

ajax

***AJAX: Asynchronous
JavaScript and XML***



Ajax

- Con Ajax se pueden actualizar las páginas web de forma asíncrona.
- Ajax permite intercambiar datos entre el cliente y el servidor en segundo plano.
- Usando Javascript se podrán actualizar partes de la página web, sin necesidad de tener que recargarla.



XMLHttpRequest

```
var peticionAsincrona;  
if(window.XMLHttpRequest){  
    peticionAsincrona = new XMLHttpRequest();  
}
```



Métodos

- El método **open(metodo, url, asíncrono)** se usa para definir la petición asíncrona.
- El método **send(cadena)** se usa para realizar la petición.



GET o POST

- Get es más simple.
- Get es más rápido.
- Post permite enviar mayor cantidad de datos.



GET

```
peticionAsincrona.open("GET",  
    "select.php?buscar=dato", true);  
peticionAsincrona.send();
```



POST

```
peticionAsincrona.open("POST",  
    "select.php", true);  
peticionAsincrona.setRequestHeader(  
    "Content-type",  
    "application/x-www-form-urlencoded");  
peticionAsincrona.send("buscar=dato");
```



Codificación de cadenas

- Tanto en las peticiones asíncronas por GET como por POST, es posible que se envíen datos que hayan sido introducidos por el usuario, para ello usamos la función **encodeURIComponent()** o **encodeURIComponent()**.

encodeURIComponent(dato)



Respuesta del servidor

- El servidor debe generar una respuesta a la petición asíncrona.
- La respuesta puede ser en texto plano o en XML.
- La respuesta en texto plano suele ser una respuesta en formato JSON.



Respuesta XML

```
<?php
header('Content-type: application/xml');
echo '<?xml version = "1.0" encoding = "utf-8"?>';
echo '<respuesta>';
...
echo '</respuesta>';
```



Respuesta JSON

```
<?php
header('Content-Type: application/json');
echo '{"respuesta":
    [{"atributo":"valor", "atributo2": "valor2"},
     {...}, ...]
}';
```



onreadystatechange I

- Permite definir la función que va a recibir la respuesta del servidor.
- readyState, es un valor de 0 a 4 que indica el estado de la respuesta, cuando vale 4 ha finalizado
- status, su valor es 200 si todo se ha realizado correctamente



onreadystatechange II

```
peticionAsincrona.onreadystatechange = function() {  
  if (peticionAsincrona.readyState == 4 &&  
      peticionAsincrona.status == 200){  
    var respuesta = peticionAsincrona.responseText;  
    // o  
    var respuesta = peticionAsincrona.responseXML;  
  }  
}
```

En lugar de onreadystatechange, se puede usar también **onload**.

Los valores de readyState son: 0 - unsent, 1 - opened, 2 - headers_received, 3 - loading, 4 - done. Los valores de status son los valores que utiliza el protocolo http. El código 200 corresponde a la recepción de una respuesta correcta.



responseType

Antes de lanzar la petición se puede solicitar el tipo de la respuesta:

- "arraybuffer"
- "blob"
- "document"
- "json"
- "text"
- "" (default)



Procesar respuesta XML I

```
function procesar(xml){  
  var respuesta = xml.documentElement.  
    getElementsByTagName("producto");  
  for (var i = 0; i < respuesta.length; i++) {  
    var precio=respuesta[i].getAttribute("precio");  
    var nombre=respuesta[i].getAttribute("nombre");  
  }  
}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<respuesta>  
  <producto precio="1.12" nombre="pera"></producto>  
  <producto precio="1.14" nombre="pera blanca"></producto>  
</respuesta>
```



Procesar respuesta XML II

```
function procesar(xml){  
  var respuesta = xml.documentElement.  
    getElementsByTagName("producto");  
  for (var i = 0; i < respuesta.length; i++) {  
    var precio = respuesta[i].getElementsByTagName  
      ("precio")[0].childNodes[0].nodeValue;  
    var nombre=respuesta[i].  
      getElementsByTagName("nombre")[0].childNodes[0].nodeValue;  
  }  
}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<respuesta>  
  <producto><precio>1.12</precio><nombre>pera</nombre></producto>  
  <producto><precio>1.14</precio><nombre>pera blanca</nombre></producto>  
</respuesta>
```



Procesar respuesta JSON

```
function procesar(json){  
  var ojson = eval("(+json+)"); //opción primera  
  var ojson = JSON.parse(json); //opción segunda, preferida  
  for(var i=0;i<ojson.respuesta.length;i++){  
    var precio = ojson.respuesta[i].precio;  
    var nombre = ojson.respuesta[i].nombre;  
  }  
}
```

```
{"respuesta":[{"precio":"1.12","nombre":"pera"}, {"precio":"1.14","nombre":"pera  
blanca"}]}
```

