Conexión entre MySQL y PHP

1. Requisitos Previos:

Tener instalado un servidor web (como Apache).

Tener PHP y MySQL instalados.

2. Extensiones de PHP para MySQL:

PHP ofrece varias extensiones para interactuar con MySQL, siendo las más comunes mysqli (MySQL Improved) y PDO (PHP Data Objects).

3. Conexión usando mysqli:

La función mysqli_connect() se usa para establecer la conexión.

Ejemplo:

```
$conexion = mysqli_connect("localhost", "usuario", "contraseña", "basededatos");
if (!$conexion) {
    die("Conexión fallida: " . mysqli_connect_error());
}
```

4. Conexión usando PDO:

Proporciona una interfaz más flexible y segura.

Ejemplo:

```
try {
    $conexion = new PDO("mysql:host=localhost;dbname=basededatos", "usuario", "contraseña");
    $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Conexión fallida: " . $e->getMessage();
}
```

5. Ejecución de Consultas:

Para ejecutar consultas, se pueden usar funciones como mysqli_query() o métodos de PDO como prepare() y execute().

Ejemplo con mysqli:

```
$resultado = mysqli_query($conexion, "SELECT * FROM tabla");
while ($fila = mysqli_fetch_assoc($resultado)) {
    echo $fila['columna'];
}
```

Ejemplo con PDO:

```
$stmt = $conexion->prepare("SELECT * FROM tabla");
$stmt->execute();
$resultados = $stmt→fetchAll();
```

6. Cierre de Conexión:

Para mysqli, se utiliza mysqli_close($conexion).

Para PDO, no es necesario, ya que el objeto se destruye automáticamente.

7. Buenas Prácticas:

Usar declaraciones preparadas para prevenir inyecciones SQL.

Manejar errores adecuadamente para facilitar la depuración.