

# Lenguaje PHP

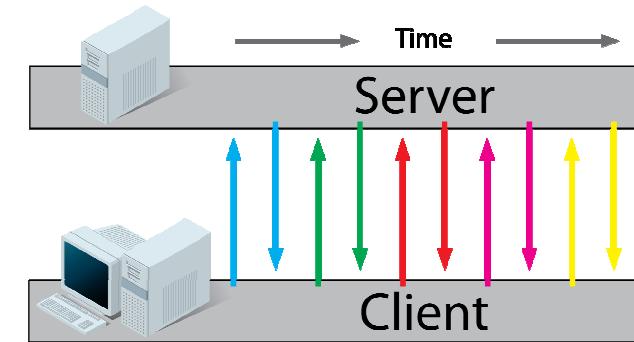
## AJAX



# Lenguaje PHP: comunicación Cliente-Servidor

- El protocolo de aplicación HTTP/S presenta incomodidades debido a su comportamiento **stateless**:

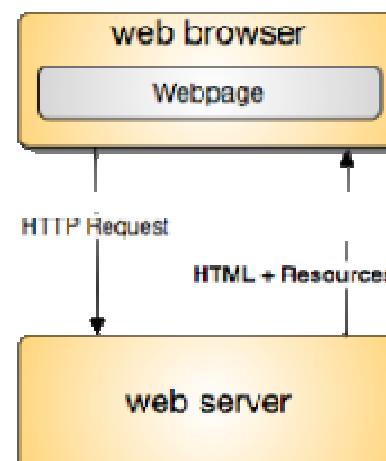
Cada visita del cliente al servidor se toma como una conexión aislada e independiente (sin sesiones).



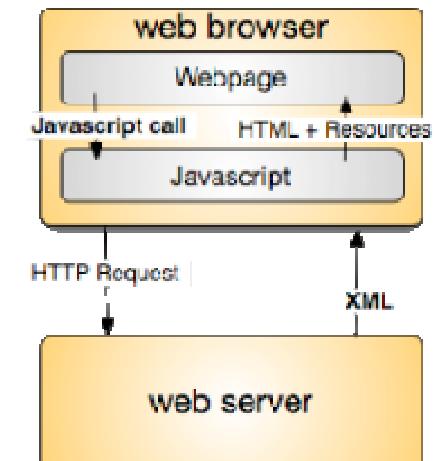
- Además la **comunicación entre cliente y servidor es síncrona**: el intercambio de información entre cliente y servidor sólo tiene lugar **si se recarga toda la web** (mediante GET o POST).

En las aplicaciones de escritorio no ocurre este fenómeno, resultando más interactivas.

Traditional web model



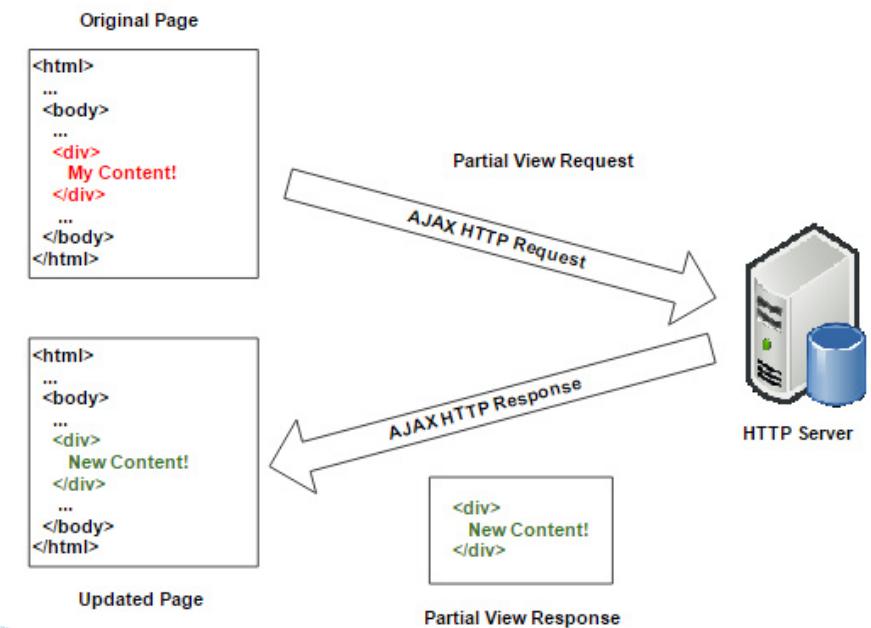
AJAX web model



# Lenguaje PHP: actualización parcial de páginas - AJAX

- **AJAX** (Asynchronous JavaScript + XML): consiste en actualizar el contenido de una página accediendo al servidor web pero **sin recargar toda la página**, sólo aquella sección HTML que deseemos.

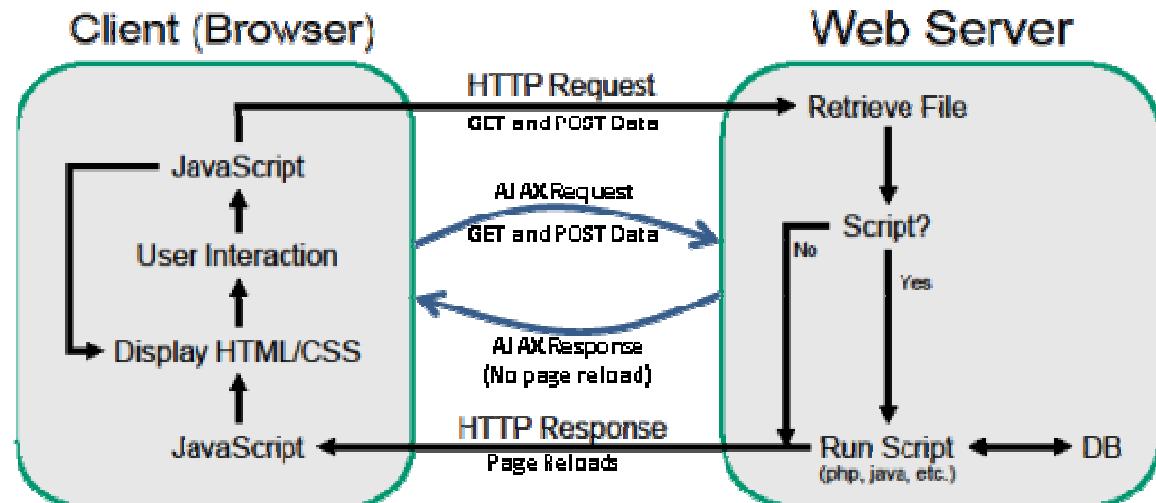
- **AJAX evita el comportamiento síncrono** de las aplicaciones web:  
permite la comunicación asíncrona entre cliente y servidor (no al unísono): el navegador (cliente) se puede comunicar con el servidor sin tener que recargar toda la web  
→ permite recargar el contenido de parte/s de una web.



# Lenguaje PHP: AJAX - ¿Por qué recargar lo que no cambia?

- Recargar toda una web para actualizar únicamente el contenido de una sección es ineficiente.
- AJAX permite realizar un **refresco selectivo** de las secciones de la web cuyo contenido haya cambiado.

AJAX mejora la interactividad, **velocidad** y **amabilidad** de las aplicaciones web, comportándose casi como si fueran aplicaciones de escritorio.



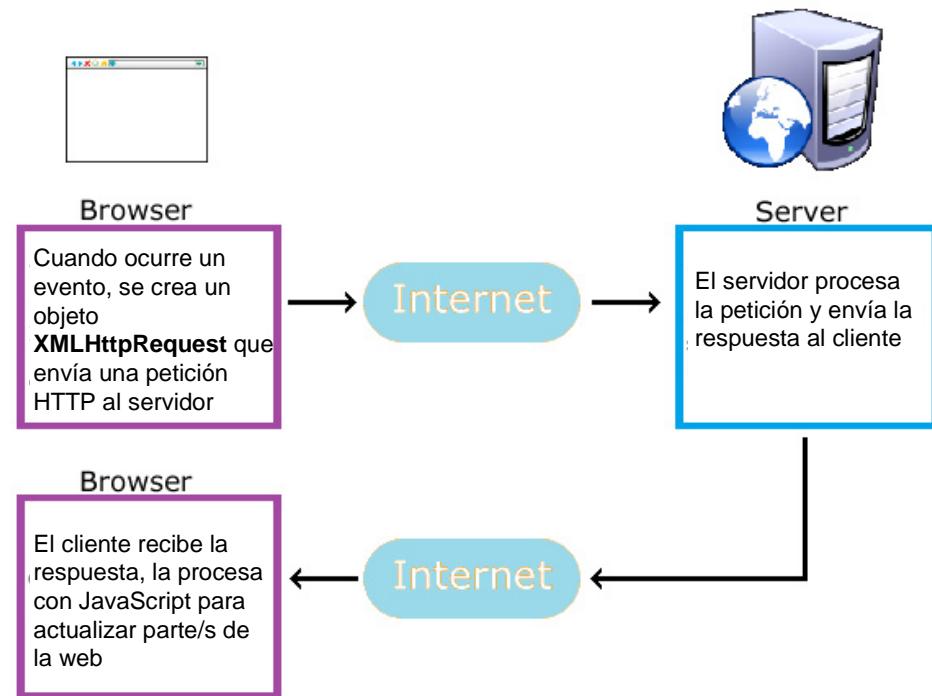
# Lenguaje PHP: Funcionamiento de AJAX

- **Funcionamiento de AJAX:**

**1º.- Cuando sucede un evento en el cliente:** cierto código JavaScript de la web llama a un fichero PHP del lado del servidor y le pasa variables.

**2º.- El fichero PHP del lado del servidor genera un código JavaScript de respuesta.**

**3º.- Este código JavaScript se ejecuta en la página para modificar el código HTML de la página, o bien realizar alguna otra acción.**



El objeto XMLHttpRequest permite al navegador comunicarse con el servidor sin requerir una recarga completa de la página.



# Lenguaje PHP: AJAX

- **Ejemplo:** Aplicación que genera un número aleatorio refrescando sólo la parte de la web donde éste se visualiza (div) → el vídeo no se refresca.

```
<html>
<head> <title>Ejemplo recargar con AJAX</title>
      <script language="JavaScript" type="text/javascript">
          function generar () {
              var nuevoscript = document.createElement("script");
              nuevoscript.src = 'recargar.php';
              nuevoscript.setAttribute('language', 'javascript');
              nuevoscript.setAttribute('type', 'text/javascript');
              document.body.appendChild(nuevoscript);
          }
      </script>
  </head>
<body bgcolor="#AAAAFF" onLoad="generar();">
    <br> <iframe width="560" height="315"
    src="https://www.youtube.com/embed/515Xn_LOEY4?rel=0&autoplay=1" frameborder="0"
    allowfullscreen></iframe> <br>
    <p> Nuevo número aleatorio generado: <p>
        <div id="numero">
            <!-- el contenido es generado por AJAX -->
        </div>
        <br>   <input type="submit" onClick="generar();" value="GENERAR OTRO NÚMERO" >
    </body>
</html>
```

Cuando sucede el **evento** `onClick` en el cliente, el código JavaScript de la web llama a un fichero PHP que a su vez genera código JavaScript para actualizar el contenido de la sección (div) donde se muestra el número → genera un nuevo número.

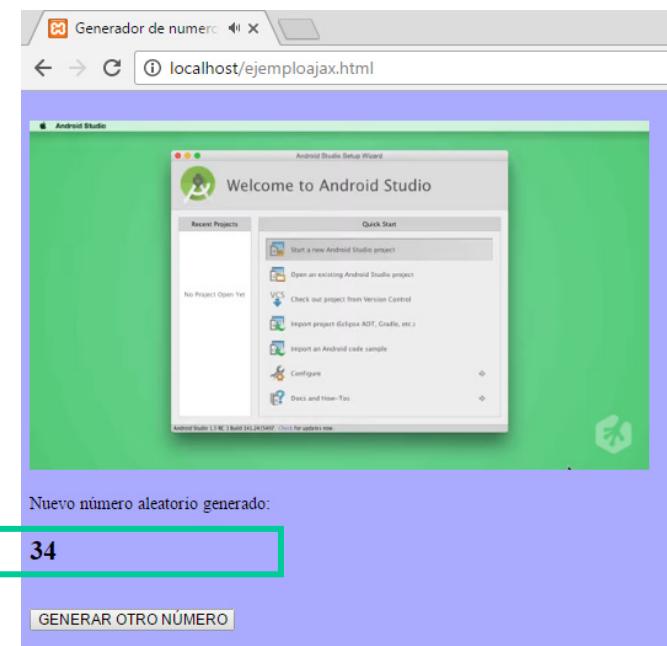
# Lenguaje PHP: AJAX

- Archivo **recargar.php**

```
<?php  
    header( "Content-type: text/javascript" );  
?  
  
    <!-- código JavaScript que es enviado al cliente -->  
    document.getElementById("numero").innerHTML=<h2><?php echo rand(1,99);?></h2>;
```



El código JavaScript generado se encarga de actualizar el contenido de la sección (div) donde se muestra el número. El resto de la web permanece igual ;-)



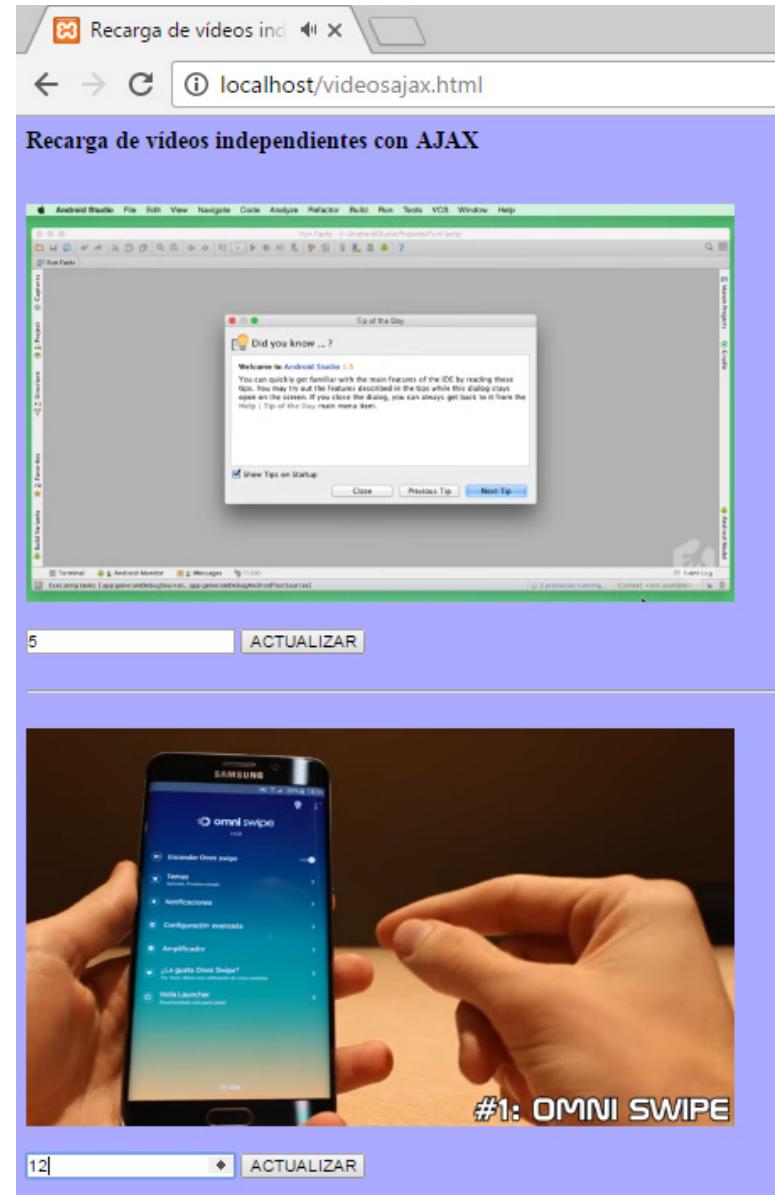
# Lenguaje PHP: AJAX

- **Actividad:** crear una aplicación web que muestre dos vídeos diferentes de YouTube embebidos en la propia página.

Cada vídeo contará con un control para introducir el tiempo (en segundos) a partir del cual queremos que se reproduzca.

Si se actualiza el tiempo de comienzo de un vídeo, el otro vídeo debe permanecer reproduciéndose por donde iba.

El código de recarga de ambos vídeos debe implementarse en *un único archivo PHP* que reciba mediante el método GET (en la URL) los parámetros necesarios.



# Lenguaje PHP: AJAX

## • Archivo videosAJAX.html

```
<html> <head> <title>Recarga de v&iacute;deos independientes con AJAX</title>
    <script language="JavaScript" type="text/javascript">
        function recargar (id, tiempo) {
            var nuevoscript = document.createElement("script");
            nuevoscript.src = 'recargarVideos.php?id=' + id + '&tiempo=' + tiempo;
            nuevoscript.setAttribute('language', 'javascript');
            nuevoscript.setAttribute('type', 'text/javascript');
            document.body.appendChild(nuevoscript);
        }
    </script>
</head>
<body bgcolor="#AAAAFF">
    <h3>Recarga de v&iacute;deos independientes con AJAX</h3> <br>
    <div id="video1">
        <iframe width="560" height="315" src="https://www.youtube.com/embed/515Xn_LOEY4?rel=0&autoplay=1"
            frameborder="0" allowfullscreen></iframe>
    </div>
    <br> <input type="number" id="tiempoVideo1" min="0">
    <input type="submit" onClick="recargar('video1',document.getElementById('tiempoVideo1').value);"
        value="ACTUALIZAR">
    <br> <br> <hr> <br>
    <div id="video2">
        <iframe width="560" height="315" src="https://www.youtube.com/embed/voywIbE8P7M?rel=0&autoplay=1"
            frameborder="0" allowfullscreen></iframe>
    </div>
    <br> <input type="number" id="tiempoVideo2" min="0">
    <input type="submit" onClick="recargar('video2',document.getElementById('tiempoVideo2').value);"
        value="ACTUALIZAR">
</body> </html>
```

Para que JavaScript invoque al archivo PHP que recarga pasándole parámetros, se emplea una URL con el método GET: URL?campo=valor&campo=valor...

# Lenguaje PHP: AJAX

- Archivo **recargarVideos.php**

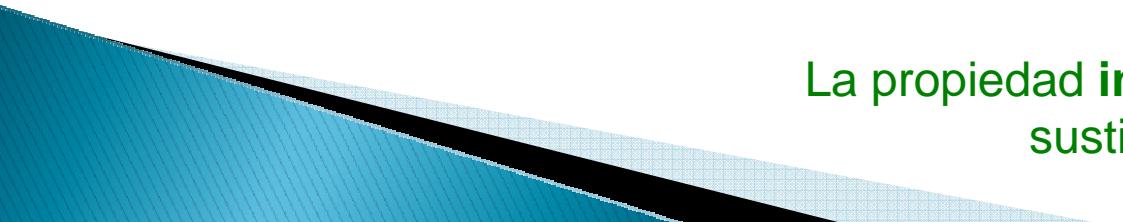
```
<?php
    header("Content-type: text/javascript");

    if (isset($_GET['id'])) {
        // Parámetros enviados en la URL para recargar un vídeo en particular:
        $id = $_GET['id'];
        $tiempo = $_GET['tiempo'];

        $paramVideo = "?rel=0&showinfo=0&autoplay=1&start=".$tiempo."' frameborder='0' allowfullscreen";

        if ($id == 'video1') {
            $video = "<iframe width='560' height='315' src='https://www.youtube.com/embed/515Xn_LOEY4"
                    . $paramVideo . "></iframe>";
        } else {
            $video = "<iframe width='560' height='315' src='https://www.youtube.com/embed/voywIbE8P7M"
                    . $paramVideo . "></iframe>";
        }

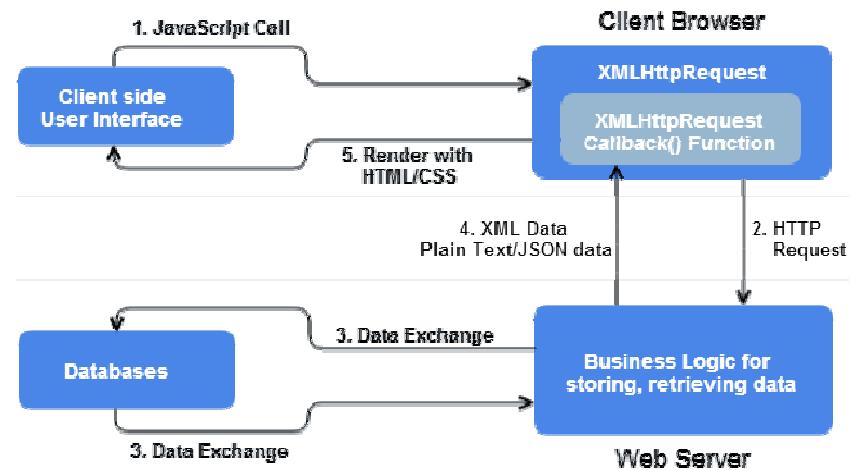
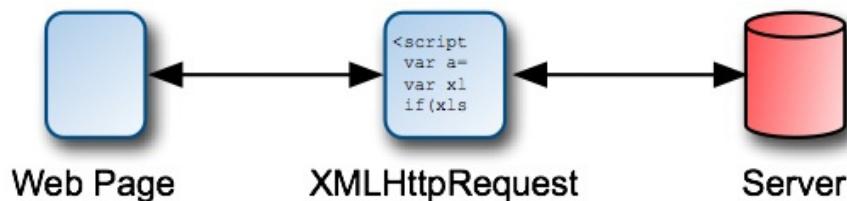
        // código JavaScript generado para recargar el vídeo en cuestión y en el segundo indicado:
        echo "document.getElementById(\"" . $id . "\").innerHTML=\\" . $video . "\"";
    }
?>
```



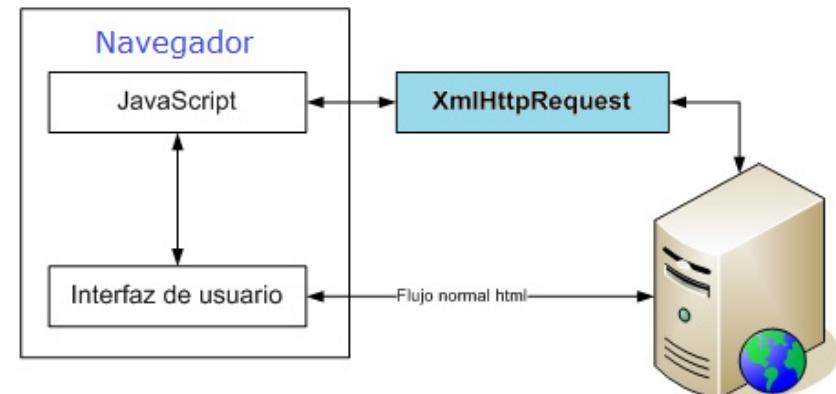
La propiedad **innerHTML** de un elemento del DOM sustituye el contenido HTML que tiene el elemento por el indicado.

# Lenguaje PHP: Objeto XMLHttpRequest

- El objeto JavaScript **XMLHttpRequest** permite al navegador comunicarse con el servidor sin requerir una recarga completa de la página:



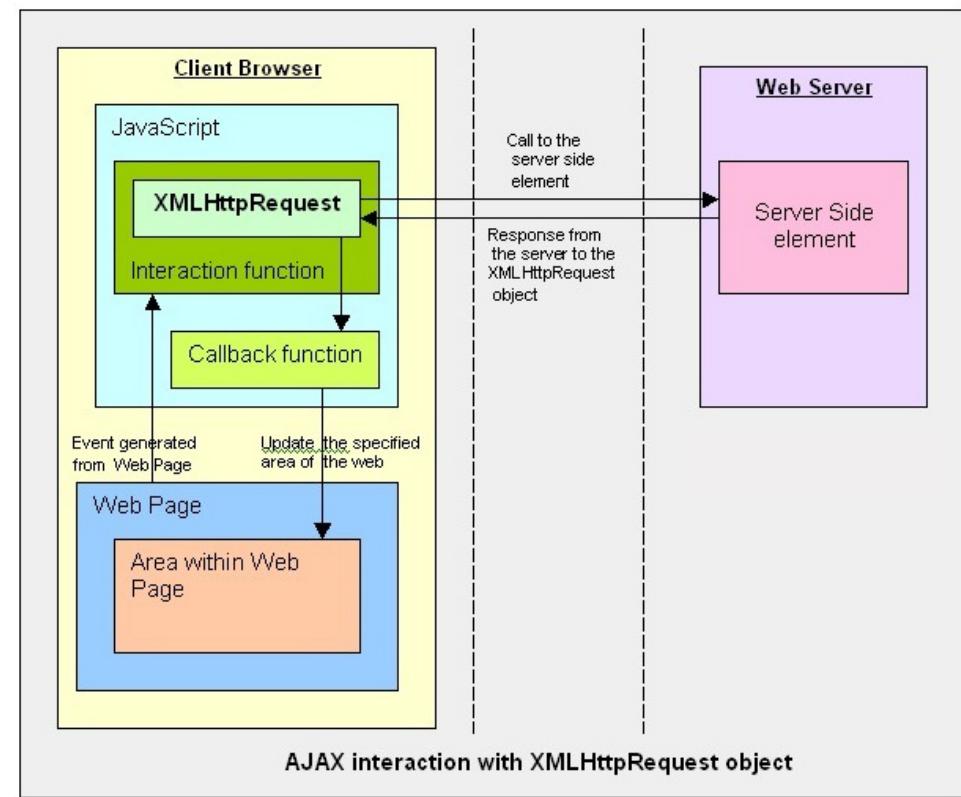
- Este objeto encapsula la petición de actualización; y el procesamiento de la respuesta del servidor con la información actualizada:



**Más info** sobre el objeto **XMLHttpRequest**:  
[http://www.w3schools.com/xml/dom\\_httprequest.asp](http://www.w3schools.com/xml/dom_httprequest.asp)

# Lenguaje PHP: Objeto XMLHttpRequest

- **Funcionamiento AJAX con XMLHttpRequest:** cuando ocurre algún **evento** sobre algún control, elemento o sección de la página web:
  1. Se crea con JavaScript un objeto XMLHttpRequest.
  2. Se crea la función que se ejecutará cuando el server devuelva la respuesta.
  3. Se envía la petición a un archivo PHP en el servidor.
  4. El servidor ejecuta el script PHP para procesar la petición; y envía la respuesta a la página web.
  5. JavaScript lee la respuesta (responseText) y ejecuta la acción prevista para actualizar la sección de la página web.



# Lenguaje PHP: Objeto XMLHttpRequest

- **Ejemplo:** web que lanza datos que se actualizan mediante AJAX

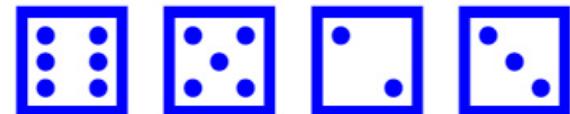
```
<html>
<head> <script>
    function lanzar (valor) {
        if (valor.length == 0) {
            document.getElementById("dados").innerHTML = "";
        } else {
            var xhr = new XMLHttpRequest();
            // función JavaScript que se ejecutará al recibir la respuesta PHP del server:
            xhr.onreadystatechange = function() {
                if (this.readyState == 4 && this.status == 200) {
                    document.getElementById("dados").innerHTML = this.responseText;
                }
            };
            xhr.open("GET", "lanzardados.php?cuantos=" + valor, true); // URL con parámetros
            xhr.send();
        }
    }
</script> </head>
<body>
    <p><b>Indique la cantidad de datos que desea lanzar:</b></p>
    <form><input type="number" min="1" max="10" value="1" onChange="lanzar(this.value)"></form>
    <hr>
    <div id="dados" style="font-size:120px; color:blue;"> </div>
</body> </html>
```

**Usando el método  
GET para el envío**

# Lenguaje PHP: Objeto XMLHttpRequest

```
<?php
    $cuantos = $_REQUEST['cuantos']; // equivale a $_GET[] o $_POST[] según el método de envío usado
    if ($cuantos > 0) {
        $resultado = "";
        for ($i = 0; $i < $cuantos; $i++) {
            $dato = rand(1, 6);
            switch ($dato) {
                case 1:
                    $resultado .= "&#9856"; break;
                case 2:
                    $resultado .= "&#9857"; break;
                case 3:
                    $resultado .= "&#9858"; break;
                case 4:
                    $resultado .= "&#9859"; break;
                case 5:
                    $resultado .= "&#9860"; break;
                case 6:
                    $resultado .= "&#9861"; break;
                default:
                    break;
            }
        }
    } else {
        $resultado = "Cantidad de dados errónea";
    }
    echo $resultado; // código HTML devuelto a la página y que es recuperado por JavaScript
                     // mediante la propiedad responseText del objeto XMLHttpRequest
?>
```

← archivo [lanzardados.php](#)



# Lenguaje PHP: Objeto XMLHttpRequest

- **Ejemplo:** web que lanza datos que se actualizan mediante AJAX

```
<html> <head> <script>
    function lanzar (valor) {
        if (valor.length == 0) {
            document.getElementById("dados").innerHTML = "";
        } else {
            var xhr = new XMLHttpRequest();
            // código JavaScript que se ejecutará al recibir la respuesta del PHP del server:
            xhr.onreadystatechange = function() {
                if (this.readyState == 4 && this.status == 200) {
                    document.getElementById("dados").innerHTML = this.responseText;
                    // alert(this.responseText);
                }
            };
            xhr.open("POST", "lanzardados.php", true);
            xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
            var parametros = "cuantos=" + valor;
            xhr.send(parametros);
        }
    }
</script> </head> <body>
    <p><b>Indique la cantidad de datos que desea lanzar:</b></p>
    <input id="cantidad" type="number" min="1" max="10" value="1">
    <input type="submit" value="LANZAR" onClick="lanzar(document.getElementById('cantidad').value);">
    <hr>
    <div id="dados" style="font-size:120px; color:blue;">  </div>
</body> </html>
```

**Usando el método  
POST para el envío**

¡OJO! No es necesario usar FORMS cuando empleamos AJAX mediante POST

**No es necesario cambiar absolutamente nada en el archivo lanzardados.php**

# Lenguaje PHP: Objeto XMLHttpRequest

- **Actividad:** desarrollar una web que realiza una **encuesta online**. Cada vez que un usuario vote, se debe actualizar el resultado de la encuesta mediante AJAX mostrando una gráfica de barras y los porcentajes de cada opción:



Para almacenar los votos que la encuesta vaya generando, se empleará un **archivo en el servidor** (encuesta.dat) con la siguiente estructura:

```
SI=13  
NO=7  
<EOF>
```

Modifica la página para evitar que el internauta pueda votar más de una vez durante la misma sesión. En caso de que lo intente, debes mostrarle un mensaje de error (p.e. con alert de JavaScript).

# Lenguaje PHP: Objeto XMLHttpRequest

- **1º.- Sin usar AJAX ni sesiones:** recarga toda la web al votar y permite al usuario votar más de una vez:

```
<?php
    $SI = $NO = 0;

    if (!file_exists("encuesta.dat")) {
        // Crear el fichero si no existe e inicializar la cantidad de SI y NO
        $fichero = fopen("encuesta.dat", "w");
        $si = "SI=0"; fputs($fichero, $si.PHP_EOL);
        $no = "NO=0"; fputs($fichero, $no.PHP_EOL);
        fclose($fichero);
    }

    $fichero = fopen("encuesta.dat", "r+"); // lee los valores almacenados
    $linea = fgets($fichero); $valores = explode("=", $linea); $SI = (int)$valores[1];
    $linea = fgets($fichero); $valores = explode("=", $linea); $NO = (int)$valores[1];

    if ($_SERVER["REQUEST_METHOD"] == "POST") { // actualiza los resultados de la encuesta
        if (isset($_POST['si']))
            $SI++;
        if (isset($_POST['no']))
            $NO++;
        $si = "SI=". $SI;
        $no = "NO=". $NO;
        rewind($fichero); fputs($fichero, $si.PHP_EOL); fputs($fichero, $no.PHP_EOL);
    }
    fclose($fichero);

?>
```

← archivo  
[encuestaFichero.php](#)

# Lenguaje PHP: Objeto XMLHttpRequest

```
<!DOCTYPE html>
<html>
    <head><title>Encuesta</title></head>
    <body>
        <p><span style="font-family: verdana; font-size: 18px; color: #0022DD; font-weight: bold;"> ¿Está de acuerdo con...? </span></p>

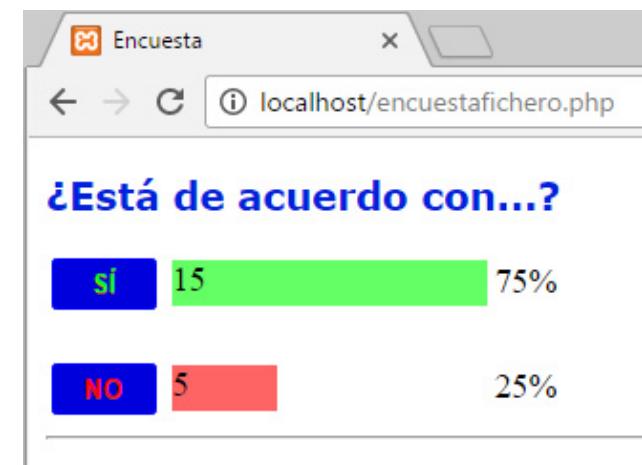
        <form method="post" action=<?php echo htmlspecialchars($_SERVER[ "PHP_SELF" ] ) ;?>>
            <table> <tr>
                <td>
                    <input type="submit" name="si" value="SÍ" style="height: 24px; width: 50px;
                        border-radius:2px 2px 2px 2px; border:1px solid #0000FF; background: #0000DD; font-family: Arial;
                        font-size: 12px; color: #00FF00; text-shadow: 0 1px #aa4040; font-weight: bold;">
                </td>
                <td></td>
                <td> <?php
                    $ancho = $SI * 10;
                    echo "<div style=\"width:".$ancho."px; height:22px; background:rgb(100,255,100);\">";
                    echo $SI;
                    echo "</div>";
                ?> </td>
                <td> <?php if ($SI+$NO != 0) {
                    $porcentajeSi = round($SI/($SI+$NO) * 100);
                    echo $porcentajeSi . "%";
                } else {
                    echo "0%";
                }
                ?> </td>
            </tr> <tr><td><br></td><td></td></tr>
```

← continuación del archivo **encuestaFichero.php**

# Lenguaje PHP: Objeto XMLHttpRequest

```
<tr>
  <td>
    <input type="submit" name="no" value="NO" style="height: 24px; width: 50px; border-radius: 2px 2px 2px 2px; border:1px solid #0000FF; background: #0000DD; font-family: Arial; font-size: 12px; color: #FF0000; text-shadow: 0 1px #aa4040; font-weight: bold;">
  </td>
  <td></td>
  <td> <?php
    $ancho = $NO * 10;
    echo "<div style=\"width:".$ancho."px; height:22px; background:rgb(255,100,100);\">";
    echo $NO;
    echo "</div>";
  ?> </td>
  <td> <?php if ($SI+$NO != 0) {
      $porcentajeNo = round($NO/($SI+$NO) * 100);
      echo $porcentajeNo. "%";
    } else {
      echo "0%";
    }
  ?> </td>
</tr>
</table>
</form>
<hr>
</body>
</html>
```

← continuación del archivo  
encuestaFichero.php



# Lenguaje PHP: Objeto XMLHttpRequest

- **2º.- Usando AJAX y sesiones:** recarga solo lo necesario y solo permite votar una vez por sesión:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Encuesta</title>
        <script>
            function actualizarEncuesta (id, modo) {
                var xhr = new XMLHttpRequest();
                xhr.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        if (this.responseText.includes('Error')) {
                            alert(this.responseText);
                        } else {
                            document.getElementById(id).innerHTML = this.responseText;
                        }
                    }
                };
                xhr.open("POST", "actualizarEncuesta.php", true);
                xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
                var parametros = "opcion=" + id + "&" + "modo=" + modo;
                xhr.send(parametros);
            }
        </script>
    </head>

    <body bgcolor="#DDDDFF" onLoad="actualizarEncuesta('si','mostrar');
        actualizarEncuesta('no','mostrar'); actualizarEncuesta('porcentajeSi','mostrar');
        actualizarEncuesta('porcentajeNo','mostrar');">
```

← archivo

**encuestaFicheroAJAX.html**

Puedes consultar el siguiente ejemplo:  
[http://www.w3schools.com/php/php\\_ajax\\_poll.asp](http://www.w3schools.com/php/php_ajax_poll.asp)

# Lenguaje PHP: Objeto XMLHttpRequest

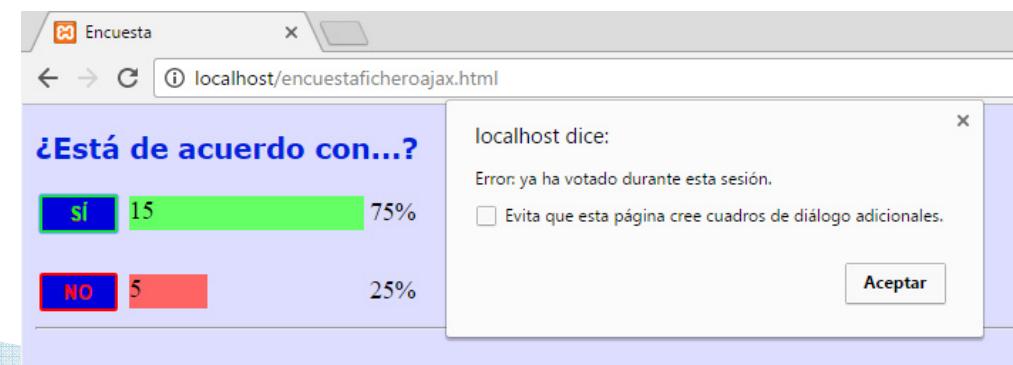
```
<p><span style="font-family: verdana; font-size: 18px; color: #0022DD; font-weight: bold;">  
    ¿Estás de acuerdo con...? </span></p>  
  
<table>  
<tr>  
    <td> <input type="submit" value="SI" style="height: 24px; width: 50px;  
        border-radius: 2px 2px 2px 2px; border: 2px solid #00FF00; background: #0000DD; font-family: Arial;  
        font-size: 12px; color: #00FF00; text-shadow: 0 1px #aa4040; font-weight: bold;"  
        onClick="actualizarEncuesta('si', 'actualizar'); actualizarEncuesta('porcentajeSi',  
        'mostrar'); actualizarEncuesta('porcentajeNo', 'mostrar');"> </td>  
    <td></td>  
    <td> <div id="si"> </div> </td>  
    <td> <div id="porcentajeSi"> </div> </td>  
</tr>  
<tr><td><br></td><td></td></tr>  
<tr>  
    <td> <input type="submit" value="NO" style="height: 24px; width: 50px; border-radius: 2px 2px  
        2px 2px; border: 2px solid #FF0000; background: #0000DD; font-family: Arial; font-size:  
        12px; color: #FF0000; text-shadow: 0 1px #aa4040; font-weight: bold;"  
        onClick="actualizarEncuesta('no', 'actualizar'); actualizarEncuesta('porcentajeNo',  
        'mostrar'); actualizarEncuesta('porcentajeSi', 'mostrar');"> </td>  
    <td></td>  
    <td> <div id="no"> </div> </td>  
    <td> <div id="porcentajeNo"> </div> </td>  
</tr>  
</table>  
<hr>  
</body>  
</html>
```

← continuación del archivo  
[encuestaFicheroAJAX.html](#)

# Lenguaje PHP: Objeto XMLHttpRequest

```
<?php
    session_start();
    if (isset($_REQUEST['opcion'])) {
        $SI = $NO = 0;      $resultado = $error = "";
        if (!file_exists("encuesta.dat")) { // crear el fichero si no existe e inicializa
            $fichero = fopen("encuesta.dat", "w");
            $si = "SI=0";      fputs($fichero, $si.PHP_EOL);
            $no = "NO=0";      fputs($fichero, $no.PHP_EOL);
            fclose($fichero);
        }
        $fichero = fopen("encuesta.dat", "r+"); // lee los valores almacenados
        $linea = fgets($fichero); $valores = explode("=", $linea); $SI = (int)$valores[1];
        $linea = fgets($fichero); $valores = explode("=", $linea); $NO = (int)$valores[1];
        if ($_REQUEST['modo'] == 'actualizar') {
            if (!isset($_SESSION['ha_votado'])) {
                if ($_REQUEST['opcion'] == 'si') $SI++;
                if ($_REQUEST['opcion'] == 'no') $NO++;
                $si = "SI=".$SI; $no = "NO=".$NO;
                rewind($fichero);
                fputs($fichero, $si.PHP_EOL); fputs($fichero, $no.PHP_EOL);
                fclose($fichero);
                $_SESSION['ha_votado'] = "ha_votado";
            } else
                $error = "Error ya ha votado.";
        }
    }
    <?php
```

← archivo actualizarEncuesta.php



# Lenguaje PHP: Objeto XMLHttpRequest

```
if ($_REQUEST['opcion'] == 'si') {
    $ancho = $SI * 10;
    $resultado = "<div style=\"width:".$ancho."px; height:22px; background:rgb(100,255,100);\">";
    $resultado .= $SI."</div>";
}
if ($_REQUEST['opcion'] == 'no') {
    $ancho = $NO * 10;
    $resultado = "<div style=\"width:".$ancho."px; height:22px; background:rgb(255,100,100);\">";
    $resultado .= $NO. "</div>";
}
if ($_REQUEST['opcion'] == 'porcentajeSi') {
    if ($SI+$NO != 0)
        $resultado = round($SI/($SI+$NO)*100). "%";
    else
        $resultado = "0%";
}
if ($_REQUEST['opcion'] == 'porcentajeNo') {
    if ($SI+$NO != 0)
        $resultado = round($NO/($SI+$NO)*100). "%";
    else
        $resultado = "0%";
}
if (!empty($error))
    $resultado = $error;
echo $resultado; // código HTML que se devuelve a JavaScript mediante responseText
}
?>
```

← continuación del archivo  
[actualizarEncuesta.php](#)

¿Cómo hacer que el usuario no vote más de una vez aunque cierre el navegador y termine la sesión?

→ mediante una cookie

¿Y para que no votase incluso empleando cualquier otro navegador? →

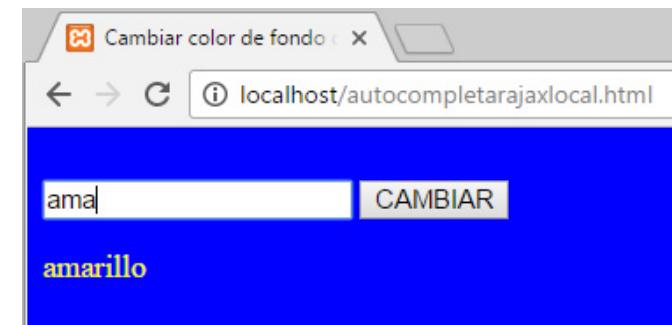
registrando en un archivo del server las IPs  
de los clientes que han votado

# Lenguaje PHP: AJAX local

- **Ejemplo AJAX local:** cambia el fondo de la web *autocompletando* el color escrito:

```
<!DOCTYPE html><html><head><title>Cambiar fondo</title>
<script>
    function autocompletarColor (color) {
        var colores = new Array("rojo", "verde", "azul",
            "amarillo");
        for (var i = 0; i < colores.length; i++) {
            if (colores[i].startsWith(color)) {
                document.getElementById("muestraColor").innerHTML = colores[i];
                cambiarColor('muestraColor', colores[i]);
            }
        }
    }

    function cambiarColor (id, color) {
        var col;
        switch (color) {
            case 'rojo': col = '#FF0000'; break;    case 'verde': col = '#00FF00'; break;
            case 'azul': col = '#0000FF'; break;    case 'amarillo': col = '#FFFF00'; break;
        }
        if (id == "body") document.body.style.backgroundColor = col;
        else document.getElementById(id).style.color = col;
    }
</script>
</head>
<body> <br>
    <input type="text" id="color" name="color" onKeyUp="autocompletarColor(this.value);">
    <input type="submit" name="cambiar" value="CAMBIAR" onClick=
        "cambiarColor('body',document.getElementById('color').value);">
    <p id="muestraColor"> color de muestra </p>
</body>
</html>
```

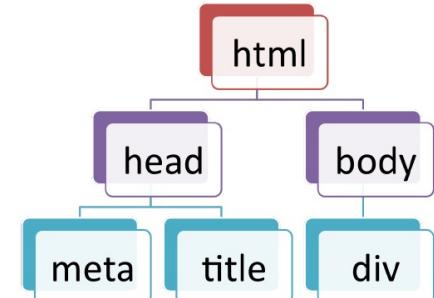


Con AJAX local, todo el código se implementa en el archivo HTML (autocompletarAJAXlocal.html)

# Lenguaje PHP: AJAX – Autocompletando valores

- El AJAX local se usa cuando los **datos necesarios** para actualizar los elementos del DOM de la página **residen en el cliente** o pueden ser calculados por el cliente **sin necesidad de acceder al servidor**.

The Document Object Model (DOM)

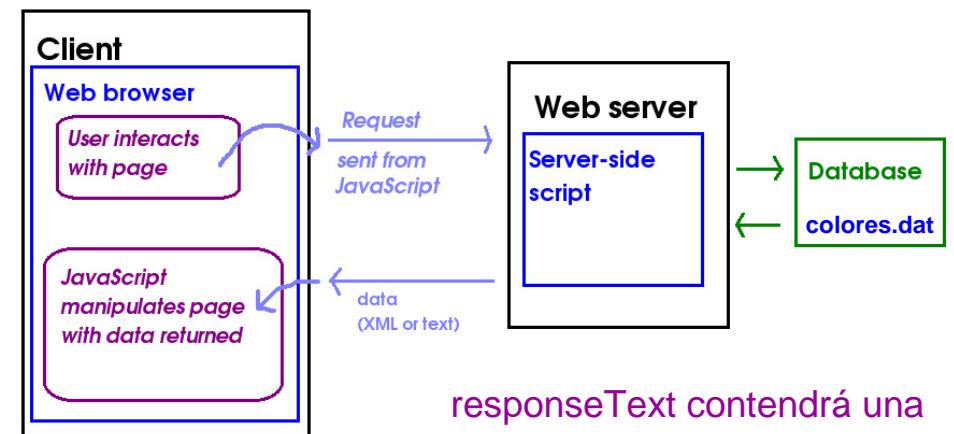


- **Actividad:** modifica el ejemplo anterior para que se *autocomplete* el color escrito mediante AJAX pero usando ahora un archivo de colores que se almacena en el servidor:



Archivo colores.dat

```
rojo:#FF0000
verde:#00FF00
azul:#0000FF
amarillo:#FFFF00
rosa:#FF00FF
cian:#00FFFF
blanco:#FFFFFF
negro:#000000
gris:#777777
violeta:#660066
marron:#996633
naranja:#FF9900
```



responseText contendrá una string con dos valores: el color autocompletado y su código

# Lenguaje PHP: AJAX

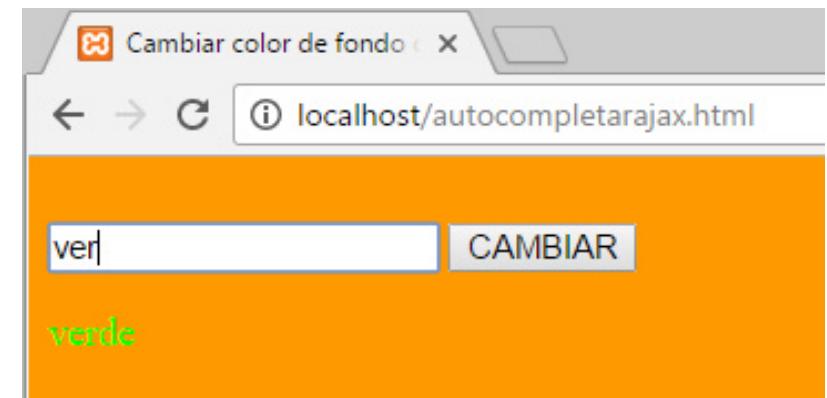
```
<!DOCTYPE html> <html> <head> <title> Cambiar color de fondo con autocompletar </title>
<script>
    var COLOR;    // almacena el código del color a aplicar
    function autocompletarColor (color) {
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                if (this.responseText.includes('Error')) {
                    alert(this.responseText);
                } else {
                    var datos = this.responseText.split(":");
                    document.getElementById('muestraColor').innerHTML = datos[0];
                    cambiarColor('muestraColor', datos[1]);
                    COLOR = datos[1];
                }
            }
        };
        xhr.open("POST", "autocompletarColor.php", true);
        xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xhr.send("color=" + color);
    }
    function cambiarColor (id, color) {
        if (id == "body")  document.body.style.backgroundColor = color;
        else                  document.getElementById(id).style.color = color;
    }
</script>
</head> <body> <br>
<input type="text" id="color" name="color" onKeyUp="autocompletarColor(this.value);">
<input type="submit" name="cambiar" value="CAMBIAR" onClick="cambiarColor('body',COLOR);">
<p id="muestraColor"> color de muestra </p>
</body> </html>
```

↑ archivo  
[autocompletarAJAX.html](#)

# Lenguaje PHP: AJAX

```
<?php
    if (!empty($_REQUEST['color'])) {
        $color = $_REQUEST['color'];
        $respuesta = "";
        if (!file_exists("colores.dat")) {
            $respuesta = "Error";
        } else {
            $fichero = fopen("colores.dat", "r"); // lee los valores almacenados
            while (!feof($fichero)) {
                $linea = fgets($fichero);
                $valores = explode(":", $linea);
                $nombreColor = $valores[0];
                $codigoColor = $valores[1];
                if (stristr($color, substr($nombreColor, 0, strlen($color)))) {
                    $respuesta = $linea;
                    break;
                }
            }
            fclose($fichero);
        }
        echo $respuesta;
    }
?>
```

↑ archivo  
**autocompletarColor.php**



# Lenguaje PHP: AJAX usando jQuery

- **Ejemplo:** web que lanza varios dados mediante AJAX local con jQuery:

```
<!DOCTYPE html> <html> <head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"> </script>
<script>
$(document).ready(principal);
function principal(){ // definición de eventos
    $("#botonLanzar").on("click", function(){
        lanzar($("#cantidad").val()); })
    }); // onClick="lanzar(document.getElementById('cantidad').value
}
function lanzar (cantidad) {
    var resultado = "";
    for (i = 0; i < cantidad; i++) {
        var dado = Math.floor((Math.random() * 6) + 1);
        switch (dado) {
            case 1: resultado += "\ud83c\udc6e"; break;
            case 2: resultado += "\ud83c\udc6f"; break;
            case 3: resultado += "\ud83c\udc70"; break;
            case 4: resultado += "\ud83c\udc71"; break;
            case 5: resultado += "\ud83c\udc72"; break; case 6: resultado += "\ud83c\udc73"; break;
        }
    }
    $("#dados").css("width", cantidad*60); // document.getElementById("dados").style.width=cantidad*60;
    $("#dados").html(resultado); // document.getElementById("dados").innerHTML = resultado;
}
</script>
</head>
<body>


Indique la cantidad de dados que desea lanzar:


<input id="cantidad" type="number" min="1" max="10" value="1">
<input id="botonLanzar" type="submit" value="LANZAR"> <br><br>
<div id="dados" style="width:200px; height:110px; background:#CCCCEE; font-size:80px; color:blue;"> </div>
</body>
</html>
```



↑ archivo  
[lanzardadosAJAXconjQuery.html](#)

# Lenguaje PHP: AJAX usando jQuery

- **Ejemplo:** actualizar contenidos con AJAX local y remoto (PHP) usando jQuery:

```
<!DOCTYPE html>
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"> </script>
    <script>
        $(document).ready(principal);

        function principal(){ // definición de eventos
            $("li").click(function(){ $(this).hide(); });

            $("p").click(function(){
                $(this).html("Mi texto ha cambiado"); //this.innerHTML="texto";
                $(this).css("color", "red"); //this.style.color="red";
            });

            $("input[name=color]:radio").click(function(){
                var color = $(this).val();
                if(color=="verde") $("body").css("color", "green");
                if(color=="azul") $("body").css("color", "blue");
            });

            $("#botonHora").click(function() { // AJAX remoto con POST:
                var valorEj1 = "4";    var valorEj2 = "10";
                $.post("hora.php", {param1: valorEj1, param2: valorEj2}, function(result){
                    $("#hora").html(result);
                });
            });
        }
    </script>
</head>
```

↑ archivo **ajaxjquery.html**

# Lenguaje PHP: AJAX usando jQuery

```
<body>
<br>
Haz click sobre los elementos para esconderlos:
<ul>
  <li> Uno </li>
  <li> Dos </li>
  <li> Tres </li>
</ul>

<p><b>Haz click para cambiar el texto</b></p>

Elige un color para el texto:
<input type="radio" name="color" value="verde">verde <input type="radio" name="color" value="azul">azul

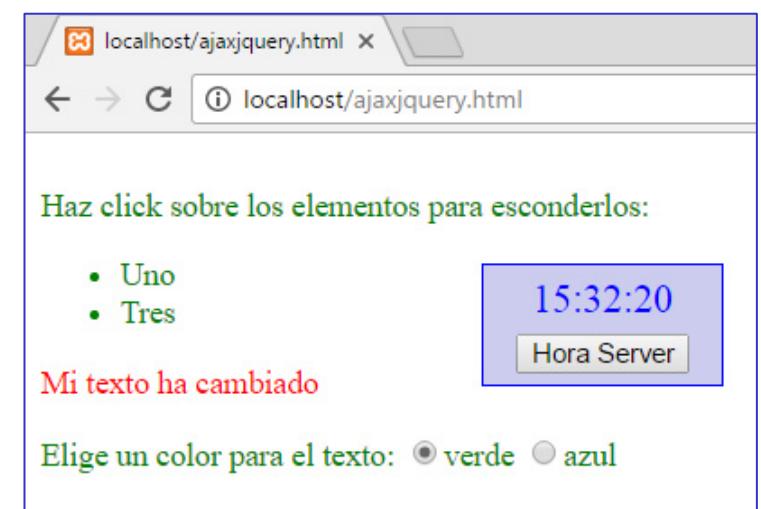
<div align="center" id="reloj" style="position:absolute; left:230px; top:65px; width:120px; height:60px;
border:1px solid blue; background:#CCCCEE; font-size:20px; color:blue;">
  <div id="hora" style="height:22px; padding: 5px 5px 2px 5px;"> </div>
  <div style="padding:2px 5px 5px 5px;"> <input id="botonHora" type="submit" value="Hora Server"> </div>
</div>

</body>
</html>
```

← continuación del  
archivo ajaxjquery.html

archivo hora.php →

```
<?php
  $parametro1 = $_REQUEST['param1'];
  $parametro2 = $_REQUEST['param2'];
  $respuesta = date("H:i:s");
  echo $respuesta;
?>
```



# AJAX: resumen

## Consideraciones a tener en cuenta para implementar AJAX:

- Elimina la etiqueta `<form></form>` si vas a usar el método POST para el envío de datos desde cliente→servidor usando AJAX.
- Añade el atributo **id** a los elementos HTML:
  - `<div>`: para que se pueda acceder mediante JavaScript (usando su id) a su contenido (propiedad `innerHTML`) y actualizarlo usando el valor de tipo string `responseText` que es devuelto por el script PHP. P.e: actualizar el contenido del `<table>` o del `<select>` que pueda contener ese `<div>`.
  - `<input type="text|radio|checkbox|number|password|hidden">` y `<select>`: para que se pueda acceder mediante JavaScript a su valor (propiedad `value` o `checked` para el checkbox) para consultar o modificarlo.
- Añade los **eventos** que necesites para realizar cálculos y actualizar contenidos:
  - `<body>`: `onLoad`
  - `<input type="submit|radio">` y `<button>`: `onClick`
  - `<input type="checkbox">` y `<select>`: `onChange`, `onKeyUp`
  - `<img>`, `<tr>`, `<td>`: `onClick`, `onMouseEnter`, `onMouseOut`, `onMouseOver`