dos botones XHTML, el primero llamando a funcion1 y el segundo llamando a funcion2
- funcion1 toma 5 segundos para obtener resultado del servidor
- funcion2 toma 1 segundo para obtener resultado del servidor

- **Problema**

- Suponga que se presiona el boton1, y un segundo después el boton2.
 - Cuando funcion1 busca request.responseText, obtiene el texto de la respuesta para la funcion2!
 - La función que proporciona a onreadystatechange debe tener cero argumentos, así que no se puede usar una función nombrada normal.

- **Solución**

- Use una función anónima con una copia *local* del objeto de solicitud incrustado dentro del código.

Enfoque Corregido (Variable Local de Solicitud)

```
function getRequestObject() { ... }
```

```
function sendRequest() {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { handleResponse(request); };  
    request.open("GET", "...", true);  
    request.send(null);  
}
```

```
function handleResponse(request) {  
    ...  
}
```

Código JavaScript Completo (muestra-mensaje.js)

```
function getRequestObject() {  
    if (window.XMLHttpRequest) {  
        return(new XMLHttpRequest());  
    } else if (window.ActiveXObject) {  
        return(new ActiveXObject("Microsoft.XMLHTTP"));  
    } else {  
        return(null);  
    }  
}
```

```
function sendRequest() {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { handleResponse(request); };  
    request.open("GET", "message-data.html", true);  
    request.send(null);  
}
```

```
function handleResponse(request) {  
    if (request.readyState == 4) {  
        alert(request.responseText);  
    }  
}
```

Código XHTML

- **Use XHTML, no HTML 4**

- Para poder manipular el DOM

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">...</html>
```

- **Cargue el archivo JavaScript**

```
<script src="url-relativo-de-archivo-JavaScript"
type="text/javascript"></script>
```

- Use una etiqueta de cierre </script> separada

- **Designe el control para iniciar la solicitud**

```
<input type="button" value="etiqueta del boton"
onclick="funcionPrincipal()" />
```

Defectos en XHTML de Internet Explorer

- **No puede manejar el encabezado XML**

- Los documentos XML en general deben supuestamente iniciar con un encabezado XML :

- `<?xml version="1.0" encoding="UTF-8"?>`
 `<!DOCTYPE html ...>`
 `<html xmlns="http://www.w3.org/1999/xhtml">...</html>`

Omita esto!

- La especificación de XHTML recomienda usarlo
- *Pero...* Internet Explorer cambiará a un modo especial si DOCTYPE no es la primer línea.
 - Muchos formatos de hojas de estilo reciente serán ignorados
 - **Por tanto, omita el encabezado XML**

- **Requiere etiquetas de finalización separadas en algunos lugares**

- Los script no cargarán si usa `<script .../>`
en vez de `<script...></script>`

No haga esto.

Mejor haga esto.

Código XHTML (muestra-mensaje.html)

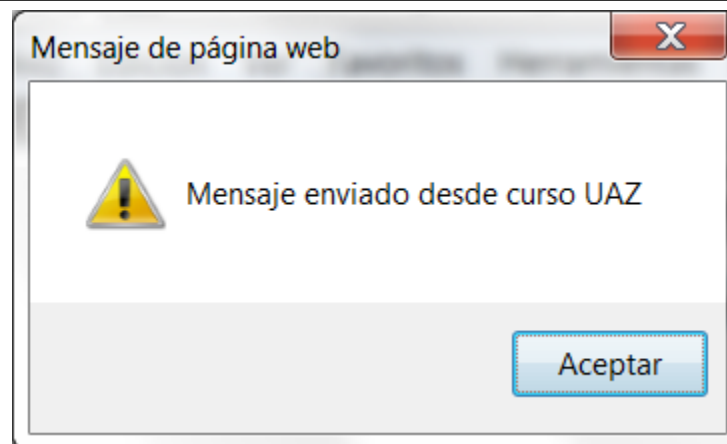
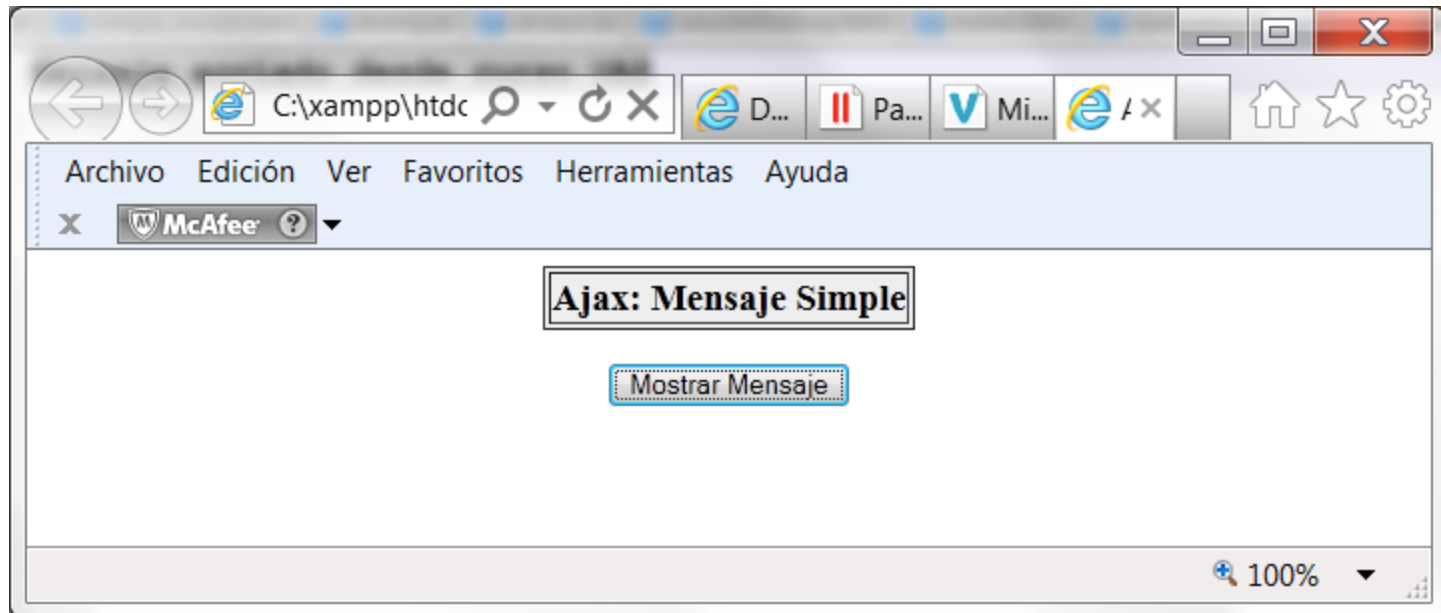
```
<!DOCTYPE html PUBLIC "..."  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head><title>Ajax: Mensaje Simple</title>  
<script src="muestra-mensaje.js"  
    type="text/javascript"></script>  
</head>  
<body>  
<center>  
<table border="1" bgcolor="gray">  
    <tr><th><big>Ajax: Mensaje Simple</big></th></tr>  
</table>  
<p/>  
<input type="button" value="Mostrar Mensaje"  
    onclick="sendRequest()" />  
</center></body></html>
```

Código XHTML (datos-mensaje.html)

Mensaje enviado desde curso UAZ

- **Nota: ejecución de este ejemplo**
 - Ya que la página principal usa URLs relativos y el código XHTML aquí no tiene contenido dinámico, puede ejecutar este ejemplo directamente del disco sin usar un servidor. Posteriores ejemplos requerirán contenido dinámico y deberán por lo tanto usar un servidor.

Proceso Básico: Resultados



Pruebas de Ajax

- **JavaScript es notoriamente inconsistente**
 - Es lógico esperar que librerías tales como Prototype, jQuery, etc. tomen esto en cuenta y oculten las diferencias en los navegadores. Aun así, debería hacer pruebas.
- **Pruebas en multiples navegadores**
 - Si está haciendo una aplicación interna, pruebela en todos los navegadores oficiales aprobados y en todos los sistemas operativos soportados.
 - Si está haciendo una aplicación externa, pruebe en la mayor cantidad de navegadores posible. Preferiblemente: IE 6, IE 7, IE 8, una implementación reciente de Firefox y Chrome. Probar en Safari y Opera no está mal, pero son menos usados.
 - Pruebe regularmente en Chrome y Firefox. Pruebe en IE y un mayor conjunto de navegadores antes de liberar su aplicación.
 - Mercado de navegadores:
http://www.w3schools.com/browsers/browsers_stats.asp
 - Estadísticas actuales(Mayo 2012)
 - IE: 18.1%, Firefox: 35.2%, Chrome: 39.3%, Safari: 4.3%, Opera: 2.2%



Contenido Dinámico desde PHP

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Primer Ejemplo: Deficiencias de Diseño

- **El contenido fue el mismo en cada solicitud**
 - Pudieramos haber puesto el valor de la alerta directamente en el código JavaScript
 - En vez de ello, invoquemos una página PHP en el servidor
- **Dirección de recurso fija en código de**
 - Evita que las funciones se apliquen a varias situaciones
 - En vez de ello, haga una función genérica y pásese la dirección
- **El archivo JavaScript estaba en la misma carpeta que el código XHTML**
 - Dificulta el reuso de JavaScript en diferentes paginas
 - En vez de ello, haga un directorio especial para JavaScript
- **No se uso hoja de estilos**
 - Menos cosas con las que JavaScript puede trabajar cuando esta manipulando la página
 - Use CSS por las razones que se indicaron en el caso de JavaScript

Pasos

- **JavaScript**

- Defina un objeto para enviar solicitudes HTTP
- Inicie solicitud
 - Obtenga un objeto de solicitud
 - Diseñe una función anónima de manejo de respuesta
 - Proporcionado como atributo onreadystatechange de la solicitud
 - Inicie una solicitud GET o POST a una página PHP
 - Obtenga la dirección de una variable en vez de ponerla fija en código
 - Envíe datos
- Maneje respuesta
 - Espere por un valor de readyState de 4 y status HTTP de 200
 - Extraiga texto de retorno con responseText o responseXML
 - Haga algo con el resultado

- **XHTML**

- Cargue JavaScript de un directorio centralizado. Use hoja de estilo.
- Diseñe control que inicia la solicitud
- De ids a la región de salida

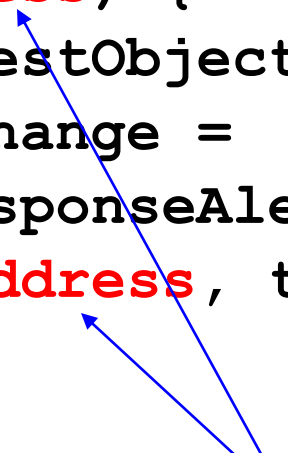
Defina un Objeto de Solicitud

```
function getRequestObject() {  
    if (window.XMLHttpRequest) {  
        return(new XMLHttpRequest());  
    } else if (window.ActiveXObject) {  
        return(new ActiveXObject("Microsoft.XMLHTTP"));  
    } else {  
        return(null);  
    }  
}
```

Sin cambios con respecto al ejemplo previo.
Este código permanece igual para toda la sección.

Inicie Solicitud

```
function ajaxAlert(address) {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { showResponseAlert(request); };  
    request.open("GET", address, true);  
    request.send(null);  
}
```



URL relativa del recurso del lado del servidor.
(En este ejemplo, le pasaremos la dirección de una página PHP.)

Maneje Respuesta

```
function showResponseAlert(request) {  
    if ((request.readyState == 4) &&  
        (request.status == 200)) {  
        alert(request.responseText);  
    }  
}
```



La respuesta del servidor regresó sin errores
(Código de status HTTP 200).

Código JavaScript Completo (Parte de ajax-utils.js)

```
function getRequestObject() {  
    if (window.XMLHttpRequest) {  
        return(new XMLHttpRequest());  
    } else if (window.ActiveXObject) {  
        return(new ActiveXObject("Microsoft.XMLHTTP"));  
    } else {  
        return(null);  
    }  
}
```

```
function ajaxAlert(address) {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { showResponseAlert(request); }  
    request.open("GET", address, true);  
    request.send(null);  
}
```

```
function showResponseAlert(request) {  
    if ((request.readyState == 4) &&  
        (request.status == 200)) {  
        alert(request.responseText);  
    }  
}
```

Código XHTML

- **Cargue JavaScript de una ubicación central**

```
<script src="scripts/ajax-utils.js"  
        type="text/javascript"></script>
```

- **Passe la dirección PHP a la función principal**

```
<input type="button" value="Mostrar Hora del Servidor"  
        onclick='ajaxAlert("muestra-hora.php")' />
```

- **Use hoja de estilo**

```
<link rel="stylesheet"  
        href="css/styles.css"  
        type="text/css"/>
```

Note los apóstrofes (debido a las comillas dentro de los paréntesis).

Código XHTML

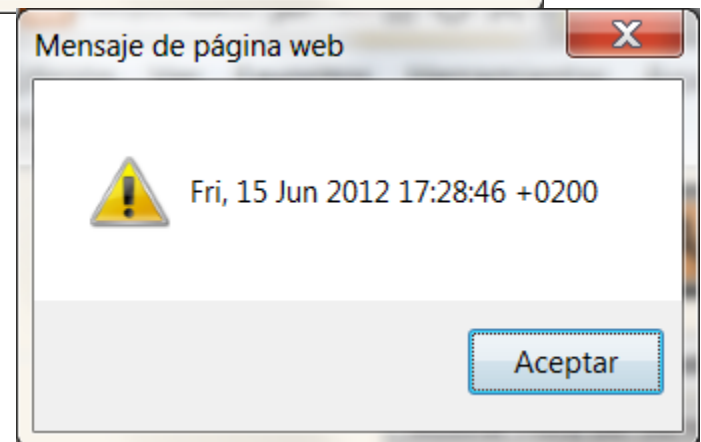
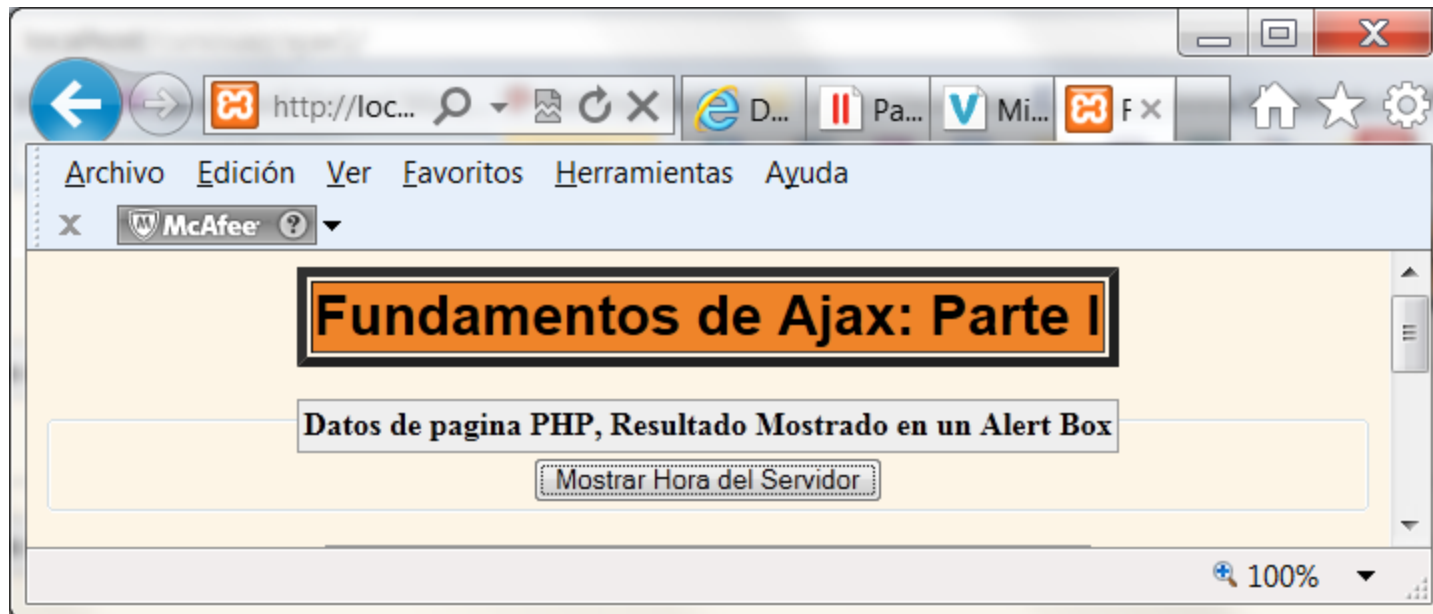
```
<!DOCTYPE html PUBLIC "...">
<html xmlns="http://www.w3.org/1999/xhtml">...
<link rel="stylesheet"
      href="css/styles.css"
      type="text/css"/>
<script src="scripts/ajax-utils.js"
        type="text/javascript"></script>...
<body>...
<fieldset>
  <legend> Datos de pagina PHP, Resultado Mostrado en un
  Alert Box </legend>
  <form action="#">
    <input type="button" value="Mostrar Hora del Servidor"
          onclick='ajaxAlert("muestra-hora.php")' />
  </form>
</fieldset>
...
```

Codigo PHP(muestra-hora.php)

```
<?php print date("r"); ?>
```

- **Nota: ejecución de este ejemplo**
 - Debe ejecutar desde el servidor, no directamente del disco
 - De otra manera, PHP no se ejecuta
 - Y el código de status es -1, no 200

Mensaje desde PHP: Resultados





Desplegado de Resultados en XHTML

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Ejemplo con PHP: Deficiencias de Diseño

- **Resultados siempre se muestran en una cuadro de diálogo (alert)**
 - Las alertas usualmente reservadas para errores o advertencias
 - Los usuarios prefieren resultados normales dentro de la página
 - Solución: use XHTML dinámico para actualizar la página con el resultado
 - XHTML mas estilos CSS representados en el DOM
 - DOM es por "Document Object Model", una vista XML de la página
 - » Note que Firebug tiene un asombroso explorador de DOM explorer.
 - JavaScript puede insertar elementos en el DOM
 - Encuentre un elemento con un id dado
 - » `algunElemento = document.getElementById(id);`
 - Inserte código XHTML dentro
 - » `algunElemento.innerHTML = "<h1>blah</h1>";`
 - JavaScript puede también leer el DOM
 - Por ejemplo, buscar valores de campos de textotextfield
 - » `document.getElementById(id).value`

Inserción Dinámica de Texto

- **XHTML**

- `<div id="region-resultados"></div>`

- **JavaScript**

- `resultRegion =`

- `document.getElementById("region-resultados");`

- `resultRegion.innerHTML = "<h2>Orale!</h2>";`

- Para el texto a poner en innerHTML, usualmente usa `request.responseText` o algun string derivador de `request.responseText`

- **Resultado después de ejecutar el código**

- `<div id="region-resultados"><h2>Orale!</h2></div>`

- “Ver código fuente” no mostrará esto, pero Firebug si.

- **Advertencia**

- Asegúrese de insertar código XHTML legal

Resumen de Nuevas Características

- **XHTML**

- Defina un elemento div inicialmente en blanco

```
<div id="textoResultado"></div>
```

- **Manejador de respuesta JavaScript**

- Proporciona un id (resultRegion), encuentre el elemento con ese id, e inserte el texto de respuesta en la propiedad innerHTML

```
document.getElementById(resultRegion).innerHTML =  
    request.responseText;
```


Pasos

- **JavaScript**

- Defina un objeto para enviar solicitudes HTTP
- Inicie solicitud
 - Obtenga un objeto de solicitud
 - Diseñe una función anónima de manejo de respuesta
 - Proporcionado como atributo onreadystatechange de la solicitud
 - Inicie una solicitud GET o POST a una página PHP
 - Envíe datos
- Maneje respuesta
 - Espere por un valor de readyState de 4 y status HTTP de 200
 - Extraiga texto de retorno con responseText o responseXML
 - Use innerHTML para insertar resultado en elemento designado

- **XHTML**

- Cargue JavaScript de un directorio centralizado. Use hoja de estilo.
- Diseñe control que inicia la solicitud
- De ids a la región de salida

Defina un Objeto de Solicitud

```
function getRequestObject() {  
    if (window.XMLHttpRequest) {  
        return(new XMLHttpRequest());  
    } else if (window.ActiveXObject) {  
        return(new ActiveXObject("Microsoft.XMLHTTP"));  
    } else {  
        return(null);  
    }  
}
```

[Sin cambios de los ejemplos anteriores](#)

Inicie Solicitud

```
function ajaxResult(address, resultRegion) {  
    var request = getRequestObject();  
    request.onreadystatechange =  
        function() { showResponseText(request,  
                                         resultRegion); };  
    request.open("GET", address, true);  
    request.send(null);  
}
```

Maneje Respuesta

```
function showResponseText(request, resultRegion) {  
    if ((request.readyState == 4) &&  
        (request.status == 200)) {  
        htmlInsert(resultRegion, request.responseText);  
    }  
}
```

```
function htmlInsert(id, htmlData) {  
    document.getElementById(id).innerHTML = htmlData;  
}
```

Código XHTML

```
...
<link rel="stylesheet"
      href="css/styles.css"
      type="text/css"/>
<script src="scripts/ajax-utils.js"
        type="text/javascript"></script>
...
<fieldset>
  <legend>Datos de pagina PHP, Resultado Mostrado en
  XHTML</legend>
  <input type="button" value="Mostrar Hora del Servidor"
        onclick='ajaxResult("muestra-hora.php", "resultadoHora")' />
  <div id="resultadoHora" class="ajaxResult"></div>
</fieldset>
...
```

Código de Hoja de Estilo (css/styles.css)

```
.ajaxResult { color: #440000;  
              font-weight: bold;  
              font-size: 18px;  
              font-family: Arial, Helvetica, sans-serif;  
}
```

Código PHP

- **Sin cambios con respecto al código anterior**
 - Regresa un string formateado con fecha y hora



Regla del Mismo Origen (Misma Fuente)

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

JavaScript Sólo se puede Conectar al mismo Host que la Página XHTML Principal

- **Motivación**

- La razón más importante es preservar seguridad en el firewall.
 - JavaScript está ejecutandose en *su* computadora, así que tiene los mismos privilegios que *usted*. Si pudiera acceder sitios Web que “públicos” para usted, podría acceder sitios propietarios dentro de su firewall a los que usuarios externos están bloqueados. Podría entonces reportar esta información al exterior.
- Los applets de Java tienen la misma regla, por la misma razón
- Consecuencia: use URLs relativos para direcciones de servidor

- **Formas de evitar esta regla**

- Hacerlo en dos pasos.
 - JavaScript se conecta a su programa PHP, que a su vez obtiene datos de otro sitio
- Sóloo en Firefox : haga que el sitio web envíe un encabezado especial
 - Access-Control-Allow-Origin: *

Resumen

- **JavaScript**

- Defina un objeto de solicitud
 - Verifique tanto objetos Microsoft y no MS.
- Inicie la solicitud
 - Obtenga el objeto de solicitud
 - Diseñe una función anónima de manejo de respuesta
 - Inicie una solicitud GET con un URL relativo
- Maneje la respuesta
 - Espere a un valor de readyState de 4 y un status HTTP de 200
 - Extraiga el texto de retorno con responseText
 - Haga algo con el resultado
 - Use innerHTML para insertar resultado en elemento designado

- **XHTML**

- De id a la región donde colocará el resultado (a menudo una sección div). Inicie el proceso cuando el usuario haga alguna acción.

- **PHP**

- Genere el contenido dinámico.



Preguntas?

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, JSF 2.0, Java 6, Ajax, jQuery, GWT, Spring, Hibernate, RESTful Web Services, Android.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.