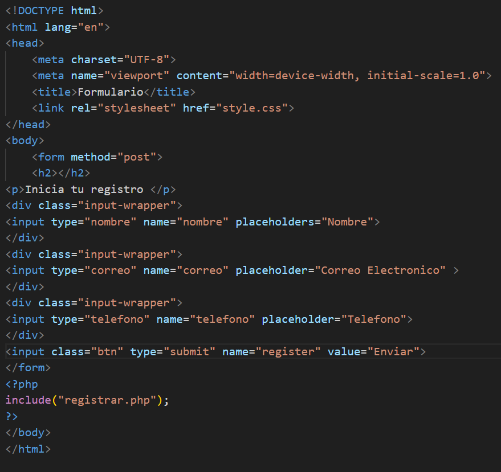
Documentación del código registro programador 2

Diseño de formulario



Código HTML (index.html):

1. `**<!DOCTYPE html>**`: Declaración doctype que define el tipo de documento como HTML5.

2. `**<html lang="en">**`: La etiqueta raíz que envuelve todo el contenido HTML, con el atributo `lang` especificado como "en" (inglés).

3. `**<head>**`: La sección de encabezado del documento HTML que contiene metadatos, vínculos a hojas de estilo y otros recursos.

4. `**<meta charset="UTF-8">**`: Especifica la codificación de caracteres UTF-8 para el documento.

5. **`<meta name="viewport" content="width=device-width, initial-scale=1.0">**`: Configura la escala inicial y el ancho del dispositivo para una visualización adecuada en dispositivos móviles.

6. **`<title>Formulario</title>`:** Define el título de la página que se muestra en la pestaña del navegador.

7. **`<link rel="stylesheet" href="style.css**">`: Enlaza la hoja de estilo externa "style.css" al documento HTML.

8. **`<body>`:** La etiqueta que contiene el contenido visible de la página.

9. **`<form method="post">`:** Inicio del formulario que utiliza el método POST para enviar datos al servidor.

10. **`<h2></h2>`:** Encabezado del formulario, que está vacío en el código proporcionado.

11. **`<p>Inicia tu registro </p>`:** Un párrafo que proporciona una breve descripción del formulario.

12. **`.input-wrapper`:** Un contenedor para cada campo de entrada en el formulario.

13. **`<input>`:** Elementos de entrada de texto para el nombre, correo y teléfono. Cada uno tiene un atributo `name` que se usa para identificar los datos en el servidor y un atributo `placeholder` que proporciona una pista sobre lo que se espera en el campo.

14. **`<input class="btn" type="submit" name="register" value="Enviar">`:** Un botón de envío que permite al usuario enviar el formulario. Tiene una clase CSS llamada "btn" y se identifica con el nombre "register" en el servidor.

15. **`<?php include("registrar.php"); ?>`:** Una instrucción PHP que incluye el archivo "registrar.php", que probablemente contiene la lógica de procesamiento del formulario.

CSS (style.css):

1. **`body`:** Establece varias propiedades de estilo para el cuerpo de la página, como la fuente, alineación y altura mínima de la vista.

2. `**form`:** Estiliza el formulario en sí, con sombra, relleno, bordes redondeados y alineación.

3. **`form h2`:** Estiliza el encabezado del formulario, estableciendo tamaño de fuente, color y margen.

4. **`form p`:** Estiliza el párrafo que proporciona una descripción del formulario, configurando el tamaño de fuente, color y margen inferior.

5. `.**input-wrapper`:** Establece estilos generales para los contenedores de campos de entrada, incluyendo el color, ancho y margen inferior.

6. `**input`:** Estiliza los campos de entrada de texto, especificando color, relleno, ancho, bordes redondeados, sombra, y tamaño de fuente.

7. `.**input-icon`:** Estiliza un posible icono dentro de los campos de entrada (aunque no está presente en el código HTML proporcionado). Define su posición y tamaño.

8. **`.btn`:** Estiliza el botón de envío, estableciendo propiedades de color, relleno, fondo, sombra y tamaño de fuente. También cambia el cursor al puntero cuando se pasa el ratón por encima y define un estilo de fondo diferente para el estado de hover (sobre).

Conexión (conexion.php):

1**. `<?php` y `?>`:** Estas etiquetas encierran el código PHP y señalan el inicio y el final de un bloque de código PHP en un archivo HTML o PHP.

2. **`$conex`:** Esta línea declara una variable llamada `$conex` que se utilizará para almacenar la conexión a la base de datos.

3. `mysqli\_connect**("localhost", "root", "", "registro")`:** Esta es la llamada a la función `mysqli\_connect()` que establece la conexión a la base de datos MySQL. Aquí está lo que significa cada argumento:

- **`"localhost"`:** Es el nombre del servidor de la base de datos. En este caso, se utiliza "localhost" para indicar que la base de datos está en la misma máquina que el código PHP. Si la base de datos estuviera en un servidor remoto, aquí se proporcionaría la dirección IP o el nombre del servidor remoto.

- **`"root"`:** Es el nombre de usuario de la base de datos. En este caso, se utiliza "root", que es el usuario por defecto de MySQL. En un entorno de producción, es una buena práctica utilizar un usuario con permisos específicos en lugar del usuario "root" por razones de seguridad.

- **`""`:** Aquí se encuentra la contraseña del usuario de la base de datos. En este caso, se deja en blanco, lo que significa que no se proporciona una contraseña. En un entorno de producción, se debe utilizar una contraseña segura.

- **`"registro"`:** Este es el nombre de la base de datos a la que se desea conectar. En este caso, se especifica "registro" como el nombre de la base de datos a la que se conectará.

La variable **`$conex`** almacenará la conexión a la base de datos y se utilizará posteriormente en el código PHP para realizar consultas y operaciones en la base de datos.

Registrar (Registrar.php):

Este código PHP se encarga de procesar un formulario de registro y guardar los datos enviados en una base de datos. Aquí está la documentación de las partes importantes de este código:

1. **`<?php` y `?>`:** Estas etiquetas encierran el código PHP y señalan el inicio y el final de un bloque de código PHP en un archivo HTML o PHP.

2. **`include("conexion.php");`:** Esta línea incluye el archivo "conexion.php" en el que se establece la conexión a la base de datos. Esto permite que este archivo tenga acceso a la variable de conexión `$conex` que se utiliza más adelante en el código.

3. `**if (isset($\_POST['register'])) {`:** Esta línea verifica si el formulario se ha enviado y si el campo "register" (el botón de envío) está presente en los datos POST. Esto indica que se ha enviado el formulario.

4. `**if (strlen($\_POST['nombre']) >= 1 && strlen($\_POST['telefono']) >= 1 && strlen($\_POST['correo']) >= 1) {`:** Este bloque de código verifica si los campos del formulario (nombre, teléfono y correo) tienen al menos una longitud de 1 caracter. Si todos los campos tienen datos, se procede a procesar el formulario.

5. **`$nombre = trim($\_POST['nombre']);`, `$telefono = trim($\_POST['telefono']);`, `$correo = trim($\_POST['correo']);`:** Estas líneas de código almacenan los datos enviados en variables y utilizan la función `trim()` para eliminar espacios en blanco innecesarios alrededor de los datos.

6. **`$consulta = "INSERT INTO datos(nombre, telefono, correo) VALUES('$nombre', '$telefono', '$correo')";`:** Se crea una consulta SQL para insertar los datos en la base de datos. Los valores de las variables `$nombre`, `$telefono`, y `$correo` se insertan en la tabla "datos" en las columnas correspondientes.

7. **`$resultado = mysqli\_query($conex, $consulta);`:** Se ejecuta la consulta SQL utilizando la función `mysqli\_query()`, que toma la conexión a la base de datos `$conex` y la consulta SQL `$consulta`. El resultado de la consulta se almacena en la variable `$resultado`.

8. El código también maneja el caso en el que el formulario se envía sin datos (los campos están vacíos), pero el manejo de errores en este caso es similar al caso de fallo en la inserción.