

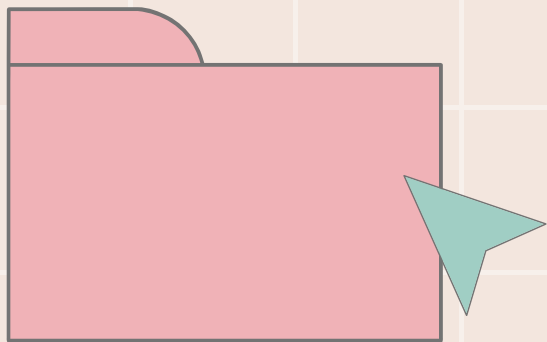


進階網站開發

PHP、SQL



<https://github.com/if-else-master/Login-System-and-Key-SQL-Commands-Tutorial.git>

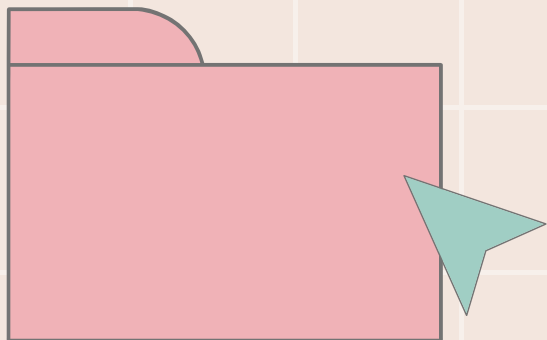


```
<?php  
echo "Hello, World!";  
?>
```

PHP介紹

PHP 是一種伺服器端的腳本語言，全名為 Hypertext Preprocessor。它主要用於開發動態網頁和 Web 應用程式。以下是 PHP 的幾個關鍵特點：

1. 易於學習：語法簡單，適合初學者快速上手。
2. 動態內容生成：可與 HTML 結合，生成動態網頁內容。
3. 跨平台：可在 Windows、Linux 和 macOS 等多種平台上運行。
4. 資料庫整合：支持多種資料庫（如 MySQL、PostgreSQL）。
5. 開源與免費：PHP 是開源的，任何人都可以自由下載和使用。
6. 應用廣泛：常用於部落格、電子商務網站及社群網路平台的後端開發。



```
SELECT 列名 FROM 表名 WHERE 條件;  
  
INSERT INTO 表名 (列名1, 列名2) VALUES (值1, 值2);  
  
UPDATE 表名 SET 列名 = 新值 WHERE 條件;  
  
DELETE FROM 表名 WHERE 條件;
```


SQL介紹

SQL (Structured Query Language) 是用於管理和操作關聯式資料庫的標準語言。主要功能包括：

1. 資料查詢：從資料庫中檢索數據。
2. 資料操作：插入、更新和刪除數據。
3. 資料結構定義：建立或修改資料庫結構（表、索引）。
4. 資料控制：控制對數據的存取權限。
- 5.



PHP

DEVSENSE 

All-in-One PHP

Disable 

U



PHP Intelephense

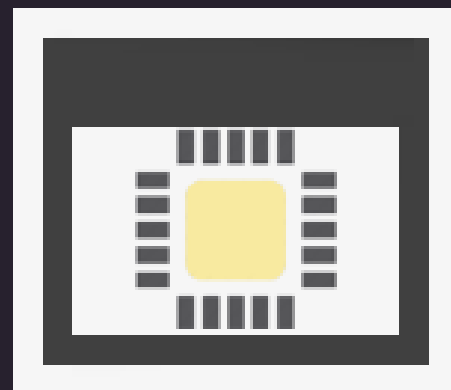
Ben Mewburn  intelephense.com

PHP code intelligence for Visual Stu


Disable 

Uninstall 

Switch to P



PHP Profiler

DEVSENSE  devsense.com

Support for PHP (Xdebug)

Disable 

Uninstall 

S



PHP Server

brapifra |  2,348,068

Serve your Project with PHP

Disable 

Uninstall 



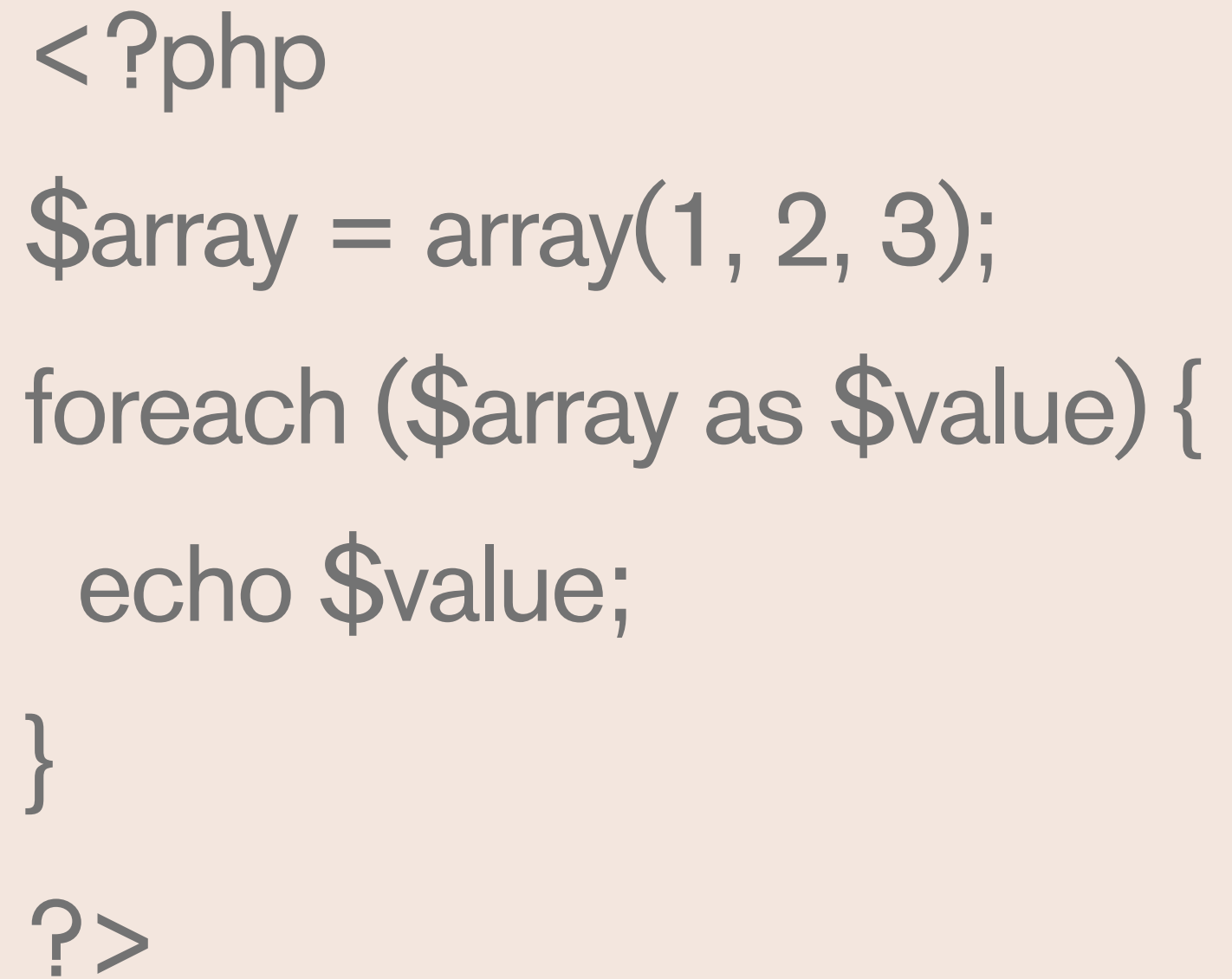
PHP：變數

```
<?php
$integer = 10;
$float = 10.5;
$string = "Hello, World!";
$boolean = true;
$array = array(1, 2, 3);
?>
```

PHP：for迴圈

```
<?php
for ($i = 0; $i < 10; $i++) {
    echo $i;
}
?>
```

PHP：foreach 迴圈



```
<?php  
$array = array(1, 2, 3);  
foreach ($array as $value) {  
    echo $value;  
}  
?>
```

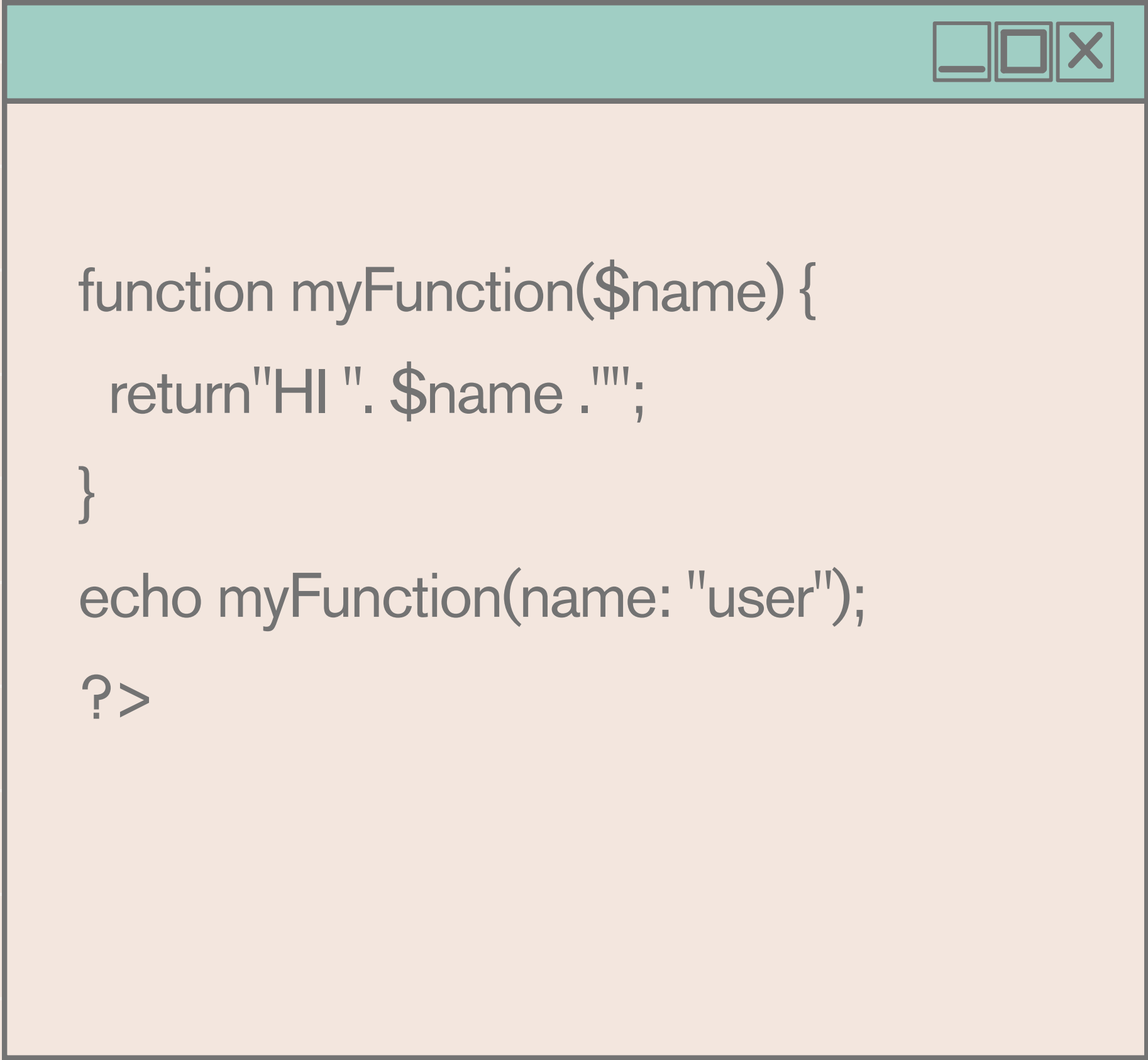
PHP : if-else

```
$x = 10;  
if ($x > 5) {  
    echo "x is greater than 5";  
} elseif ($x == 5) {  
    echo "x is equal to 5";  
} else {  
    echo "x is less than 5";  
}
```


PHP：陣列


```
<?php
$array = array(1, 2, 3, 4, 5);
for ($i=0; $i < 5 ; $i++) {
    echo $array[$i];
}
?>
```

PHP : Function



```
function myFunction($name) {  
    return "Hi ". $name ."";  
}  
echo myFunction(name: "user");  
?>
```

範例程式碼



```
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// 檢查連接  
if ($conn->connect_error) {  
    die("連接失敗: " . $conn->connect_error);  
}
```

連接 MYSQL

```
<?php
$host = "mysql00.clmamy4i8bzn.ap-northeast-1.rds.amazonaws.com";
$username = "root";
$password = "";
$database = "";

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect_error) {
    die("連接失敗: " . $conn->connect_error);
}
?>
```

搜尋

```
$sql = "SELECT * FROM web00";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // 輸出每行數據
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["username"]. "<br>";
    }
} else {
    echo "0 結果<br>";
}
```

新增

```
$sql = "INSERT INTO web00 (username) VALUES ('user2')";  
if ($conn->query($sql) === TRUE) {  
    echo "新記錄創建成功<br>";  
} else {  
    echo "錯誤: " . $sql . "<br>" . $conn->error;  
}  
}
```

修改

```
$sql = "UPDATE web00 SET username='newuser' WHERE id=2";  
if ($conn->query($sql) === TRUE) {  
    echo "記錄更新成功<br>";  
} else {  
    echo "錯誤: " . $sql . "<br>" . $conn->error;  
}
```



```
$sql = "DELETE FROM web00 WHERE id=2";  
if ($conn->query($sql) === TRUE) {  
    echo "記錄刪除成功<br>";  
} else {  
    echo "錯誤: " . $sql . "<br>" . $conn->error;  
}
```

刪除

register

```
// 註冊功能
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $new_username = $_POST['username'];
    $new_password = password_hash(password: $_POST['password'], algo: PASSWORD_DEFAULT);
    // 準備 SQL 插入語句，使用預處理語句來防止 SQL 注入
    $stmt = $conn->prepare(query: "INSERT INTO users (username, password) VALUES (?, ?)");
    // 綁定參數到預處理語句，"ss" 表示兩個字符串參數
    $stmt->bind_param(types: "ss", var: &$new_username, vars: &$new_password);
    // 執行預處理語句
    if ($stmt->execute()) {
        // 如果執行成功，輸出成功訊息
        echo "註冊成功";
    } else {
        // 如果執行失敗，輸出失敗訊息和錯誤信息
        echo "註冊失敗: " . $stmt->error;
    }
}
$stmt->close();
}

$conn->close();
?>
```

login

```
// 登入功能
// 檢查請求方法是否為 POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // 從 POST 請求中獲取使用者名稱和密碼
    $login_username = $_POST['username'];
    $login_password = $_POST['password'];
    // 準備 SQL 查詢語句，使用預處理語句來防止 SQL 注入
    $stmt = $conn->prepare(query: "SELECT id, username, password FROM users WHERE username = ?");
    // 綁定使用者名稱參數到預處理語句
    $stmt->bind_param(types: "s", var: &$login_username);
    // 執行預處理語句
    $stmt->execute();
    // 綁定結果變數到預處理語句
    $stmt->bind_result(var: &$id, vars: &$username, $hashed_password);
    // 獲取查詢結果
    $stmt->fetch();
    // 驗證輸入的密碼是否與存儲的哈希密碼匹配
    if (password_verify(password: $login_password, hash: $hashed_password)) {
        // 如果驗證成功，設置會話變數
        $_SESSION['user_id'] = $id;
        $_SESSION['username'] = $username;

        // 重定向到 messages.php 頁面
        header(header: "Location: messages.php");
    } else {
        // 如果驗證失敗，輸出錯誤訊息
        echo "登入失敗：使用者名稱或密碼錯誤";
    }

    // 關閉預處理語句
    $stmt->close();
}
```

message 留言功能

```
// 留言功能
// 檢查請求方法是否為 POST 且是否有設置 'message' 參數
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['message'])) {
    // 從會話中獲取使用者 ID
    $user_id = $_SESSION['user_id'];
    // 從 POST 請求中獲取留言內容
    $message = $_POST['message'];
    // 準備 SQL 插入語句，使用預處理語句來防止 SQL 注入
    $stmt = $conn->prepare(query: "INSERT INTO messages (user_id, message) VALUES (?, ?)");
    // 綁定使用者 ID 和留言內容參數到預處理語句。is 表示'整數'和'字串'
    $stmt->bind_param(types: "is", var: &$user_id, vars: &$message);
    // 執行預處理語句
    if ($stmt->execute()) {
        // 如果執行成功，輸出成功訊息
        echo "留言成功";
    } else {
        // 如果執行失敗，輸出失敗訊息和錯誤信息
        echo "留言失敗: " . $stmt->error;
    }
    // 關閉預處理語句
    $stmt->close();
}
```

message 删除留言

```
// 删除留言功能
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['delete'])) {
    $message_id = $_POST['message_id'];
    $user_id = $_SESSION['user_id'];

    $stmt = $conn->prepare(query: "DELETE FROM messages WHERE id = ? AND user_id = ?");
    $stmt->bind_param(types: "ii", var: &$message_id, vars: &$user_id);

    if ($stmt->execute()) {
        echo "留言删除成功";
    } else {
        echo "留言删除失败: " . $stmt->error;
    }

    $stmt->close();
}
```

message 顯示留言

```
// 顯示留言
// 準備SQL查詢語句，從messages和users表中選取資料
// 選取messages表中的id、users表中的username、messages表中的message和created_at欄位
// 使用JOIN將messages表和users表連接，條件是messages.user_id等於users.id
$stmt = $conn->prepare(query: "SELECT messages.id, users.username, messages.message, messages.created_at FROM messages JOIN users ON messages.user_id = users.id");
// 執行準備好的SQL查詢語句
$stmt->execute();
// 獲取查詢結果
$result = $stmt->get_result();
```

message 修改留言

```
// 獲取留言ID
$message_id = $_GET['id'];
$user_id = $_SESSION['user_id'];
// 獲取留言內容
$stmt = $conn->prepare(query: "SELECT message FROM messages WHERE id = ? AND user_id = ?");
$stmt->bind_param(types: "ii", var: &$message_id, vars: &$user_id);
$stmt->execute();
$stmt->bind_result(var: &$message);
$stmt->fetch();
$stmt->close();
// 修改留言功能
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $new_message = $_POST['new_message'];
    $stmt = $conn->prepare(query: "UPDATE messages SET message = ? WHERE id = ? AND user_id = ?");
    $stmt->bind_param(types: "sii", var: &$new_message, vars: &$message_id, $user_id);
    if ($stmt->execute()) {
        header(header: "Location: messages.php");
    } else {
        echo "...留言修改失敗: " . $stmt->error;
    }
    $stmt->close();
}
```



Thanks



<https://github.com/if-else-master/login-system.git>