



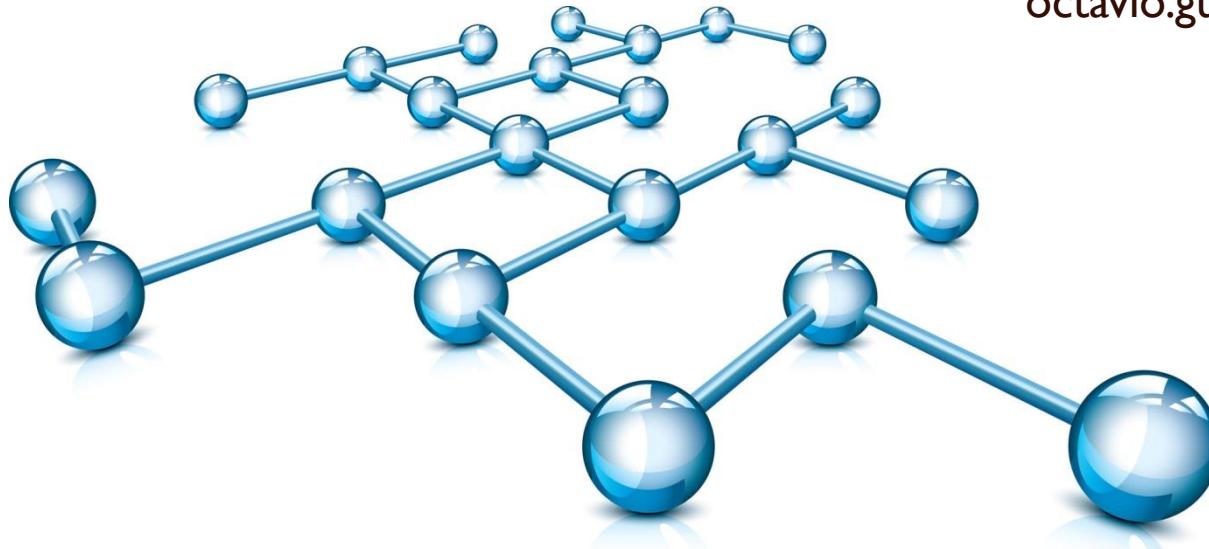
AJAX

Asynchronous Javascript And XML

Profesor:

Dr. J. Octavio Gutiérrez García

octavio.gutierrez@itam.mx





$\approx \neq$



- Operadores
- Estructuras de control
 - Condiciones
 - Ciclos

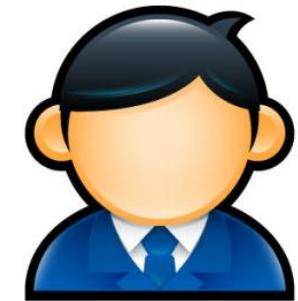




JavaScript

Lenguaje script de la WEB,
interpretado en tiempo de ejecución
(en lugar de ser compilado).

Utilizado del lado del cliente



Se puede insertar en páginas HTML.

Compatible con múltiples navegadores



Haciendo referencia a elementos de HTML

DOM (Document Object Model)

- Estándar oficial de W3C para acceder a elementos HTML

```
document.getElementById("someId")
```



Introduciendo JavaScript en documentos HTML

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <script>
      document.write("<p>This is my first line using JavaScript</p>");
    </script>
  </body>
</html>
```

Se pueden insertar scripts en el `<body>` o `<head>`



JavascriptMiscellaneous



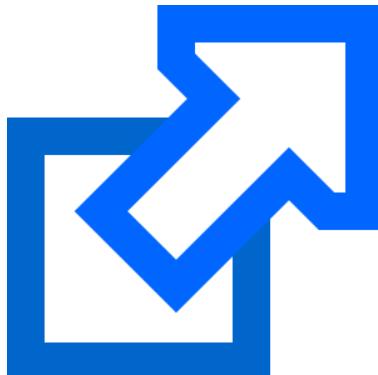
JAVASCRIPT

Archivos externos de JavaScript

```
<html>
<body>

    <script src="myjavascriptfile1.js"></script>
    <script src="myjavascriptfile2.js" defer></script>
    <script src="myjavascriptfile3.js" async></script>

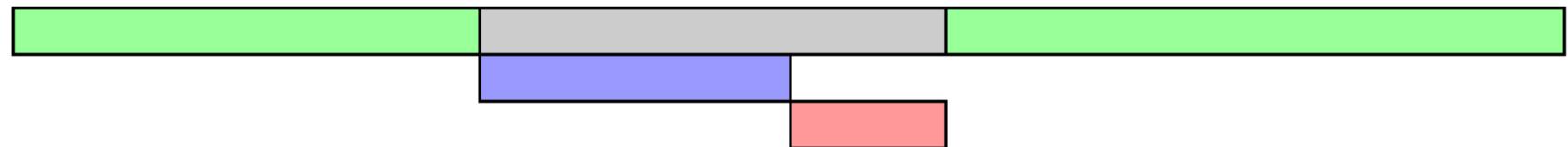
</body>
</html>
```



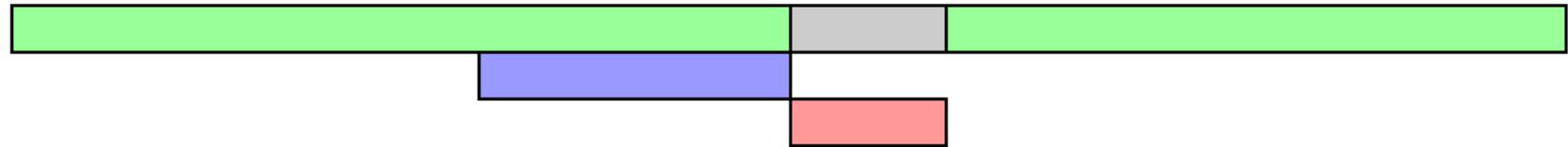
Orden de Ejecución de Scripts

- Descarga y parseo de HTML
- Descarga y parseo de HTML detenidos
- Descarga del script
- Ejecución del script

`<script>`



`<script async>`



`<script defer>`



Variables



- Las variables **locales** se declaran con “**var**”

Para declarar una variable global, se omite “**var**”

```
var pi=3.14;  
var person="Juan Perez";
```

- Tipos dinámicos:** la misma variable puede almacenar diferentes tipos de datos

```
var x;           // Ahora x no está definida  
x = 5;          // Ahora x es un número  
x = "Juan";     // Ahora x es una String
```

- Arrays**

```
var cars=new Array();  
cars[0]="Saab";  
cars[1]="Volvo";  
cars[2]="BMW";
```

```
var cars=new Array("Saab","Volvo","BMW");
```

```
var cars=["Saab","Volvo","BMW"];
```

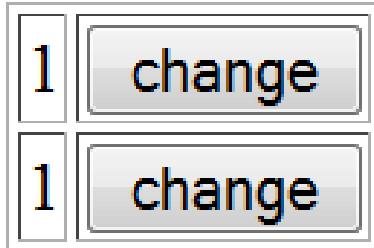
Funciones



```
<script>
```

```
function changeText(elementId) {  
    var element=document.getElementById(elementId);  
    element.innerHTML="12345678910";  
}
```

```
</script>
```



Normalmente las funciones se agregan en el head o al final del documento

```
function myFunction() {  
    var x=5;  
    return x;  
}
```

```
<input type="button" value="change" onclick="changeText('one');"/>
```

Reaccionando a eventos

```
<script>
```

```
    function changelImage() {  
        element=document.getElementById('myimage')  
        if (element.src.match("down")) {  
            element.src="IMAGES/up.jpg";  
        } else {  
            element.src="IMAGES/down.jpg";  
        }  
    }  
</script>
```

Referencia a elemento
Funciones en los eventos de los componentes

```

```



JSON: Objetos en JavaScript



```
<p>
```

```
    Name: <span id="aName"> </span> <br/>
```

```
    Year: <span id="aYear"> </span> <br/>
```

```
</p>
```

```
<script>
```

```
var anObject= {  
    name:"Smith",  
    year:1984};
```

```
anObject.name;  
anObject["name"];
```

```
document.getElementById("aName").innerHTML=anObject.name;  
document.getElementById("aYear").innerHTML=anObject.year;
```

```
var person=new Object();  
person.firstname="Juan";  
person.age=50;  
document.write(person.firstname + " is " + person.age + " years old.");  
</script>
```

JSON: Objetos en JavaScript

```
<script>
    var person=new Object();
    person.firstname="Juan";
    person.age=50;
    person.aMethod= function(x){
        person.age = person.age + x;
    }
    person.aMethod(3);
    document.write(person.firstname + " is " + person.age + " years old.");
</script>
```



Excepciones en JavaScript

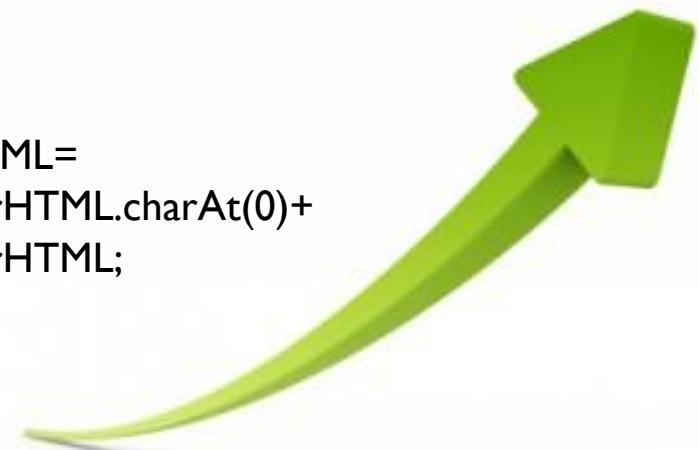
```
<script>
function myFunction() {
    var y=document.getElementById("mess");
    y.innerHTML="";
    try {
        var x=document.getElementById("demo").value;
        if(x=="")      throw "empty";
        if(isNaN(x))  throw "not a number";
        if(x>10)      throw "too high";
        if(x<5)       throw "too low";
    } catch(err) {
        y.innerHTML="Error: " + err + ".";
    }
}
</script>
```



```
<p>Please input a number between 5 and 10:</p>
<input id="demo" type="text">
<button type="button" onclick="myFunction()">Test Input</button>
<p id="mess"></p>
```

Añadiendo eventos: Ejemplo I

```
<script>
    function increaseText(id){
        document.getElementById(id).innerHTML=
            document.getElementById(id).innerHTML.charAt(0) +
            document.getElementById(id).innerHTML;
    }
</script>
```

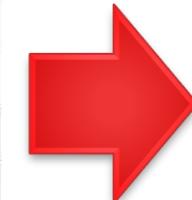


```
<h1 id ="anyId1" onclick="increaseText('anyId1')">Click on this text!</h1>
<h1 id ="anyId2" onclick="increaseText('anyId2')">Click on this text!</h1>
```

Añadiendo eventos: Ejemplo 2

```
<script>  
    function upperCase(id){  
        var text=document.getElementById(id);  
        text.value=text.value.toUpperCase();  
    }  
</script>
```

a	b	c
d	e	f
g	h	i
j	k	l
m	n	o
p	q	r
s	t	u
v	w	x
y	z	



A	B	C
D	E	F
G	H	I
J	K	L
M	N	O
P	Q	R
S	T	U
V	W	X
Y	Z	

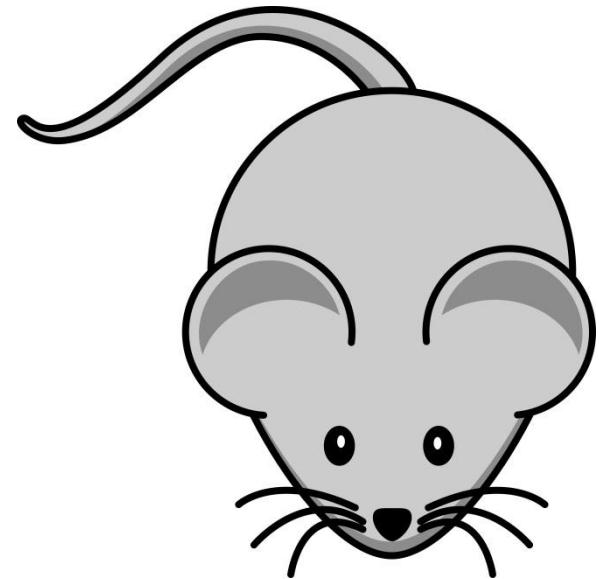
```
<input type="text" id="anyId3" onchange="upperCase('anyId3');">
```

Añadiendo eventos: Ejemplo 3

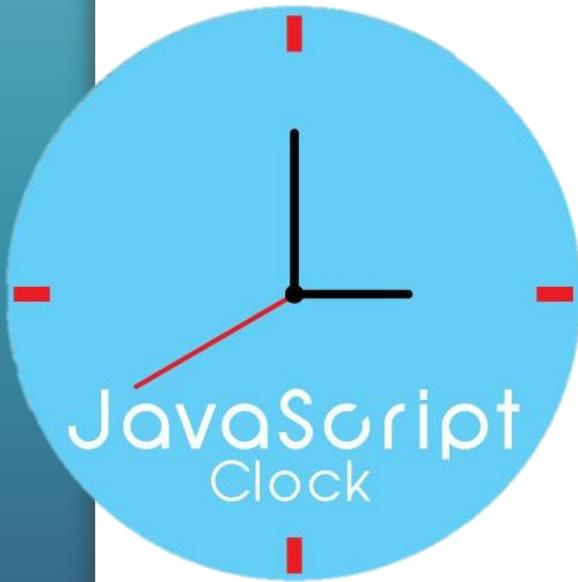
```
<script>  
function eventMouseOver(obj) {  
    obj.innerHTML="Thank You";  
}  
</script>
```

```
function eventMouseOut(obj){  
    obj.innerHTML="Mouse Over Me";  
}  
</script>
```

```
<div onmouseover="eventMouseOver(this)"  
     onmouseout="eventMouseOut(this)">  
    Mouse Over Me  
</div>
```



Temporización



- **setTimeout(función, tiempo)**
 - Dispara la función una sola vez
 - `clearTimeout()` borra el evento
- **setInterval(función, tiempo)**
 - Dispara la función repetidamente
 - `clearInterval()` borra el evento

Validación de Formularios

```
<script>
function validateForm(){
    var x=document.forms["myForm"]["fname"].value;
    if (x==null || x=="") {
        alert("First name must be filled out");
        return false;
    }
    return true;
}
</script>
```

```
<form name="myForm"
      action="nextpage.jsp"
      onsubmit="return validateForm()"
      method="post">
```

First name: <input type="text" name="fname">


```
          <input type="submit" value="Submit">
</form>
```



Eventos de la “ventana”

- onafterprint
- onbeforeprint
- onbeforeunload
- onerror
- onhaschange
- onload
- onmessage
- onoffline
- ononline
- onpagehide
- onpageshow
- onpopstate
- onredo
- onresize
- onstorage
- onundo
- onunload



Eventos de “Form”

- onblur
- onchange
- oncontextmenu
- onfocus
- onformchange
- onforminput
- oninput
- oninvalid
- onreset
- onselect
- onsubmit



Eventos del teclado

- onkeydown
- onkeypress
- onkeyup



Eventos del mouse



- onclick
- ondblclick
- ondrag
- ondragend
- ondragenter
- ondragleave
- ondragover
- ondragstart
- ondrop
- onmousedown
- onmousemove
- onmouseout
- onmouseover
- onmouseup
- onmousewheel
- onscroll

Browser Object Model - BOM

- Objeto **windows**. Ejemplos:
 - window.innerHeight
 - window.moveTo()
 - window.resizeTo()
- Objeto **screen**. Ejemplos:
 - screen.pixelDepth
- Objeto **history**. Ejemplos:
 - history.back()
 - history.forward()

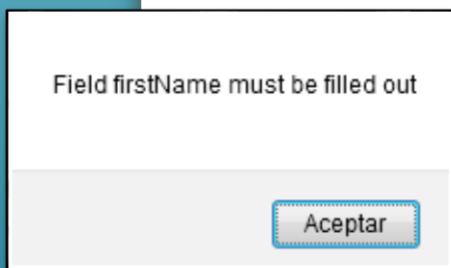


Browser Object Model - BOM

- Objeto **navigator**. Ejemplos:
 - navigator.cookieEnabled
 - navigator.systemLanguage
- **Mensajes**. Ejemplos:
 - alert
 - prompt
 - confirm



Práctica de Laboratorio JavaScript: Validación de formularios



My first bulletproof form!

Description	Value
First name:	<input type="text"/>
Last name:	<input type="text"/>
E-mail:	<input type="text"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female
Country region	<input type="text"/>
all the vaccines?	<input type="checkbox"/>
Additional comments	<input type="text"/>

Requerido

Mayúsculas
y sin
números
on-the-fly

E-mail válido

Práctica de Laboratorio JavaScript: Validación de formularios



Tips:

RadioGroup

```
var x=document.forms[form][id];
x[0].checked
x[1].checked
```

Texto requerido

```
var x=document.forms[form][id].value;
if (x==null || x=="") {}
```

alert("msg")

Field firstName must be filled out

Aceptar

Mayúsculas y números

```
x.toUpperCase();
x.charAt(i)
isNaN("a") // isNotANumber
x.substr(0, i)
x.concat("String")
x.length
```

onkeydown
onkeyup

Email

```
x.indexOf("@")
x.lastIndexOf(".")
```



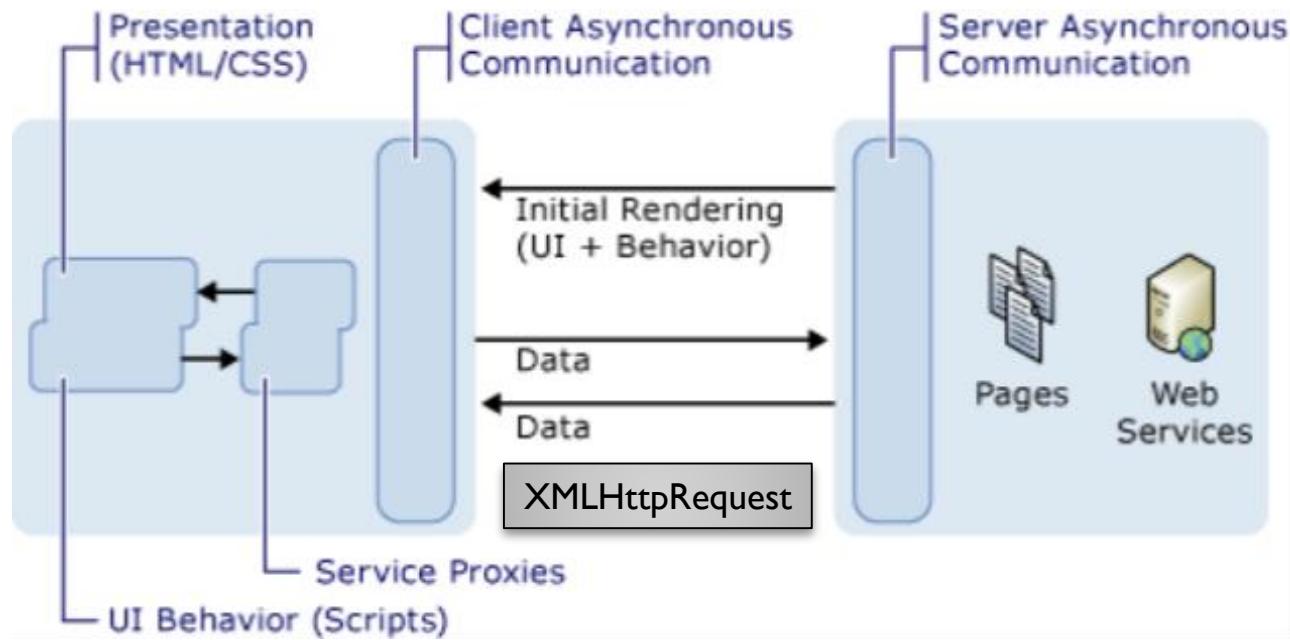
WebApplicationValidatedFormBefore
WebApplicationValidatedFormAfter

AJAX

Asynchronous Javascript And XML



AJAX es el arte de intercambiar datos con el servidor y actualizar sólo partes de una página web, sin tener que recargar toda la página.



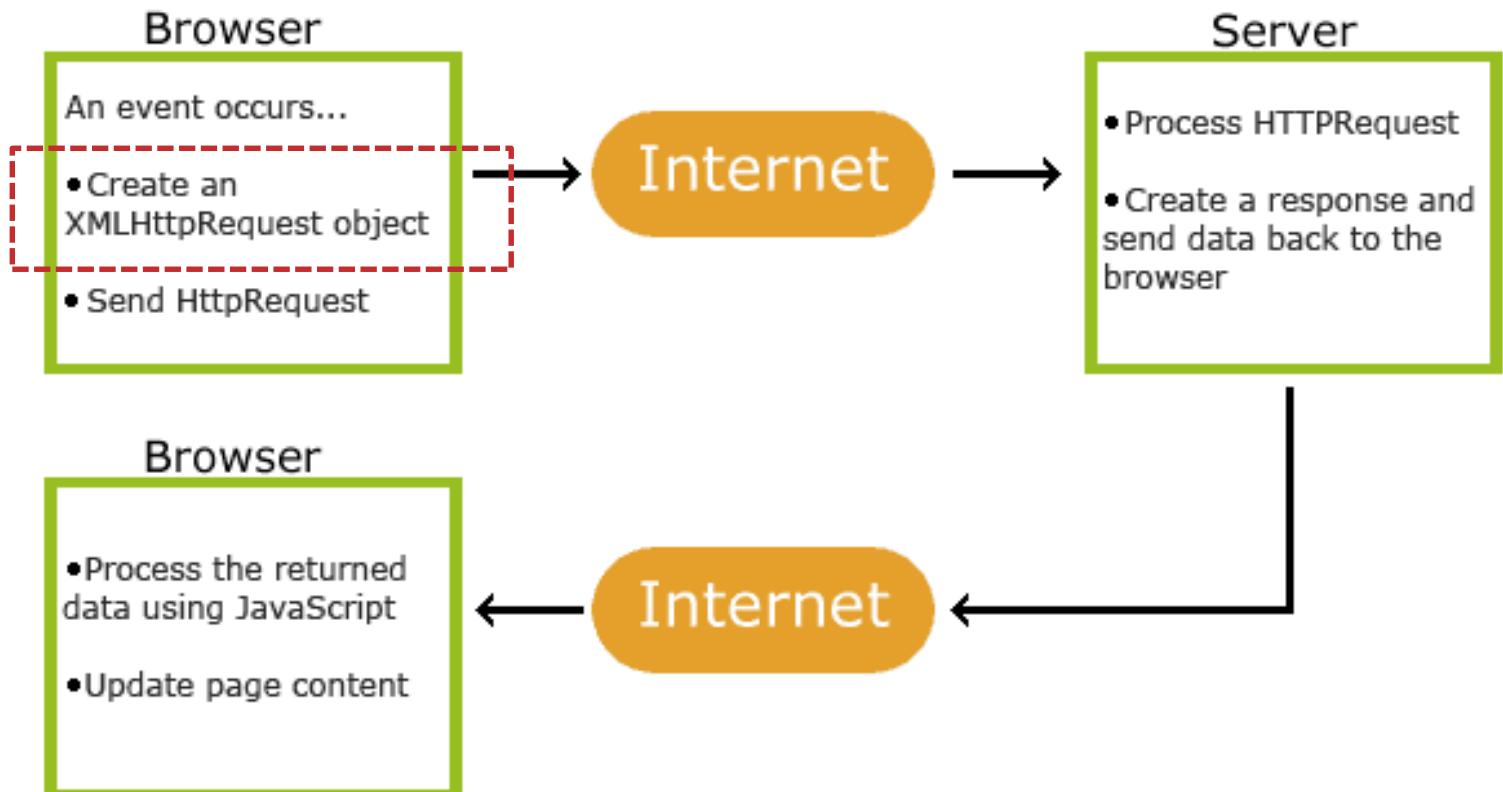
Orientados a datos y no a páginas.



Asynchronous Javascript And XML



XMLHttpRequest



Creando un objeto XMLHttpRequest

Browser Object Model - BOM

```
var ajaxRequest:  
if (window.XMLHttpRequest) {  
    // IE7+, Firefox, Chrome, Opera, Safari  
    ajaxRequest=new XMLHttpRequest();  
} else {  
    // IE6, IE5  
    ajaxRequest=new ActiveXObject("Microsoft.XMLHTTP");  
}
```

Enviando una solicitud al Servidor

Tipo de Solicitud
Get | Post

URL de
recurso

```
ajaxRequest.open ("GET", target, true /*async*/);  
  
ajaxRequest.send();
```

Solicitud
asíncrona

Envía la solicitud.
En solicitudes POST,
send puede llevar un
parámetro String

Async=true

Con Callback
Function

El evento se activa
cuando se encuentra
lista la respuesta del
servidor

200: OK
404: Page not found

```
ajaxRequest.onreadystatechange = function() {
    if (ajaxRequest.readyState==4 && ajaxRequest.status==200) {
        document.getElementById(id).innerHTML=ajaxRequest.responseText;
    }
}
```

- 0: solicitud sin inicializar
- 1: conexión con el servidor establecida
- 2: respuesta a solicitud recibida
- 3: procesando la respuesta
- 4: respuesta lista

Respuesta en texto
responseText

Respuesta en XML
responseXML

Async=false

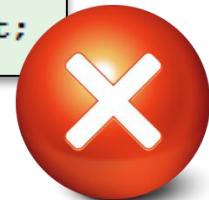
Sin Callback
Function



```
ajaxRequest.open("GET", target, false /*async*/);

ajaxRequest.send();

document.getElementById(id).innerHTML=ajaxRequest.responseText;
```





Hola mundo

```
function loadNewContent(id, target) {  
    var ajaxRequest;  
    if (window.XMLHttpRequest){  
        ajaxRequest=new XMLHttpRequest(); // IE7+, Firefox, Chrome, Opera, Safari  
    } else {  
        ajaxRequest=new ActiveXObject("Microsoft.XMLHTTP"); // IE6, IE5  
    }  
  
    ajaxRequest.onreadystatechange = function(){  
        if (ajaxRequest.readyState==4 && ajaxRequest.status==200){  
            document.getElementById(id).innerHTML=ajaxRequest.responseText;  
        }  
    }  
  
    ajaxRequest.open("GET", target, true /*async*/);  
    ajaxRequest.send();  
}
```

id: nombre del componente a modificar.
Normalmente definido con etiquetas
 o <div id="x">

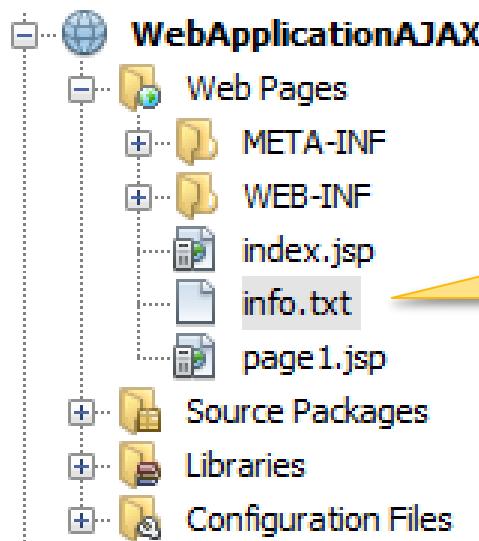
Target: URL del recurso del servidor



Hola mundo

AJAX

Asynchronous Javascript And XML



Posibles recursos:

- 1) info.txt : archivo de texto plano
- 2) page1.jsp : página JSP

Hola mundo

AJAX

Asynchronous Javascript And XML

Hello World
from  index.jsp!

Description	Value
Cell A	10.0
Cell B	20.0

Change Content



Hello World
from
index.jsp!

AJAX is not a new
programming language.

AJAX is a technique for
creating fast and dynamic
web pages.

Change Content

La tabla se definió dentro de
`<div id="myDiv"> </div>`

¿Get o Post?

- **GET** es más simple y más rápido.
- **POST** es mejor cuando...
 - un archivo de **caché no** es **opción**
 - se envía una **gran cantidad** de datos
 - los datos **no** sean **texto plano**
 - se desee mayor **seguridad**.



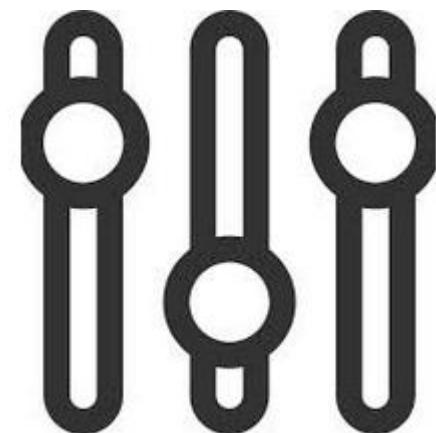
Preferred



XMLHttpRequest - Get

```
ajaxRequest.open( "GET",
    "page1.jsp?name=Octavio&age=49", true);
ajaxRequest.send();
```

Parámetros



XMLHttpRequest - Post

```
ajaxRequest.open("POST", "page1.jsp", true);
```

Nombre del encabezado

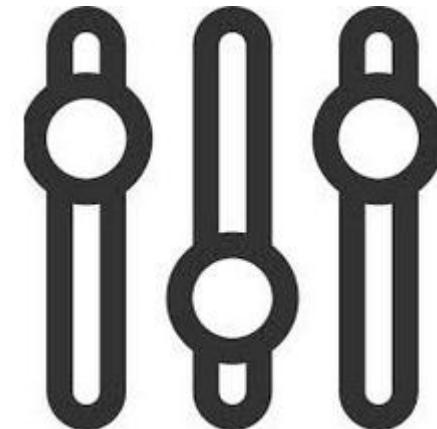
```
ajaxRequest.setRequestHeader("Content-type",  
                            "application/x-www-form-urlencoded");
```

Añade
encabezado
a la
solicitud

```
ajaxRequest.send("name=Octavio&age=49");
```

Valor del encabezado

Parámetros



XMLHttpRequest.responseXML



```
ajaxRequest.onreadystatechange = function(){
    if (ajaxRequest.readyState==4 && ajaxRequest.status==200){
        var xmlDoc=ajaxRequest.responseXML;
        var txt="";
        var x=xmlDoc.getElementsByTagName("ARTIST");
        for (var i=0;i<x.length;i++) {
            txt=txt + x[i].childNodes[0].nodeValue + "<br>";
        }
        document.getElementById("myDiv").innerHTML=txt;
    }
}
```

```
<CATALOG>
    <CD>
        <TITLE>Empire Burlesque</TITLE>
        <ARTIST>Bob Dylan</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    •
    •
    •
```



XMLHttpRequest.responseXML

```
alert(xmlDoc.getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
alert(xmlDoc.getElementsByTagName("ARTIST")[1].childNodes[0].nodeValue);
alert(xmlDoc.getElementsByTagName("ARTIST")[2].childNodes[0].nodeValue);
```



```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
  • • •
```

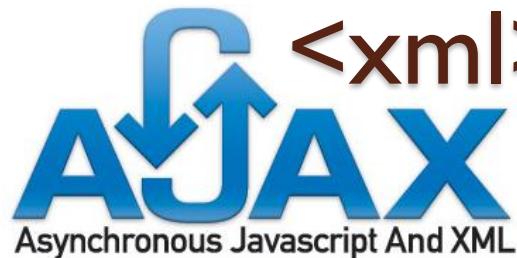
XMLHttpRequest.responseXML

```
alert(xmlDoc.getElementsByTagName("CD")[0].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
alert(xmlDoc.getElementsByTagName("CD")[1].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
alert(xmlDoc.getElementsByTagName("CD")[2].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
```



```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
  ...

```



<xml> Hola mundo </xml>

Welcome to my old CD Store!

[Get CD Catalog](#)



WebApplicationResponseXML

Welcome to my old CD Store!

Bob Dylan
Bonnie Tyler
Dolly Parton
Gary Moore
Eros Ramazzotti
Bee Gees
Dr.Hook
Rod Stewart
Andrea Bocelli
Percy Sledge
Savage Rose
Many
Kenny Rogers
Will Smith
Van Morrison
Jorn Hoel
Cat Stevens
Sam Brown
T'Pau
Tina Turner
Kim Larsen
Luciano Pavarotti
Otis Redding
Simply Red
The Communards
Joe Cocker

[Get CD Catalog](#)

Práctica de Laboratorio – AJAX

Sugerencias en tiempo real

JSP Page - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

< y C... JSP Page JSP Page x > +

localhost Google

Indica un estado de México para conocer su capital:
C

Sugerencia: [Campeche – Campeche] [Coahuila – Saltillo] [Colima – Colima] [Chiapas – Tuxtla Gutierrez] [Chihuahua – Chihuahua]



JSP Page - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

< y C... JSP Page JSP Page x > +

localhost Google

Indica un estado de México para conocer su capital:
CO

Sugerencia: [Coahuila – Saltillo] [Colima – Colima]

HINT

Práctica de Laboratorio – AJAX

Sugerencias en tiempo real

Tips:

Servlet

Solicitud AJAX Get con parámetro

Event onkeyup

Arreglo con posibles sugerencias

toLowerCase()

substring(0, n)

contains(String)

out.write(hints)



WebApplicationAjaxHints

HINT



&



- Descargar jar



<http://code.google.com/p/json-simple/downloads/detail?name=json-simple-1.1.1.jar&can=2&q=>



Empaquetando respuestas JSON en Servlets

```
JSONObject jsonResponse = new JSONObject();
```

Agrega nuevo atributo al objeto JSON

```
jsonResponse.put("field1", "value1");  
jsonResponse.put("field2", "value2");  
jsonResponse.put("field3", "value3");
```

Se indica que la respuesta contiene objetos JSON

```
response.setContentType("application/json");
```

```
out.write(jsonResponse.toString());
```



Empaquetando respuestas JSON en Servlets

```
JSONObject jsonResponse = new JSONObject();
```

```
JSONArray addresses = new JSONArray();
```

```
for (int i=0 ;i<10; i++) {  
    JSONObject address = new JSONObject();  
    address.put("name" , "Decepticon" + i);  
    address.put("id" , i);  
    addresses.add(address);  
}
```

```
jsonResponse.put("addresses", addresses);  
response.setContentType("application/json");  
out.write(jsonResponse.toString());
```

La respuesta contiene un sólo atributo (**addresses**) tipo arreglo



Desempaquetando respuestas JSON con JavaScript

Función de
JavaScript

```
ajaxRequest.onreadystatechange = function(){
    if (ajaxRequest.readyState==4 && ajaxRequest.status==200){
        var jsonObject = JSON.parse(ajaxRequest.responseText);
        var txt = "field1: " + jsonObject.field1 +
                  " field2: " + jsonObject.field2 +
                  " field3: " + jsonObject.field3 +
                  "<br />";
        document.getElementById(id).innerHTML=txt;
    }
}
```

Los atributos
se acceden
normalmente



Práctica de Laboratorio – AJAX & JSON & Java DB

Desplegando detalles

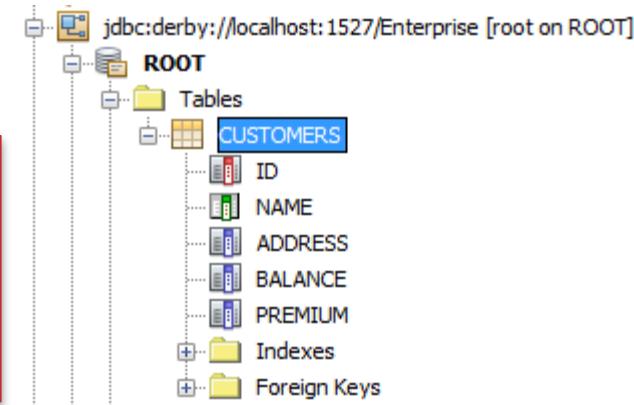
Select a customer ▾ No details

Select a customer

John Doe
Juan Perez
Octavio Gutierrez



Extraer clientes
de la base de
datos (onload())
usando un
Servlet



ID: 3
NAME: Juan Perez
ADDRESS: 321 Washington Blvd
BALANCE: 111
PREMIUM: false

Juan Perez ▾



Cuando se seleccione
un nuevo cliente
mostrar sus datos
(onchange())
usando un servlet

Select a customer ▾ No details



Práctica de Laboratorio – AJAX & JSON & Java DB

Desplegando detalles

Tips:

```
<body onload="getCustomerNames()">  
  
<select name='customers' onchange='getDetails(this.value)'>
```

GET DETAILS

