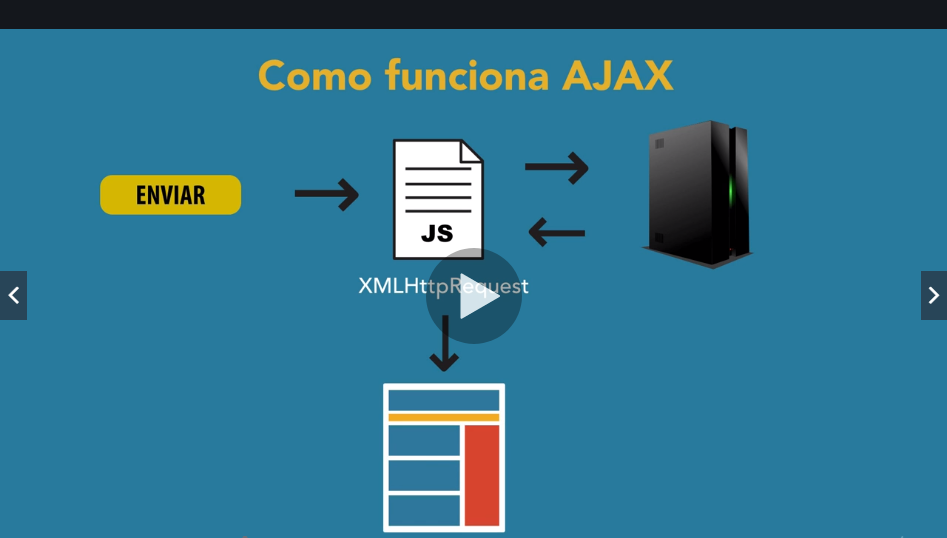
**Ajax**

Ajax es un conjunto de tecnologías que nos permiten realizar consultas al servidor en el background de la página, sin la necesidad de que esta tenga que actualizarse. Se maneja con archivos xml, jason, etc.



Como vemos en la imagen, en un ejemplo de enviar un formulario, el contacto con el servidor es intervenido por el archivo xml, que el cual recibe y envía información. De esta forma no se necesita intervenir la página y de esta forma actualizarse.

XMLHttpRequest

Este es el objeto con el que vamos a intercambiar datos detrás de escena. Se suele crean en el lenguaje JavaScript con el código:

**Var xhr = new XMLHttpRequest();**

**Métodos de XHR**

**New:** Crea un nuevo request.

**.abort():** Cancela el request.

**.open():** Se abre la conexión y le pasas tus parámetros.

**.send():** Envia el request al servidor.

**.getAllResponseHeaders():** Retorna información del header.

**.setRequestHeader:** Agrega valores al header que se desea enviar.

Ejemplo de envio Ajax en un formulario:

**Var xhr = new XMLHttpRequest();**

**xhr.open(“POST** o **GET”, “archivo.php”, “true”);**

**xhr.setRequestHeader(“Content-type”, “application/x-www-form-urlencoded”)**

**xhr.send(“**parametros del form**”);**

**\***Es recomendable si estamos utilizando un archivo con arreglos o datos del database que vienen en un arreglo, convertirlos a **json** asi es mas fácil de leer (con un **echo json\_encode($varN)** en el archivo php)

**Post o Get**

**POST** es utilizado para enviar o intercambiar información.

**GET** es utilizado para recibir información.

**Propiedades del XHR**

**readyState:** Estado del xhr request, nos regresa un valor numérico (0- request sin iniciar, 1 – conexión establecida, 2 - request recibida, 3 – procesando el request, 4 – request finalizado y respuesta lista).

**responseText:** Retorna la respuesta com un string.

**responseXML:** Retorna la respuesta como un xml.

**status:** en los response aparece y define el estado del request. Sus números suelen ser (200 – Correcto, 403 – prohibido, 404 – no ecnontrado)

**onreadyStateChange:** Define una función que deberá ser llamada cuando cambien el estado del request, es decir cuando **readyState** cambie. Ejemplo:

**xhr.onreadyStateChange = function() {**

**if(xhr.readyState = 4 && xhr.status == 200) {**

**var divUsuario = document.getElementById(“registro”);**

**divUsuario.innerHTML = xhr.responseText;**

**}**

**}**

Se realiza un cambio en el html si se realiza la request de forma correcta.

**Ejemplos**

Archivo HTML de los dos:

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title>Ajax</title>

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css" />

        <link rel="stylesheet" href="css/estilos.css" />

    </head>

    <body>

        <div class="contenedor">

            <h1>Aprendiendo AJAX</h1>

            <button id="cargar" value="cargar">Cargar</button>

            <div id="contenido"></div>

        </div>

        <!-- <script src="js/05.js"></script> -->

        <script src="js/06.js"></script>

    </body>

</html>

\*05.js es el text y 06.js es el php. Se comentara cada uno dependiendo el que usamos

**Ejemplo Con archivo Text**

Archivo Javascript:

// pasos para la creación de un request

var btnCargar = document.getElementById("cargar");

function cargarContenido() {

    // crearlo

    var xhr = new XMLHttpRequest();

    // abrirlo

    xhr.open("GET", "texto\_prueba.txt", true);

    // revisar que cambie

    xhr.onreadystatechange = function () {

        //console.log(xhr.readyState);

        if (xhr.readyState == 4 && xhr.status == 200) {

            //console.log("Se cargo correctamente");

            var contenido = document.getElementById("contenido");

            contenido.innerHTML = xhr.responseText;

        }

    };

    xhr.send();

}

btnCargar.addEventListener("click", cargarContenido);

* El archivo text solo tiene simple texto.

**Ejemplo Con archivo PHP**

Archivo PHP:

<?php

    $tecnologias = array(

        "frontend" => "html5, css3, javascript",

        "backend" => "php, mysql, apache",

        "fullstack" => "html5, css3, javascript, php, mysql, apache"

    );

    echo json\_encode($tecnologias);

?>

Archivo Javascript:

// pasos para la creación de un request

var btnCargar = document.getElementById("cargar");

function cargarContenido() {

    // crearlo

    var xhr = new XMLHttpRequest();

    // abrirlo

    xhr.open("GET", "servidor.php", true);

    // revisar que cambie

    xhr.onreadystatechange = function () {

        //console.log(xhr.readyState);

        if (xhr.readyState == 4 && xhr.status == 200) {

            //console.log("Se cargo correctamente");

            var json = JSON.parse(xhr.responseText);

            var contenido = document.getElementById("contenido");

            contenido.innerHTML = json.fullstack;

        }

    };

    xhr.send();

}

btnCargar.addEventListener("click", cargarContenido);