



Chapter 12 로그인과 회원 정보 수정 페이지 제작

목차

01 로그인 페이지

02 로그인과 로그아웃 처리 페이지

03 회원 정보 수정 페이지

학습목표

- 로그인 폼 양식의 구성을 이해할 수 있다.
- 로그인 및 로그아웃 처리 방법을 이해할 수 있다.
- 웹 페이지에서 로그인 여부를 판단할 수 있다.
- 회원 정보 수정 폼 양식의 구성을 이해할 수 있다.
- 수정된 회원 정보를 데이터베이스에 업데이트할 수 있다.

- 로그인 페이지를 불러오려면 웹 브라우저 주소 창에 주소 입력

`http://localhost/part3/login_form.php`

- 또는 실습 사이트 메인 페이지 오른쪽 상단의 [로그인(12장)] 메뉴 클릭, 로그인 페이지 호출.

The screenshot shows a web page titled 'PHP 프로그래밍 입문' (Introduction to PHP Programming). The page has a dark header with navigation links: 'HOME', '쪽지 만들기(13장)' (Make a note (Chapter 13)), '게시판 만들기(14장)' (Make a board (Chapter 14)), and '사이트 완성하기(15장)' (Finish the site (Chapter 15)). Below the header is a main content area with a dark background. On the left, it says 'PHP Coding' and 'Server Side Web Programming'. On the right, there is a diagram showing 'PHP', 'HTML', 'CSS', 'JAVASCRIPT', and 'CMS' connected by arrows. In the center, there is a login form titled '로그인' (Login). The form has two input fields: '아이디' (ID) and '비밀번호' (Password), followed by a 'Log In' button. At the bottom of the page, there is a footer with contact information: 'PHP 프로그래밍 입문 | 한빛아카데미' (Introduction to PHP Programming | Hanbit Academy), '예제 소스 다운로드' (Download example source code) with links to '한빛아카데미' (http://hanbit.co.kr) and '코딩스쿨' (http://codingschool.info), and '저자 문의 메일' (Email for author inquiries) with the address 'goldmont@naver.com'.

그림 12-1 로그인 페이지(login_form.php)

- 회원 가입 페이지 만들 때 사용한 아이디와 비밀번호를 로그인 페이지에 입력.
- <Log In> 클릭하면 로그인. 로그인 상태로 사이트 이용.
- 아이디와 비밀번호를 잘못 입력하면 경고 창에 오류 메시지 출력.

표 12-1 로그인 페이지 구성에 필요한 파일

구분	파일	설명
CSS	css/common.css	사이트에 공통으로 사용되는 CSS 파일
	css/login.css	로그인 페이지의 CSS 파일
자바스크립트	js/login.js	로그인 페이지의 자바스크립트 파일
로그인 페이지	login_form.php	로그인 페이지
기능 페이지	login.php	사용자가 입력한 아이디와 비밀번호가 맞으면 세션 변수에 값을 할당하여 로그인 처리
	logout.php	세션 변수를 삭제하여 로그아웃 처리

- **로그인 페이지(login_form.php)** 아이디와 비밀번호를 입력할 수 있는 폼 양식 제공.
- **login.php 파일** 사용자가 입력한 아이디와 비밀번호가 데이터베이스에 존재하는지 확인. 데이터베이스에 존재하면 로그인 처리하고 웹 페이지에 로그인 상태를 알려주기 위해 세션 변수 생성.
- **logout.php 파일** 로그인 상태를 표시하는 세션 변수를 삭제하여 로그아웃 처리.

[예제 12-1] 로그인 페이지 `part3/login_form.php`

```
<!-- 생략 -->
```

```
<link rel="stylesheet" type="text/css" href="/css/common.css">
```

```
<link rel="stylesheet" type="text/css" href="/css/login.css">
```

```
<script type="text/javascript" src="/js/login.js"> </script>
```

①

```
<!-- 생략 -->
```

```
<div id="login_box">
```

```
  <div id="login_title">
```

```
    <span>로그인</span>
```

```
  </div>
```

```
<div id="login_form">
```

```
  <form name="login_form" method="post" action="login.php">
```

②

[예제 12-1] 로그인 페이지 `part3/login_form.php`

```
<ul>
  <li> <input type="text" name="id" placeholder="아이디"> </li> ③
  <li> <input type="password" id="pass" name="pass"
    placeholder="비밀번호"> </li>
</ul>
<div id="login_btn">
  <a href="#">  </a>
</div>
</form>
</div> <!-- login_form -->
</div> <!-- login_box -->
<!-- 생략 -->
```


[예제 12-1] 로그인 페이지 `part3/login_form.php`

- ① 자바스크립트 파일(login.js) 불러오기 자바스크립트 파일인 login.js를 불러옴.
- 이 자바스크립트 파일은 로그인 페이지의 아이디와 비밀번호 입력 창에 데이터가 있는지 검사 후 경고 창 출력. 데이터가 제대로 입력되었다면 login.php(②에서 설정)로 이동.

[예제 12-1] 로그인 페이지 `part3/login_form.php`

- 로그인 페이지의 자바스크립트 `part3/js/login.js`

```
function check_input()
{
    if (!document.login_form.id.value)
    {
        alert("아이디를 입력하세요");
        document.login_form.id.focus();
        return;
    }
    if (!document.login_form.pass.value)
    {
        alert("비밀번호를 입력하세요");
        document.login_form.pass.focus();
        return;
    }
    document.login_form.submit();
}
```

[예제 12-1] 로그인 페이지 `part3/login_form.php`

- login.js 파일에는 check_input() 함수 정의.
- check_input() 함수: 로그인 페이지의 입력 창에 아이디와 비밀번호를 입력하고 <Log In> 버튼(④의 login.png)을 클릭하면 동작. 아이디 입력 창에 데이터가 입력되었는지 검사하고, 비어 있는 상태에서 <Log In> 버튼을 클릭하면 경고 창 띄움.



그림 12-2 아이디를 입력하지 않고 <Log In>을 클릭하면 나타나는 경고 창

- 아이디와 비밀번호가 모두 입력되었다면 document.login_form.submit()에 의해 ②의 action 속성에 설정된 login.php 파일로 페이지 이동.

[예제 12-1] 로그인 페이지 `part3/login_form.php`

② 폼 문 <form> 태그의 name 속성을 login_form, action 속성을 login.php로 설정.

③ 아이디와 비밀번호 입력 창 name 속성을 각각 id와 pass로 설정.

④ 로그인 버튼

- 버튼을 클릭하면 자바스크립트의 onclick 이벤트 핸들러에 설정된 check_input() 함수 실행.
- Check_input() 함수는 아이디와 비밀번호 입력 창에 데이터가 입력되었는지 검사.
- 데이터가 입력되었으면 ②에서 <form> 문의 action 속성에 설정된 login.php로 페이지 이동.

1. 로그인 처리

- 로그인 페이지 입력 창에 아이디와 비밀번호를 입력하고 <Log In>을 클릭하면 [예제 12-1]에서 ②의 action 속성에 설정된 login.php 파일로 페이지 이동.
- login.php 파일은 사용자가 입력한 아이디와 비밀번호가 11장에서 생성한 members 테이블에 존재하는지 검사하여 로그인 가능 여부를 판단하고 그에 따라 처리.

02 로그인과 로그아웃 처리 페이지

IT COOKBOOK 250

[예제 12-2] 로그인 처리 페이지 `part3/login.php`

```
<?php
    $id  = $_POST["id"];
    $pass = $_POST["pass"];

    $con = mysqli_connect("localhost", "user1", "12345", "sample");
    $sql = "select * from members where id='$id'";
    $result = mysqli_query($con, $sql);

    $num_match = mysqli_num_rows($result);

    if (!$num_match)
    {
        echo(
            <script>
                window.alert('등록되지 않은 아이디입니다!')
                history.go(-1)
            </script>
        );
    }
```

[예제 12-2] 로그인 처리 페이지 `part3/login.php`

```
}  
else  
{  
    $row = mysqli_fetch_array($result);  
    $db_pass = $row["pass"];  
  
    mysqli_close($con);  
  
    if ($pass != $db_pass)  
    {  
        echo("  
            <script>  
                window.alert('비밀번호가 틀립니다!')  
                history.go(-1)  
            </script>  
        ");  
        exit;
```

[예제 12-2] 로그인 처리 페이지 `part3/login.php`

```
}  
else  
{  
    session_start();  
    $_SESSION["userid"] = $row["id"];  
    $_SESSION["username"] = $row["name"];  
    $_SESSION["userlevel"] = $row["level"];  
    $_SESSION["userpoint"] = $row["point"];  
  
    echo("  
        <script> location.href = 'index.php';  
        </script>  
    ");  
}  
}  
?>
```


[예제 12-2] 로그인 처리 페이지 `part3/login.php`

① 아이디와 비밀번호 전달받기

로그인 페이지에서 입력한 아이디와 비밀번호를 POST 방식으로 전달받아 각각 \$id와 \$pass에 저장.

② MySQL 서버 접속

mysqli_connect() 함수로 MySQL 서버에 접속.

③ 아이디 검색

입력한 아이디인 \$id를 mysqli_query() 함수로 members 테이블에서 검색, 결과를 \$result에 저장.

④ 레코드 개수 세기

mysqli_num_rows() 함수로 ③의 검색 결과로 저장된 레코드 개수를 세고 \$num_match에 저장.

⑤ 아이디 존재 여부 판단

- \$num_match를 확인하여 사용자가 입력한 아이디가 존재하는지 검사.
- \$num_match가 0이면 입력한 아이디가 members 테이블에 존재하지 않고, \$num_match가 값을 가지면 입력한 아이디가 members 테이블에 존재.

[예제 12-2] 로그인 처리 페이지 `part3/login.php`

⑥ 아이디가 존재하지 않는 경우

\$num_match가 0이면 ⑤의 if문 조건식이 참이므로 '등록되지 않은 아이디입니다!'라는 경고 창 출력.

⑦ 아이디가 존재하는 경우

\$num_match가 값을 가지면 ⑤의 if문 조건식이 거짓이므로 else 문장 수행.

⑧ DB에서 비밀번호 가져오기

mysqli_fetch_array() 함수로 해당 레코드를 가져와 \$row에 저장한 뒤 \$row의 pass 필드에 있는 데이터, 즉 DB에 저장된 비밀번호인 \$row["pass"]를 \$db_pass에 저장.

⑨ 입력된 비밀번호와 DB의 비밀번호 비교

사용자가 입력한 비밀번호인 \$pass와 DB에서 가져온 비밀번호인 \$db_pass 비교.

⑩ 비밀번호가 다른 경우

입력 비밀번호와 DB 비밀번호가 다르면 '비밀번호가 틀립니다!'라는 경고 창 띄우고 다시 로그인 페이지로 돌아감.

[예제 12-2] 로그인 처리 페이지 `part3/login.php`

⑪ 비밀번호가 같은 경우

- 입력된 비밀번호와 DB의 비밀번호가 같으면 `session_start()`로 세션 시작.
- DB에 저장된 아이디인 `$row["id"]` 값으로 세션 변수인 `$_SESSION["userid"]` 생성.
- 같은 방법으로 세션 변수 `$_SESSION["username"]`, `$_SESSION["userlevel"]`, `$_SESSION["userpoint"]` 생성.
- 접속한 웹 페이지에서 사용자가 로그인 상태인지를 알려면 `$_SESSION["userid"]` 값 조사하면 됨.
- `$_SESSION["username"]`, `$_SESSION["userlevel"]`, `$_SESSION["userpoint"]`는 로그인한 사용자 이름, 레벨, 포인트 의미.

⑫ 메인 페이지로 이동

세션 생성하여 로그인이 완료되면 메인 페이지(index.php)로 이동.

2. 로그인 상태 판단

- 웹 사이트에서는 사용자가 로그인한 상태인지 판단하여 웹 페이지 구성과 동작 방법을 달리 하는 경우가 종종 있음.
- 로그인 상태를 판단하는 방법은 [예제 12-2](login.php)의 ⑪에서 생성한 세션 변수가 존재하는지 조사하는 것.
- 실습 사이트의 각 웹 페이지에서 로그인 상태를 검사하기 위해 상단 헤더 페이지인 header.php에 세션 관련 코드를 삽입.

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

```
<?php
    session_start();
    if (isset($_SESSION["userid"])) $userid = $_SESSION["userid"];
    else $userid = "";
    if (isset($_SESSION["username"])) $username = $_SESSION["username"];
    else $username = "";
    if (isset($_SESSION["userlevel"])) $userlevel = $_SESSION["userlevel"];
    else $userlevel = "";
    if (isset($_SESSION["userpoint"])) $userlevel = $_SESSION["userpoint"];
    else $userpoint = "";
?>
<div id="top">
    <h3>
        <a href="index.php">PHP 프로그래밍 입문</a>
    </h3>
    <ul id="top_menu">
```

①

②

③

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

```
<?php
    if (!$userid) {
?>
        <li> <a href="member_form.php">회원 가입(11장)</a> </li>
        <li> | </li>
        <li> <a href="login_form.php">로그인(12장)</a> </li>
<?php
    } else {
        $logged = $username."(".$userid.")님[Level: ".$userlevel.",
        Point: ".$userpoint."];
?>
        <li> <?=$logged?> </li>
        <li> | </li>
        <li> <a href="logout.php">로그아웃(12장)</a> </li>
        <li> | </li>
        <li> <a href="member_modify_form.php">정보 수정(12장)</a> </li>
```

④

⑤

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

```
<?php
    }
?>
<?php
    if ($userlevel == 1) {
?>
        <li> | </li>
        <li> <a href="admin.php">관리자 모드(15장)</a> </li>
<?php
    }
?>
    </ul>
</div>
```

⑥

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

```
<div id="menu_bar">
  <ul>
    <li> <a href="index.php">HOME</a> </li>
    <li> <a href="message_form.php">쪽지 만들기(13장)</a> </li>
    <li> <a href="board_form.php">게시판 만들기(14장)</a> </li>
    <li> <a href="index.php">사이트 완성하기(15장)</a> </li>
  </ul>
</div>
```


[예제 12-3] 상단 헤더 페이지 `part3/header.php`

① 세션 시작

- `session_start()` 함수로 세션 시작.
- 세션을 저장하거나 저장된 세션을 사용할 때는 먼저 세션을 시작해야 함.

② 세션 변수 검사

- `isset($_SESSION["userid"])`는 세션 변수인 `$_SESSION["userid"]`에 값이 있는지 검사.
- 값이 있으면 `true`, 값이 없으면 `false` 반환.
- 세션 변수 `$_SESSION["userid"]`에 값이 설정되어 있으면 그 값을 `$userid`에 저장.

isset() 함수	
형식	<code>isset(var)</code>
기능	변수에 값이 설정되어 있는지 검사한다.
인자	var: 변수명
반환 값	변수 값 설정 시: <code>true</code> 변수 값 미설정 시: <code>false</code>

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

③ \$userid에 null 값 설정

- 세션 변수 값이 설정되어 있지 않으면 ②의 if문 조건식이 거짓이므로 \$userid에 null 값 설정.
즉, 로그인 상태가 아니면 \$userid는 null 값.
- 로그인 상태에서는 [예제 12-2] ⑪의 세션 변수 저장과 [예제 12-3] ②의 세션 변수 가져오기를
통해 \$userid가 해당 아이디 값을 가지게 됨.

④ 로그인되지 않은 상태

- \$userid가 null 값인 " "라면 로그인되지 않은 상태.
- if문 조건식인 !\$userid에 의해 웹 페이지 상단에 [회원 가입(11장)]과 [로그인(12장)] 메뉴 출력.



그림 12-3 로그인되지 않은 상태의 상단 메뉴

[예제 12-3] 상단 헤더 페이지 `part3/header.php`

⑤ 로그인 상태

- \$userid가 값을 가진 경우라면 로그인 상태, ④의 if문 조건식이 거짓이 되어 else 다음 문장 수행.
- 로그인 상태이므로 사용자 이름, 아이디, 레벨, 포인트 출력. 웹 페이지 상단에 [로그아웃(12장)]과 [정보 수정(12장)] 메뉴 나타남.



그림 12-4 로그인 상태의 상단 메뉴

⑥ 관리자 모드 버튼

- 로그인한 사용자가 관리자인지 확인.
- 관리자라면 관리자 페이지에 접속할 수 있는 메뉴 생성.

3. 로그아웃 처리

- 세션을 이용하여 로그인 처리를 한 경우에는 생성된 세션을 삭제하면 로그아웃 처리 가능.
- 웹 사이트에 로그인된 상태의 사용자가 상단의 [로그아웃(12장)] 메뉴를 클릭하면 링크된 로그아웃 처리 페이지(`logout.php`)에 의해 로그아웃 상태로 변경.

[예제 12-4] 로그아웃 처리 페이지 `part3/logout.php`

```
<?php
    session_start();
    unset($_SESSION["userid"]);
    unset($_SESSION["username"]);
    unset($_SESSION["userlevel"]);
    unset($_SESSION["userpoint"]);

    echo("
        <script>
            location.href = 'index.php';
        </script>
    ");
?>
```

[예제 12-4] 로그아웃 처리 페이지 `part3/logout.php`

- unset() 함수를 이용하여 세션 변수 삭제. 메인 페이지인 index.php로 이동.
- 세션 변수인 \$_SESSION["userid"]가 삭제되면 [예제 12-3]의 ②와 ③에 의해 \$userid는 null 값인 " "가 되고 ④의 if문 조건식이 참이 되므로 [그림 12-3]과 같이 로그인되지 않은 상태의 상단 메뉴가 나타남.

1. 회원 정보 수정

- 로그인 상태의 메인 페이지 상단에는 회원 정보를 수정할 수 있는 [정보 수정(12장)] 메뉴가 있음.
- 이 메뉴를 클릭하면 [예제 12-3]의 ⑤에 링크된 회원 정보 수정 페이지 (member_modify_form.php)로 이동하여 DB에 있는 회원 정보를 수정할 수 있음.

03 회원 정보 수정 페이지

IT COOKBOOK 250

PHP 프로그래밍 입문

루바토(rubato)님 [Level:9, Point:0] | 로그아웃(12장) | 정보 수정(12장)

HOME

쪽지 만들기(13장)

게시판 만들기(14장)

사이트 완성하기(15장)

PHP Coding
Server Side Web Programming



회원 정보 수정

아이디

rubato

비밀번호

비밀번호 확인

이름

루바토

이메일

@naver.com

저장하기

취소하기

PHP 프로그래밍 입문 | 한빛아카데미

예제 소스 다운로드

- 한빛아카데미(<http://hanbit.co.kr>)

- 코딩스쿨(<http://codingschool.info>)

저자 문의 메일

- 메일 주소 : goldmont@naver.com

그림 12-5 회원 정보 수정 페이지(member_modify_form.php)

- 회원 정보 수정 페이지는 회원 가입 페이지(member_form.php)와 거의 유사. 가장 큰 차이점은 회원 가입 시 입력한 정보가 입력 창에 나타난다는 것.

표 12-2 회원 정보 수정 페이지 구성에 필요한 파일

구분	파일	설명
CSS	css/common.css	사이트에 공통으로 사용되는 CSS 파일
	css/member.css	회원 정보 수정 페이지의 CSS 파일[회원 가입 페이지(member_form.php)와 동일한 CSS 파일 사용]
자바스크립트	js/member_modify.js	회원 정보 수정 페이지의 자바스크립트 파일
회원 정보 수정 페이지	member_modify_form.php	회원 정보 수정 페이지
기능 페이지	member_modify.php	DB에 저장된 회원 정보 수정

[예제 12-5] 회원 정보 수정 페이지 `part3/member_modify_form.php`

```
<!-- 생략 -->
<link rel="stylesheet" type="text/css" href="/css/common.css">
<link rel="stylesheet" type="text/css" href="/css/member.css">
<script type="text/javascript" src="/js/member_modify.js"> </script>
<!-- 생략 -->
<?php
    $con = mysqli_connect("localhost", "user1", "12345", "sample");
    $sql = "select * from members where id='$userid'";
    $result = mysqli_query($con, $sql);
    $row = mysqli_fetch_array($result);
    $pass = $row["pass"];
    $name = $row["name"];
    $email = explode("@", $row["email"]);
    $email1 = $email[0];
    $email2 = $email[1];
    mysqli_close($con);
?>
```

①

②

③

[예제 12-5] 회원 정보 수정 페이지

part3/member_modify_form.php

```
<section>
```

```
  <div id="main_img_bar">
```

```
    
```

```
  </div>
```

```
  <div id="main_content">
```

```
    <div id="join_box">
```

```
      <form name="member_form" method="post" action="member_
        modify.php?id=<?=$userid?>">
```

④

```
      <h2>회원 정보 수정</h2>
```

```
      <div class="form id">
```

```
        <div class="col1">아이디</div>
```

```
        <div class="col2">
```

```
          <?=$userid?>
```

⑤

```
        </div>
```

```
      </div>
```

```
      <div class="clear"></div>
```

[예제 12-5] 회원 정보 수정 페이지 `part3/member_modify_form.php`

```
<div class="form">
  <div class="col1">비밀번호</div>
  <div class="col2">
    <input type="password" name="pass" value=
      "<?=$pass?>">
  </div>
</div>
```

⑥

```
<div class="clear"></div>
<div class="form">
  <div class="col1">비밀번호 확인</div>
  <div class="col2">
    <input type="password" name="pass_confirm"
      value="<?=$pass?>">
  </div>
</div>
```

⑦

[예제 12-5] 회원 정보 수정 페이지

part3/member_modify_form.php

```
<div class="clear"> </div>
<div class="form">
  <div class="col1">이름</div>
  <div class="col2">
    <input type="text" name="name" value="<?=$name?>">
  </div>
</div>
```

⑧

```
<div class="clear"> </div>
<div class="form email">
  <div class="col1">이메일</div>
  <div class="col2">
    <input type="text" name="email1"
      value="<?=$email1?>">@<input type="text"
      name="email2" value="<?=$email2?>">
  </div>
</div>
```

⑨

[예제 12-5] 회원 정보 수정 페이지

part3/member_modify_form.php

```
<div class="clear"> </div>
<div class="bottom_line"> </div>
<div class="buttons">
    &nbsp;
    
</div>
</form>
</div> <!-- join_box -->
</div> <!-- main_content -->
</section>
<footer>
    <?php include "footer.php";?>
</footer>
</body> </html>
```

⑩

[예제 12-5] 회원 정보 수정 페이지 `part3/member_modify_form.php`

① 회원 가입 정보를 DB에서 가져오기

MySQL 서버에 접속하여 회원 가입 시 저장한 데이터를 members 테이블에서 호출.

② 비밀번호, 이름 저장

DB에서 가져온 비밀번호(\$row["pass"])와 이름(\$row["name"])을 각각 \$pass와 \$name에 저장.

③ 이메일 주소 분리

- 이메일 주소는 XXX@XXX 형태의 문자열 하나로 묶어 DB에 저장.
- 하나로 병합되어 있는 이메일 주소를 이메일 입력 창에 나누어 입력하려면 DB에서 가져온 이메일 주소인 \$row["email"]을 @을 기준으로 분리.
- explode("@",\$row["email"])은 @을 기준으로 \$row["email"]을 분리하여 배열에 저장.

explode() 함수	
형식	explode(separator, string)
기능	문자열을 분리하여 배열에 저장한다.
인자	separator: 분리의 기준이 되는 문자열 string: 분리 대상 문자열
반환 값	문자열의 배열

[예제 12-5] 회원 정보 수정 페이지 `part3/member_modify_form.php`

④ 정보 수정 폼 양식

- <form> 태그를 이용하여 정보 수정 폼 양식 정의.
- name 속성은 member_form, action 속성은 member_modify.php로 설정. 이 은 ⑩에서 사용.
- ⑩의 <저장하기> 버튼(button_save.gif)을 클릭하면 onclick="check_input()"에 의해 member_modify.js 파일에 정의한 check_input() 함수가 실행.
- check_input() 함수는 입력 창에 데이터가 제대로 입력되었는지, 비밀번호와 비밀번호 확인에 입력한 값이 일치하는지 확인.
- 이상이 없으면 ④의 action 속성에 설정된 member_modify.php로 이동.

⑤ 아이디 출력

- 로그인 시 입력한 아이디는 [예제 12-3]의 ②에서 \$userid에 저장.
- 이 값을 출력하여 아이디 항목 옆에 사용자 아이디가 나타나게 함([그림 12-5]).

⑥ ~ ⑦ 비밀번호와 비밀번호 확인 입력 창에 데이터 삽입

DB에 저장된 비밀번호인 \$pass를 비밀번호와 비밀번호 확인 입력 창에 삽입.

[예제 12-5] 회원 정보 수정 페이지 `part3/member_modify_form.php`

⑧ 이름 입력 창에 데이터 삽입

DB에 저장된 이름인 `$name`을 이름 입력 창에 삽입.

⑨ 이메일 입력 창에 데이터 삽입

DB에 저장된 이메일을 가져와서 분리한 `$email1`과 `$email2`를 이메일 입력 창에 삽입.

⑩ 회원 정보 수정 버튼

- 회원 정보 수정 버튼인 <저장하기>(button_save.gif)를 클릭하면 ④에서 설명한 것과 같이 member_modify.php로 이동.
- member_modify.php에 서는 사용자가 변경한 회원 정보로 DB를 업데이트.

[예제 12-6] 회원 정보 수정 처리 페이지

part3/member_modify.php

```
<?php
```

```
$id = $_GET["id"];
```

①

```
$pass = $_POST["pass"];
```

②

```
$name = $_POST["name"];
```

```
$email1 = $_POST["email1"];
```

```
$email2 = $_POST["email2"];
```

```
$email = $email1."@".$email2;
```

```
$con = mysqli_connect("localhost", "user1", "12345", "sample");
```

③

```
$sql = "update members set pass='$pass', name='$name', email='$email'";
```

④

```
$sql .= " where id='$id'";
```

```
mysqli_query($con, $sql);
```

[예제 12-6] 회원 정보 수정 처리 페이지 [part3/member_modify.php](#)

```
mysqli_close($con);

echo "
    <script>
        location.href = 'index.php';
    </script>
";
?>
```

⑤

2. 회원 정보 수정 처리

① 아이디 전달받기

[예제 12-5]의 ④에서 `action="member_modify.php?id=<?=$userid?>"`로 전달한 id를 GET 방식으로 가져와 `$id`에 저장.

② 비밀번호, 이름, 이메일 전달받기

회원 정보 수정 양식에서 사용자가 입력한 비밀번호, 이름, 이메일을 POST 방식으로 전달받아 `$pass`, `$name`, `$email1`, `$email2`에 저장.

③ MySQL 서버 접속

`mysqli_connect()` 함수를 이용하여 MySQL 서버에 접속.

④ DB 업데이트

MySQL의 `update` 명령을 이용하여 `members` 테이블에서 해당 아이디를 가진 레코드의 비밀번호, 이름, 이메일을 수정.

⑤ 메인 페이지로 이동

실습 사이트의 메인 페이지인 `index.php`로 이동.

- 실습 사이트 메인 페이지의 오른쪽 상단에 있는 [정보 수정(12장)] 메뉴를 클릭하여 회원 정보를 수정한 뒤 제대로 변경되었는지 확인.