Simulacro Examen 2º DAW

1. **Explica la diferencia entre una petición GET y una petición POST. Proporciona un ejemplo de código para cada uno de ellos en PHP.**

GET: Los datos se envían en la URL, se utiliza para recuperar información y no debería cambiar el estado del servidor.
POST: Los datos se envían en el cuerpo de la solicitud, es adecuado para enviar grandes cantidades de datos o información sensible y se utiliza para cambiar el estado del servidor.

GET

```
<?php
// Verificar si se ha enviado la petición GET
if (isset($_GET['nombre'])) {

    $nombre = $_GET['nombre'];
    echo "Hola, . $nombre” ;
}
?>
<!-- Formulario que envía datos mediante GET -->
<form method="GET" action="">
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre">
    <input type="submit" value="Enviar">
</form>
```

POST

```
<?php
// Verificar si se ha enviado la petición POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nombre = $_POST['nombre'];
    echo "Hola, . $nombre” ;
}
?>
<!-- Formulario que envía datos mediante POST -->
<form method="POST" action="">
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre">
    <input type="submit" value="Enviar">
</form>
```

2. **Escribe una función en PHP que reciba un array de números y devuelva el número mayor. Explícalo paso a paso.**

```php
<?php
function numeroMayor($numeros) {

   // Inicializamos la variable $mayor
    $mayor = 0;

    // Recorremos el array para comparar cada valor
    foreach ($numeros as $numero) {

        // Si encontramos un número mayor que el valor actual de $mayor, lo actualizamos
        if ($numero > $mayor) {
            $mayor = $numero;
        }
    }

    // Retornamos el número mayor encontrado
    return $mayor;
}

// Ejemplo de uso
$numeros = [4, 8, 15, 23, 42, 16];
echo "El número mayor es: " . numeroMayor($numeros);
?>
```

3. **Escribe una función en JavaScript que reciba un número como parámetro y devuelva "Par" si el número es par o "Impar" si el número es impar. Usa la función dentro de un script que muestre el resultado en la consola del navegador.**

```javascript
// Definimos la función para verificar si el número es par o impar
function esPar (numero) {

    // Usamos el operador módulo (%) para verificar si el número es divisible por 2
    if (numero % 2 === 0) {
        return "Par";
    } else {
        return "Impar";
    }
}

// Ejemplo de uso
let numero = 7;
let resultado = esPar (numero);

// Mostramos el resultado en la consola
console.log("El número " + numero + " es: " + resultado);
```

4. **Escribe una consulta SQL que devuelva el nombre y apellidos de los clientes que hayan realizado compras superiores a 1000€ en el último mes, ordenados por fecha de compra.**

```
SELECT nombre, apellidos, fecha_compra, cantidad
FROM clientes
WHERE cantidad > 1000
AND fecha_compra >= CURDATE() - 30
ORDER BY fecha_compra DESC;
```

5. **Define qué es el modelo de caja en CSS y qué propiedades lo componen.**

Es un concepto fundamental que describe cómo los elementos HTML se representan en una página web como cajas rectangulares. Cada elemento se trata como una caja con diferentes capas que controlan su tamaño y su apariencia. Estas cajas permiten determinar cómo se muestran y organizan los elementos en relación con otros.

- **Contenido (content):** Es el área donde se muestra el contenido.
- **Relleno (padding):** Es el espacio entre el contenido y el borde (border) de la caja.
- **Borde (border):** Es el contorno que rodea el contenido y el relleno.
- **Margen (margin):** Es el espacio exterior entre el borde de la caja y otros elementos.

Propiedades que componen el modelo de caja

- **width y height**: Definen el ancho y alto del contenido.
- **padding**: Controla el espacio interior entre el contenido y el borde. Puede tener valores individuales (ej: ***padding-top***, ***padding-right***, etc.)
- **border**: Define el grosor, estilo y color del borde. También se puede desglosar en ***border-width, border-style, border-color***.
- **margin**: Define el espacio exterior alrededor del borde. También se puede desglosar en ***margin-top, margin-right***, etc.

6. **Escribe un código HTML básico para una página web que incluya un formulario con los campos: nombre, correo electrónico y mensaje. Incluye también validación en JavaScript que asegure que todos los campos están llenos antes de enviar.**

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Formulario de Contacto</title>
</head>
<body>

    <h1>Formulario de Contacto</h1>
    <!-- Llamada a la función validarFormulario() al enviar el formulario -->
    <form id="contactForm" onsubmit="return validarFormulario()">
        <label for="nombre">Nombre:</label>
        <input type="text" id="nombre" name="nombre" placeholder="Tu nombre" required>
        <br><br>

        <label for="email">Correo Electrónico:</label>
        <input type="email" id="email" name="email" placeholder="Tu correo electrónico"
      required>
       <br><br>

        <label for="mensaje">Mensaje:</label>
        <textarea id="mensaje" name="mensaje" rows="5" placeholder="Escribe tu mensaje"
      required></textarea>
       <br><br>

        <button type="submit">Enviar</button>
    </form>

    <script>
        function validarFormulario() {
            // Obtener los valores de los campos
            var nombre = document.getElementById("nombre").value;
            var email = document.getElementById("email").value;
            var mensaje = document.getElementById("mensaje").value;

            // Verificar que los campos no estén vacíos
            if (nombre === "" || email === "" || mensaje === "") {
                alert("Por favor, completa todos los campos.");
                return false;  // Evita el envío del formulario
            }

            return true;
        }
    </script>

</body>
</html>
```

7. **Explica qué es AJAX y describe un caso práctico en el que sea útil implementarlo en una aplicación web.**

AJAX (Asynchronous JavaScript and XML) es una técnica que permite a las aplicaciones web enviar y recibir datos de un servidor de forma asíncrona, sin necesidad de recargar o refrescar toda la página.

Aunque el nombre incluye XML, AJAX no se limita a este formato; los datos pueden ser enviados y recibidos en otros formatos, como JSON, HTML o texto plano.

HTML (archivo index.html):

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo Simple de AJAX</title>
</head>
<body>

    <h1>Contenido dinámico con AJAX</h1>
    <p id="contenido">Este es el contenido original.</p>
    <button onclick="cargarContenido()">Cargar nuevo contenido</button>

    <script src="ajax.js"></script>
</body>
</html>
```

JavaScript (archivo ajax.js):

```
function cargarContenido() {

    var xhr = new XMLHttpRequest();
    xhr.open("GET", "contenido.txt", true);  // Suponiendo que hay un archivo llamado contenido.txt
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) // Cambiar el contenido del párrafo
        {
            document.getElementById('contenido').innerHTML = xhr.responseText;
        }
    };
    xhr.send();
}
```

Archivo de texto (archivo contenido.txt):

```
¡Este es el nuevo contenido cargado a través de AJAX!
```

Conclusión

Cuando el usuario hace clic en el botón "Cargar nuevo contenido", el contenido del párrafo cambia a "¡Este es el nuevo contenido cargado a través de AJAX!" sin necesidad de recargar la página.

8. Diferencia entre "frontend" y "backend" en el desarrollo web, mencionando dos tecnologías asociadas a cada uno.

Frontend

Es la parte de una aplicación web que interactúa directamente con el usuario. Incluye todo lo que el usuario ve y con lo que puede interactuar, como el diseño, los gráficos, los botones y otros elementos visuales. El frontend se encarga de la presentación y la experiencia del usuario.

Tecnologías asociadas

HTML: Es el lenguaje de marcado utilizado para estructurar el contenido en la web.

CSS: Se utiliza para diseñar y estilizar la presentación de las páginas web, controlando el diseño, los colores, las fuentes y otros aspectos visuales.

JavaScript: Es un lenguaje de programación que permite añadir interactividad a las páginas web, como animaciones, formularios dinámicos y la manipulación del DOM.

Backend

Es la parte de una aplicación web que no es visible para el usuario y se encarga de la lógica del servidor, la gestión de la base de datos y la comunicación entre el frontend y el servidor. Es responsable de manejar la autenticación, la lógica empresarial y el almacenamiento de datos.

Tecnologías asociadas

Node.js: Un entorno de ejecución para JavaScript del lado del servidor que permite crear aplicaciones web escalables y rápidas.

PHP: Un lenguaje de programación del lado del servidor ampliamente utilizado para el desarrollo web, especialmente en la creación de aplicaciones dinámicas y gestión de contenido.

Python: Un lenguaje de programación versátil que se utiliza a menudo en el desarrollo web con frameworks como Django o Flask.

Conclusión

El desarrollo web se compone de dos áreas complementarias: el *frontend*, que se enfoca en la interfaz de usuario y la experiencia, y el *backend*, que gestiona la lógica, el almacenamiento de datos y la comunicación entre el servidor y el cliente. Ambos son esenciales para construir aplicaciones web funcionales y efectivas.

9. **Explica qué es el DOM (Document Object Model) y cómo se puede modificar con JavaScript. Proporciona un ejemplo de cómo cambiar el contenido de un elemento HTML utilizando JavaScript.**

Es una representación estructurada de un documento HTML o XML. Es un modelo de objetos que permite a los desarrolladores acceder y manipular el contenido, la estructura y el estilo de un documento web de forma dinámica mediante programación.

JavaScript proporciona varios métodos para acceder y manipular el DOM. Algunos de los métodos más comunes incluyen:

- ***document.getElementById():*** Selecciona un elemento por su ID.
- ***document.getElementsByClassName():*** Selecciona elementos por su clase.
- **document.getElementsByTagName():** Selecciona elementos por su etiqueta.
- **document.querySelector():** Selecciona el primer elemento que coincide con un selector CSS.
- **element.innerHTML:** Permite cambiar el contenido HTML de un elemento.
- **element.style:** Permite modificar los estilos CSS de un elemento.

HTML (archivo index.html):

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo de DOM con JavaScript</title>
</head>
<body>

    <h2 id="titulo">Título Original</h2>
    <button onclick="cambiarContenido()">Cambiar Título</button>

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

</body>
</html>
```

JavaScript (archivo script.js):

```
function cambiarContenido() {
    // Seleccionamos el elemento con id "titulo"
    var titulo = document.getElementById("titulo");

    // Cambiamos el contenido del elemento
    titulo.innerHTML = "Título Modificado";
}
```

10. Una aplicación web con PHP, HTML y CSS que permite a los usuarios registrarse mediante un formulario, almacenar los datos en un archivo y mostrar una lista de usuarios registrados

Crear el archivo vacio usuarios.txt

Archivo index.php

```
<?php
// Leer usuarios desde el archivo
$usuarios = file('usuarios.txt', FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Registro de Usuarios</title>
    <link rel="stylesheet" href="estilos.css">
</head>
<body>

    <h1>Registro de Usuarios</h1>

    <form action="registro.php" method="POST">
        <input type="text" name="nombre" placeholder="Nombre" required>
        <input type="email" name="correo" placeholder="Correo Electrónico" required>
        <input type="password" name="contrasena" placeholder="Contraseña" required>
        <button type="submit">Registrarse</button>
    </form>

    <h2>Usuarios Registrados</h2>
    <ul>
        <?php foreach ($usuarios as $usuario): ?>
            <li><?php echo $usuario; ?></li>
        <?php endforeach; ?>
    </ul>

</body>
</html>
```

Archivo registro.php

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $nombre = $_POST['nombre'];
    $correo = $_POST['correo'];
    $contrasena = $_POST['contrasena'];

    // Almacenar los datos en el archivo
    $usuario = "$nombre | $correo | $contrasena\n";
    file_put_contents('usuarios.txt', $usuario, FILE_APPEND | LOCK_EX);

    // Redireccion a la pagina principal
    header('Location: index.php');
    exit;
}
?>
```

Archivo estilos.css

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
    background-color: white;
}

h1 {
    color: #333;
}

form {
    margin-bottom: 20px;
}

input {
    display: block;
    margin-bottom: 10px;
    padding: 10px;
    width: 300px;
}

button {
    padding: 10px 15px;
    background-color: blue
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: blue;
}

h2 {
    margin-top: 20px;
}

ul {
    list-style-type: none;
    padding: 0;
}
```