

Chapter 13 HTML-PHP-MySQL 연동 : CRUD 구현

Chapter 8에서 HTML 폼과 PHP의 연동을 학습했고, Chapter 12에서는 PHP와 MySQL과의 연동을 살펴봤습니다.

Chapter 13장에서는 HTML 폼, PHP, MySQL의 연동 방법을 살펴보겠습니다.

이를 위해 CRUD 기능으로 웹서비스를 구현하는 과정으로 진행합니다.

CRUD는 Create, Read, Update, Delete를 뜻하며 데이터베이스를 사용하는데 기초적인 기능입니다. 이를 HTML 폼, PHP, MySQL과 연동하여 웹서비스를 완성하는 과정을 살펴봅니다.

기능	설명
Create	테이블 생성
Read	테이블의 데이터 검색
Update	테이블의 데이터 수정
Delete	테이블의 데이터 삭제

Chapter 11~12에서 실습했던 데이터베이스, 테이블 등은 모두 삭제합니다.

제공되는 파일들은 『htdocs\source\13\friend』에 있습니다.

1. 친구 주소록 관리

다음의 정보를 이용하여 HTML 폼을 이용하여 데이터베이스와 테이블 생성하고, 데이터 입력 및 검색하는 예제를 실습합니다.

데이터베이스명 : mydb 테이블명 : friend_tbl	
필드명	데이터 타입
name	char(5)
age	int(3)
address	varchar(15)

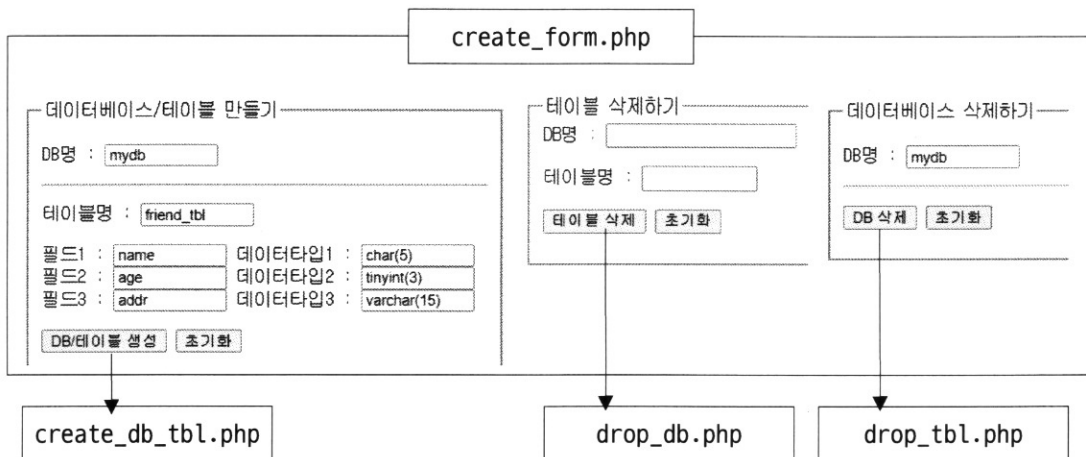
name	age	addr
윤영선	23	부산시 사하구
선영재	21	인천시 남동구
박정진	32	서울시 용산구

1.1 HTML 폼을 이용한 DB/테이블 생성

DB/테이블 생성을 위한 파일명과 기능은 다음과 같습니다.

파일명	기능
create_form.php	데이터베이스와 테이블을 생성하는 HTML 폼 화면 데이터베이스와 테이블을 삭제하는 HTML 폼 화면
create_db_tbl.php	데이터베이스와 테이블을 생성하는 PHP 파일
drop_tbl.php	테이블을 삭제하는 PHP 파일
drop_db.php	데이터베이스를 삭제하는 PHP 파일

다음 그림과 같이 create_form.php 파일에는 폼이 3개 있습니다. 각 폼의 버튼을 클릭하면 action 속성에 있는 파일이 기능을 수행하게 됩니다.



각 파일의 중요한 부분만 설명합니다. 자세한 내용은 제공된 소스코드로 분석하기 바랍니다.

```

create_form.php
1  <form name="create_db" method="post" action="create_db_tbl.php">
2    DB명 : <input type="text" name="db_name" size=10><br>
3    테이블명 : <input type="text" name="tbl_name" size=10>
4    <tr><td>필드1 : <input type="text" name="fld_name1" size=10>
5      :
6      <input type="submit" value="DB/테이블 생성"> <input type="reset">
7  <form name="drop_tbl_db" method="post" action="drop_tbl.php">
8    DB명 : <input type="text" name="db_name">
9    테이블명 : <input type="text" name="tbl_name" size=10><br>
10   <input type="submit" value="테이블 삭제"> <input type="reset">
11 <form name="dorp_db" method="post" action="drop_db.php">
12   DB명 : <input type="text" name="db_name" size=10><br>
13   <input type="submit" value="DB 삭제"> <input type="reset">
  
```

- 1~6번 라인 : 생성하기 위한 데이터베이스명과 테이블명, 필드명을 사용자가 입력하는 폼
- 7~12번 라인 : 삭제하기 위한 테이블명을 사용자가 입력하는 폼(hidden 기능으로 데이터베이스명과 테이블명을 drop_tbl.php 파일에 전달)
- 12~15번 라인 : 삭제하기 위한 데이터베이스명을 사용자가 입력하는 폼(hidden 기능으로 데이터베이스명을 drop_db.php 파일에 전달)

create_db_tbl.php

```

1  $db_name = $_POST["db_name"]; $tbl_name = $_POST["tbl_name"];
2  $fld_name1 = $_POST["fld_name1"];
3      :      :
4  $fld_type3 = $_POST["fld_type3"];
5  $host = "localhost"; $user = "root"; $password = "1234";
6  $link = new mysqli($host, $user, $password);
7  $link->query("create database $db_name");
8  $result = $link->query("show databases");
9  $fld_num = $result->field_count;
10 while ($data = $result->fetch_row()) {
11     for($i=0; $i<$fld_num; $i++)
12     {
13         echo "<td>$data[$i]</td>";
14     }
15 }
16 $link->query("use $db_name");
17 $create_tbl = "create table $tbl_name (
18 $fld_name1 $fld_type1, $fld_name2 $fld_type2, $fld_name3 $fld_type3
19 );";
20 $link->query($create_tbl);

```

- 1~4번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리
- 5~6번 라인 : DB 접속 정보를 \$link 객체에 저장
- 7번 라인 : \$db_name 변수명에 저장된 데이터베이스명으로 생성
- 8번 라인 : DB 목록 쿼리문의 결과를 \$result 객체 변수에 저장
- 9번 라인 : 필드 개수를 계산하여 \$fld_num 변수에 저장
- 10~15번 라인 : \$result 객체 변수에 저장된 내용을 fetch_row() 메서드를 이용하여 레코드 단위로 추출하고 필드 개수(\$fld_num) 만큼의 데이터를 출력
- 16번 라인 : 테이블 생성을 위한 데이터베이스 선택
- 17~20번 라인 : 폼에서 전송된 테이블명, 필드명, 데이터타입으로 테이블을 생성
- * 9~15번 라인은 『show tables』, 『desc \$tbl_name』 쿼리문의 결과를 출력하기 위해 공통적으로 사용됨
- ※ 생성된 데이터베이스와 테이블을 MySQL 콘솔로 확인해 보세요.

```
MariaDB [(none)]> show databases;
```

Database
information_schema
mydb
mysql

```
MariaDB [(none)]> use mydb;
```

```
MariaDB [mydb]> show tables;
```

Tables_in_mydb
friend_tbl

```
MariaDB [mydb]> desc friend_tbl;
```

Field	Type	Null	Key	Default	Extra
name	char(5)	YES		NULL	
age	tinyint(3)	YES		NULL	
addr	varchar(15)	YES		NULL	

※ HTML 폼없이 PHP 파일에서 직접 생성하는 방법은 『one_process.php』 파일을 참고하세요.

1.2 HTML 폼을 이용한 테이블 삭제

『create_form.php』 파일의 폼에서 전송된 정보를 이용하여 테이블을 삭제합니다.

drop_tbl.php

```

1  $db_name = $_POST["db_name"];
2  $tbl_name = $_POST["tbl_name"];
3  $host = "localhost"; $user = "root"; $password = "1234";
4  $link = new mysqli($host, $user, $password);
5  $link->query("use $db_name");
6  $link->query("drop table $tbl_name");
7  $result = $link->query("show tables");
8  $fld_num = $result->field_count;
9  while ($data = $result->fetch_row()) {
10     :
11 }
```

- 1~2번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리
- 6번 라인 : \$tbl_name 변수명에 저장된 테이블명 삭제
- 7번 라인 : 테이블 목록 쿼리문의 결과를 \$result 객체 변수에 저장
- 8~11번 라인 : 『create_db_tbl.php』 파일과 동일한 기능 수행

1.3 HTML 폼을 이용한 DB 삭제

『create_form.php』 파일의 폼에서 전송된 정보를 이용하여 데이터베이스를 삭제합니다.

```

drop_db.php
1  $db_name = $_POST["db_name"];
2  $host = "localhost"; $user = "root"; $password = "1234";
3  $link = new mysqli($host, $user, $password);
4  $link->query("drop database $db_name");
5  $result = $link->query("show databases");
6  $fld_num = $result->field_count;
7  while ($data = $result->fetch_row()) {
8      :
9  }

```

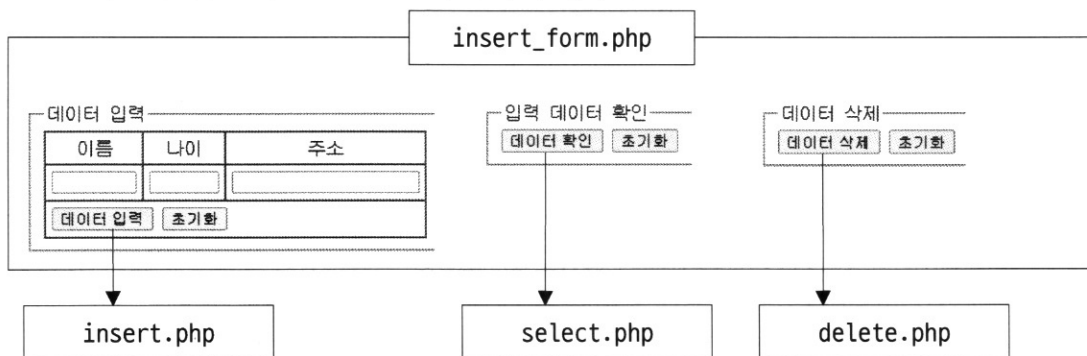
- 1번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리
- 4번 라인 : \$db_name 변수명에 저장된 데이터베이스명 삭제
- 5번 라인 : 데이터베이스 목록 쿼리문의 결과를 \$result 객체 변수에 저장
- 6-9번 라인 : 『create.php』 파일과 동일한 기능 수행

1.4 HTML 폼을 이용한 데이터 입력

데이터 입력을 위한 파일명과 기능은 다음과 같습니다.

파일명	기능
insert_form.php	데이터 입력을 위한 HTML 폼 화면
insert.php	데이터 입력을 처리하는 PHP 파일
select.php	데이터를 검색하는 PHP 파일
delete.php	데이터를 삭제하는 PHP 파일

다음 그림과 같이 insert_form.php 파일에는 폼이 3개 있습니다. 각 폼의 버튼을 클릭하면 action 속성에 있는 파일이 기능을 수행하게 됩니다.



insert_form.php	
1	<form name="insert_data" method="post" action="insert.php">
2	<input type="hidden" name="db_name" value="mydb">
3	<input type="hidden" name="tbl_name" value="friend_tbl">
4	<input type="text" name="name" size=6>
5	<input type="text" name="age" size=4>
6	<input type="text" name="addr" size=20>
7	<input type="submit" value="데이터 입력"> <input type="reset">
8	<form name="select_data" method="post" action="select.php">
9	<input type="hidden" name="db_name" value="mydb">
10	<input type="hidden" name="tbl_name" value="friend_tbl">
11	<input type="submit" value="데이터 확인"> <input type="reset">
12	<form name="delete_data" method="post" action="delete.php">
13	<input type="hidden" name="db_name" value="mydb">
14	<input type="hidden" name="tbl_name" value="friend_tbl">
15	<input type="submit" value="데이터 삭제"> <input type="reset">

- 1~7번 라인 : 데이터를 사용자가 입력하는 폼(hidden 기능으로 데이터베이스명과 테이블명을 insert.php 파일에 전달)
- 8~11번 라인 : 입력된 데이터를 검색하는 폼(hidden 기능으로 데이터베이스명과 테이블명을 select.php 파일에 전달)
- 12~15번 라인 : 입력된 데이터를 삭제하는 폼(hidden 기능으로 데이터베이스명을 delete.php 파일에 전달)

insert.php	
1	\$db_name = \$_POST["db_name"]; \$tbl_name = \$_POST["tbl_name"];
2	\$name = \$_POST["name"]; \$age = \$_POST["age"]; \$addr = \$_POST["addr"];
3	\$host = "localhost"; \$user = "root"; \$password = "1234";
4	\$link = new mysqli(\$host, \$user, \$password);
5	\$link->query("use \$db_name");
6	\$result = \$link->query("insert into \$tbl_name(name,age,addr) value('\$name', '\$age', '\$addr');");
7	\$result = \$link->query("select * from \$tbl_name");
8	\$fld_num = \$result->field_count;
9	while (\$data = \$result->fetch_row()) {
10	for(\$i=0; \$i<\$fld_num; \$i++)
11	{
12	echo "<td>\$data[\$i]</td>";
13	}
14	}

- 1~2번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리
- 6번 라인 : \$name, \$age, \$addr 변수명에 저장된 데이터를 \$tbl_name 테이블에 입력
- 7번 라인 : DB 목록 쿼리문의 결과를 \$result 객체 변수에 저장
- 8번 라인 : 필드 개수를 계산하여 \$fld_num 변수에 저장
- 9~14번 라인 : \$result 객체 변수에 저장된 내용을 fetch_row() 메서드를 이용하여 레코드 단위로 추출하고 필드 개수(\$fld_num) 만큼의 데이터를 출력

1.5 HTML 폼을 이용한 데이터 검색

insert.php

```
1 $db_name = $_POST["db_name"];
2 $tbl_name = $_POST["tbl_name"];
3 $host = "localhost"; $user = "root"; $password = "1234";
4 $link = new mysqli($host, $user, $password);
5 $link->query("use $db_name");
6 $result = $link->query("select * from $tbl_name");
```

- 1~2번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리

- 6번 라인 : \$tbl_name 테이블에 입력된 데이터를 검색

※ 입력된 데이터를 MySQL 콘솔로 확인해 보세요.

MariaDB [mydb]> select * from friend_tbl;

name	age	addr
윤영선	23	부산시 사하구
선영재	21	인천시 남동구
박정진	32	서울시 용산구

1.6 HTML 폼을 이용한 데이터 삭제

delete.php

```
1 $db_name = $_POST["db_name"];
2 $tbl_name = $_POST["tbl_name"];
3 $host = "localhost"; $user = "root"; $password = "1234";
4 $link = new mysqli($host, $user, $password);
5 $link->query("use $db_name");
6 $link->query("delete from $tbl_name");
```

- 1~2번 라인 : 폼 파일에서 전송된 사용자 입력 내용을 PHP 변수 처리

- 6번 라인 : \$tbl_name 테이블에 입력된 데이터를 삭제

※ 삭제된 데이터를 MySQL 콘솔로 확인해 보세요.

MariaDB [mydb]> select * from friend_tbl;

Empty set (0.000 sec)