

17

HTML & FORM

HTML은 크게 화면을 구성하는 태그와 사용자의 입력을 받아 서버로 전송하는 FORM 구성 요소로 구분할 수 있습니다.

FORM 구성 요소들은 서버로 데이터를 전송하는데, PHP는 이런 전송된 데이터를 받아서 처리할 수 있습니다.

17.1 FORM

HTML의 FORM 태그는 폼 요소들을 감싸고 있는 바디 부분과 그 안에 들어가는 요소들의 항목으로 구분할 수 있습니다.

17.1.1 FORM 태그

HTML에서 <form></form>으로 마크업된 코드를 form 태그라고 합니다. 폼 태그는 폼 요소들을 포함하는 바디와 같습니다.

```
<form>
    폼요소들...
</form>
```

폼 요소들은 폼 태그로 둘러싸여 있어야 합니다.

폼 태그는 몇 개의 속성 설정을 할 수 있습니다.

name

폼의 이름을 지정합니다.

method

메서드 속성은 입력된 데이터의 전송 방식을 설정합니다. POST 방식 또는 GET 방식으로 지정할 수 있습니다. 만일 POST 방식으로 설정했다면 서버 쪽 PHP에서는 슈퍼 변수 \$_POST 배열을 통해서 데이터를 가져올 수 있습니다.

GET 방식으로 설정했다면 슈퍼 변수 \$_GET 배열을 통해서 데이터를 읽어올 수 있습니다.

action

액션 속성은 해당 폼을 처리하는 PHP 스크립트의 url과 파일명을 지정합니다. 폼이 서밋 되면 액션에서 지정한 속성으로 페이지가 이동되면서 데이터를 처리합니다.

enctype="multipart/form-data"

파일 업로드와 같이 바이너리 데이터를 전송할 경우에 추가되는 속성입니다.

17.1.2 Action

폼 태그의 속성 중 action은 서밋 클릭 시 처리되는 스크립트 페이지입니다. 전형적인 HTML 코딩에서는 직접 url과 스크립트명을 입력합니다.

```
<form method="post" action=".test.php">
```

하지만 form 요소를 작성하면서 처리되는 action 파일명을 직접 지정하는 방식은 불편합니다. 또한 잘못 이름을 타이핑하여 오동작을 발생할 수도 있을 것입니다.

이런 경우 슈퍼 변수 \$_SERVER["PHP_SELF"] 값을 통하여 다음과 같이 편리하게 사용하기도 합니다.

```
<form method="post" action="$_SERVER['PHP_SELF']">
```

\$_SERVER["PHP_SELF"]은 슈퍼 글로벌 변수입니다. 즉 action 값으로 현재 실행하고 있는 PHP 스크립트의 파일명을 가리키고 있습니다.

action="" 부분에 \$_SERVER["PHP_SELF"]를 입력하는 것은 서밋 버튼 클릭 시 다른 페이지로 이동하지 말고, 다시 자기 자신으로 다시 호출하여 데이터를 처리하라는 의미입니다.

하지만 이렇게 편리하게 사용한 \$_SERVER["PHP_SELF"] 테크닉은 요즘 들어 보안적으로 문제가 발생되고 있습니다. 이러한 셀프 페이지 방식은 동일한 페이지 form에서 오류 메시지를 출력하기 좋습니다.

웹 페이지는 불특정 다수의 사람들에게 노출되는 특성이 있습니다. 그중에서는 정상적인 사용자 이외에 해커 및 블랙유저 또한 있을 것입니다. PHP의 서버 전역 변수 \$_SERVER["PHP_SELF"]의 값은 해커의 이해 노출될 수 있습니다.

PHP_SELF를 사용하면 해커들을 슬래시(/)를 입력한 후에 XSS(Cross Site Scripting) 명령을 입력하여 정상 실행을 방해할 수 있습니다. XSS는 웹 응용프로그램에서 일반적으로 발견되는 컴퓨터 보안적 취약점 유형입니다. XSS를 사용하면 공격자가 다른 사용자가 본 웹 페이지에 클라이언트 측 스크립트를 삽입할 수 있습니다.

예를 들어 현재의 스크립트 test.php에서 폼 액션 값을 다음과 같이 설정했다고 생각해 봅시다.

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

이 페이지가 정상적으로 서밋했다면 현재의 페이지 test.php일 것입니다.

```
<form method="post" action="test.php">
```

하지만 해커가 브라우저 주소에 `http://www.도메인.com/test.php%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E`처럼 입력했다면 어떻게 될까요?

특수문자를 풀어서 보면은 위의 주소는 다음과 같습니다.

```
<form method="post" action="test.php/"><script>alert('hacked')</script>
```

즉, 폼 태그 뒤에 추가로 자바스크립트 코드가 더 들어 있습니다. 변경된 폼 요소는 자바스크립트를 실행하고 페이지가 해킹될 수도 있습니다.

`$_SERVER["PHP_SELF"]`의 보안적인 부분 때문에 직접 변수를 사용하지 않고 `htmlspecialchars()` 함수를 같이 이용하여 처리합니다.

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

`htmlspecialchars()` 함수는 HTML의 특수문자들을 변환 처리합니다. 특수문자란 `<`, `>` 그리고 `<` 와 `>`를 말합니다.

이런 특수문자는 HTML 또는 자바스크립트 코드 등에서 인젝션 공격을 당할 수 있습니다. 또한 사이트가 위협에 노출될 수 있습니다.

다시 한번 예를 들면,

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

htmlspecialchars() 함수를 같이 사용하면

```
<form method="post" action="test.php/quote"><script>alert('hacked')</script></form>
```

형태로 변경이 되기 때문에 좀 더 안전한 실행이 가능할 것입니다.

17.1.3 서밋

서밋(submit)은 form에서 입력된 데이터를 action에서 지정한 url로 전송을 처리하는 승인 버튼입니다. 즉, 클라이언트 브라우저에서 사용자 데이터 값을 입력 후에 서밋 버튼을 클릭하면 서버 쪽으로 데이터를 전송합니다.

기본적으로 서밋을 지정하는 양식은 다음과 같습니다.

```
<input type='submit' value='확인'>
```

input 태그로 타입을 submit으로 설정하면 됩니다. 또한 버튼의 이름을 value 속성값으로 설정하면 됩니다.

또한 서밋 버튼을 클릭 시 자바스크립트를 실행하여 처리할 수도 있습니다.

```
<input type='submit' value='확인' onclick='\"javascript:check_submit()\"' >
```

서버로 데이터 값을 전달하기 전에 자바스크립트를 통하여 필수 항목이나 유효성 등을 확인한 후에 처리할 수 있습니다.

타입 형식을 image로 설정할 경우 이미지 버튼으로 사용할 수 있습니다.

```
<input type='image' src='img/submit.png' >
```

17.1.4 리셋

리셋(reset)은 서밋의 반대로 입력한 폼의 값들을 모두 초기화합니다.

```
<input type='reset' value='지우기'>
```

17.2 요소

HTML은 폼 태그 안에 다양한 요소들을 삽입하여 데이터를 입력받을 수 있습니다.

폼의 요소 값을 입력받아 처리할 때는 프로그램 동작 오류를 줄이기 위해서 필수 체크와 유효성 값을 같이 처리하는 것이 필요합니다.

필수 체크

폼 입력은 사용자가 입력한 데이터를 기반으로 서버 PHP에서 데이터를 처리합니다. 만일 사용자가 실수로 PHP에서 처리해야 하는 값을 누락할 수도 있습니다.

이런 경우 다음에 처리할 유효성 체크 부분과 정상적인 프로그램 로직 처리가 불가능해질 수도 있습니다. 물론, 폼을 처리하기 전에 자바스크립트 등으로 체크한다고 해도 외부에서 잘못된 경로로 처리 PHP로 접근하여 오류를 발행하거나, 해킹될 수도 있을 것입니다.

따라서 폼으로 입력을 받은 데이터 값 중에 반드시 필요로 하는 값 등은 필수 체크를 중복으로 하는 것이 좋습니다.

값을 입력받을 때 다음과 같이 empty() 함수를 응용하여,

```
if (empty($_POST["name"])) {  
    참: 비어 있습니다...  
} else {  
    거짓: 비어 있지 않습니다...  
}
```

패턴으로 필수 체크를 하는 것이 좋습니다.

유효성

폼의 유효성 데이터 값을 체크하는 것은 매우 중요합니다. 의미 없는 값을 입력하여 프로그램의 오류를 발생시키거나, 해킹 등의 원인이 될 수도 있습니다.

따라서 정확한 처리 로직과 보안을 위해서 반드시 데이터를 처리하기 전에 PHP에서도 한 번 더 데이터를 검사하는 작업을 하는 것이 좋습니다.

PHP의 필터링 함수의 기능을 이용하면 좀 더 복잡한 데이터의 유효성을 쉽게 확인할 수 있습니다.

17.2.1 input 텍스트

HTML의 input Text 요소는 폼 요소로 가장 사용 빈도가 높은 요소입니다. TEXT는 사용자로부터 다양한 문자열을 입력받을 수 있습니다. 입력되는 값이 문자열일 수도 있고, 또한 문자를 가장한 숫자일 수도 있습니다. 그 외에도 이메일, 이름, url 등 모두 문자열에 속하기 때문에 다양한 입력값들이 올 수 있습니다.

```
<input type="text" name="name">
```

예제 파일 | form_text.php

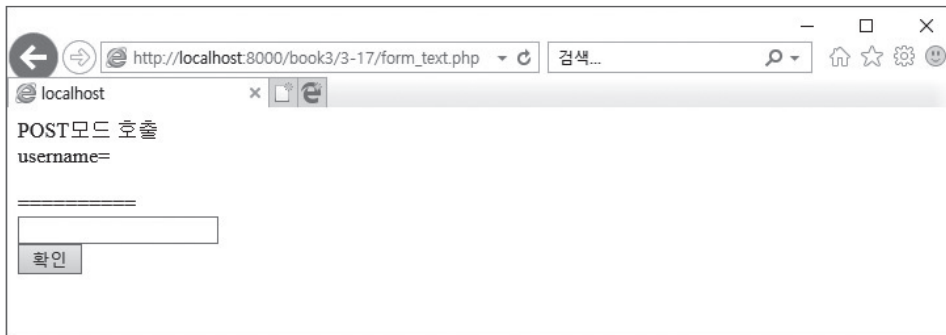
```
1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST 모드 호출 <br>";
5          echo "username= <br>" . $_POST['username'];
6          echo "<br>=====<br>";
7      }
8
9      // HTML form을 출력합니다.
10
11     echo "<form name='jiny' method='post' action='". $_SERVER['PHP_
12         SELF']. ">";
13
14     echo "<input type=\"text\" name=\"username\">";
```

```

14
15     echo "<br>";
16     echo "<input type='submit' value='확인'>";
17     echo "</form>";
18
19 ?>

```

화면 출력



다양한 타입 속성

HTML5의 input은 좀 더 많은 다양한 타입 속성을 지원합니다. 타입 속성을 이용하면 데이터 입력 시 발생할 수 있는 오류들을 먼저 브라우저 차원에서 필터링할 수 있습니다.

email
 url
 search
 tel
 number
 date
 datetime
 month
 week
 time
 range
 color

필수 체크

HTML5에는 폼의 필수 입력을 확인하는 속성인 **required**입니다. 하지만 이 속성은 브라우저에서만 동작하는 속성입니다. 브라우저 이외의 방식으로 접속할 경우에는 필수 체크를 할 수 없습니다. 이런 경우 PHP 소스상에서 다시 한번 필수 체크해주는 것이 중요합니다.

입력값 제한

HTML5는 브라우저에서 입력 문자열의 길이를 제한할 수 있습니다. `input` 태그의 속성으로 `maxlength="값"`을 추가하면 됩니다.

`maxlength`의 입력 제한 동작은 브라우저에서만 동작합니다. 브라우저를 제외한 방법으로 접속하여 데이터를 전송할 때는 `maxlength` 적용받지 않습니다.

입력 패턴

HTML5의 입력 패턴 속성을 통하여 데이터 형식을 제한할 수 있습니다. 패턴 속성과 정규표현 방법을 통하여 입력되는 데이터를 필터링합니다.

```
<input type="password" name="passwd" pattern="[0~9A-Za-z]">
```

최대, 최소

HTML5에서는 `min` 또는 `max` 속성값을 통하여 입력되는 데이터의 최소치와 최대치를 지정할 수도 있습니다.

초기값 지정

`input`의 텍스트 속성을 사용 시 사용자에게 보여주는 초기값 및 기본값을 미리 출력해서 보여줄 수 있습니다. 초기값은 약간 흐린 회색 계열의 문자로 출력함으로써 사용자 입력값과 구분합니다.

```
<input type="text" name="username" placeholder="이름을 입력해 주세요">
```

자동 완성 추천

검색 엔진이나 특정 값을 입력할 때 하단에 추천하는 단어들을 나타내는 경우를 본 적이 있을 것입니다. HTML5의 autocomplete 기능을 이용하면 쉽게 구현할 수 있습니다.

```
<input type="text" name="country" placeholder="국가를 입력해 주세요"
autocomplete="on" list="local">
<datalist id="local">
  <option value="대한민국">
  <option value="미국">
  <option value="일본">
</datalist>
```

17.2.2 hidden

text 속성은 브라우저 화면에 사용자 입력 박스를 출력합니다. 그와 반대로 hidden 속성은 서버로 데이터를 전송하는 값이지만 화면에는 출력되지 않습니다. 즉, 숨겨진 text 속성과 같습니다.

```
<input type="hidden" name="mode" value="editup">
```

17.2.3 textarea

HTML teatarea 요소는 input 요소와 달리 많은 양의 텍스트를 입력받을 수 있습니다.

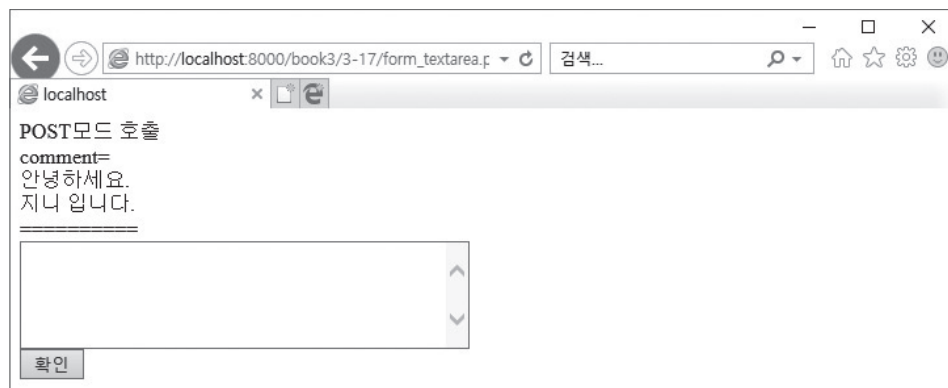
```
<textarea name="comment" rows="5" cols="40"></textarea>
```

textarea는 웹 페이지에서 글을 작성하는 본문 등 많이 이용합니다. textarea는 많은 양과 다양한 입력을 고려하여 데이터 안에 숨겨진 기호나 패턴 등을 복합적으로 판별해야 합니다.

예제 파일 | `form_textarea.php`

```
1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST 모드 호출 <br>";
5
6          // 다음 줄 n\를 <br>로 변환하여 출력함.
7          echo "comment= <br>" . nl2br($_POST['comment']);
8          echo "<br>=====<br>";
9      }
10
11     // HTML form을 출력합니다.
12
13     echo "<form name='jiny' method='post' action='". $_SERVER['PHP_
14         SELF']."'>";
15
16     echo "<textarea name=\"comment\" rows=\"5\" cols=\"40\"></textarea>";
17
18     echo "<br>";
19     echo "<input type='submit' value='확인'>";
20     echo "</form>";
21 ?>
```

화면 출력



POST모드 호출

comment=

안녕하세요.

지니 입니다.

확인

17.2.4 select

select 요소는 input 요소 다음으로 가장 많이 사용하는 html form 구성입니다. select는 드롭 다운 형태로 여러 개의 데이터 값 중에 1개 또는 다수의 데이터를 선택할 수 있습니다.

```
<select name="이름">
    <option value="옵션값1">옵션명1</option>
    <option value="옵션값2">옵션명2</option>
    ...
</select>
```

예제 파일 | **form_select1.php**

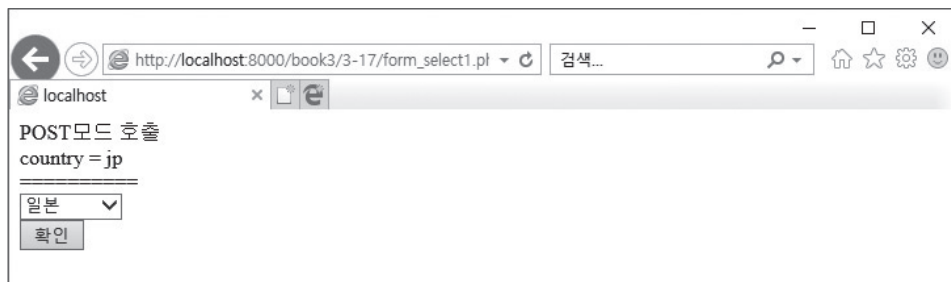
```
1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST 모드 호출 <br>";
5          echo "country = " . $_POST['country'] . "<br>";
6          echo "=====<br>";
7      }
8
9      // HTML form을 출력합니다.
10
11     echo "<form name='jiny' method='post' action='".$_SERVER['PHP_
12         SELF']."'>";
13
14     $body = "<select name=\"country\">";
15     $body .= "<option value=\"kr\">대한민국</option>";
16     $body .= "<option value=\"jp\">일본</option>";
17     $body .= "<option value=\"cn\">중국</option>";
18     $body .= "</select>";
19
20     // 기본 선택 값을 selected로 처리
21     if ($_POST['country'])
22         $body = str_replace(
23             "value=\"".$_POST['country'].\"",
24             "value=\"".$_POST['country']."\" selected",
```

```

24         $body);
25
26     echo $body;
27
28     echo "<br>";
29     echo "<input type='submit' value='확인'>";
30     echo "</form>";
31
32 ?>

```

화면 출력



select는 기본적으로 선택한 1개의 값을 지정할 수 있습니다. 하지만 여러 개의 항목 선택이 필요한 경우에는 `<select multiple name="이름[]">` 형태로 이름을 배열 형태로 `multiple` 속성을 사용하면 됩니다.

예제 파일 | form_select2.php

```

1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST모드 호출 <br>";
5          $COUNTRY = $_POST['country'];
6
7          for($i=0;$i<count($COUNTRY);$i++) {
8              echo $i . "=" . $COUNTRY[$i] . "<br>";
9          }
10         echo "=====<br>";
11     }
12
13     // HTML form을 출력합니다.

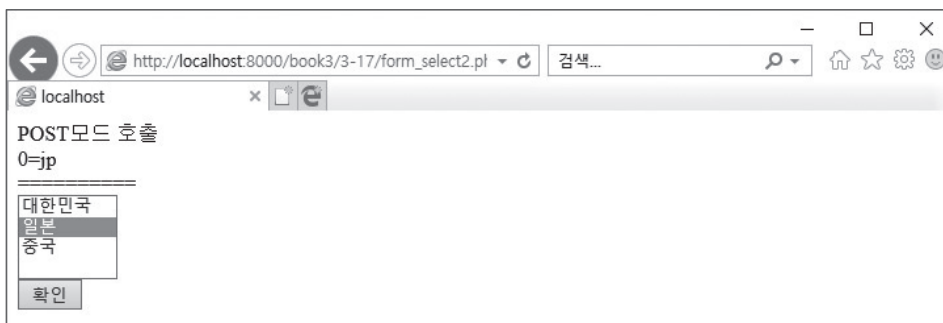
```

```

14
15     echo "<form name='jiny' method='post' action='".$_SERVER['PHP_
        SELF'].>";
16
17     $body = "<select name=\"country[]\" multiple>";
18     $body .= "<option value=\"kr\">대한민국</option>";
19     $body .= "<option value=\"jp\">일본</option>";
20     $body .= "<option value=\"cn\">중국</option>";
21     $body .= "</select>";
22
23     // 기본 선택 값을 selected로 처리
24     if ($_POST['country']) {
25         $COUNTRY = $_POST['country'];
26
27         for($i=0;$i<count($COUNTRY);$i++) {
28             $body = str_replace(
29                 "value=\"".$_COUNTRY[$i].\"",
30                 "value=\"".$_COUNTRY[$i].\" \" selected",
31                 $body);
32
33         }
34     }
35
36     echo $body;
37
38     echo "<br>";
39     echo "<input type='submit' value='확인'>";
40     echo "</form>";
41
42     ?>

```

화면 출력



17.2.5 checkbox

폼 요소 중 체크박스는 어떠한 항목에 동의 등을 구할 때 많이 사용하는 구성 요소입니다.

```
<input type='checkbox' name='이름'>
```

위와 같이 HTML 코드 입력하면 form의 데이터는 슈퍼 변수 \$_GET[] 또는 \$_POST[] 배열 변수를 통해서 값을 가져올 수 있습니다.

이때 지정한 name="이름" 값의 배열의 선택 키 값으로 사용됩니다. 만일 체크박스의 초기값을 "선택된"으로 하고자 할 때는 checked 키워드를 추가하면 됩니다.

체크박스가 선택된 값은 on 문자열 값으로 전달됩니다.

예제 파일 | **form_check1.php**

```
1  <?php
2      // 서버 및 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST모드 호출 <br>";
5          echo "agree = " . $_POST['agree'] . "<br>";
6          echo "adult = " . $_POST['adult'] . "<br>";
7          echo "=====<br>";
8      }
9
10     // HTML form을 출력합니다.
11
12     echo "<form name='jiny' method='post' action='". $_SERVER['PHP_
13         SELF']. ">";
14
15     echo "Agreement : ";
16     if ($_POST['agree']) {
17         echo "<input type='checkbox' name='agree' checked>";
18     } else {
19         echo "<input type='checkbox' name='agree'>";
20     }
21     echo "<br>";
```

```

21
22     echo "Adult : ";
23     if ($_POST['adult']) {
24         echo "<input type='checkbox' name='adult' checked>";
25     } else {
26         echo "<input type='checkbox' name='adult'>";
27     }
28
29     echo "<br>";
30     echo "<input type='submit' value='확인'>";
31     echo "</form>";
32
33 ?>

```

화면 출력

POST모드 호출
agree =
adult = on

Agreement : ☐
Adult : ☒

위의 예제1은 체크박스가 하나에 1개의 이름으로 각각 할당되어 있습니다. 다음 예제는 체크박스 이름을 배열 형태로 지정한 후에 value 속성을 이용하여 값을 할당합니다.

예제 파일 | form_check2.php

```

1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST모드 호출 <br>";
5          $TID = $_POST['TID'];
6
7          for($i=0;$i<count($TID);$i++) {
8              echo $i . "=" . $TID[$i] . "<br>";

```

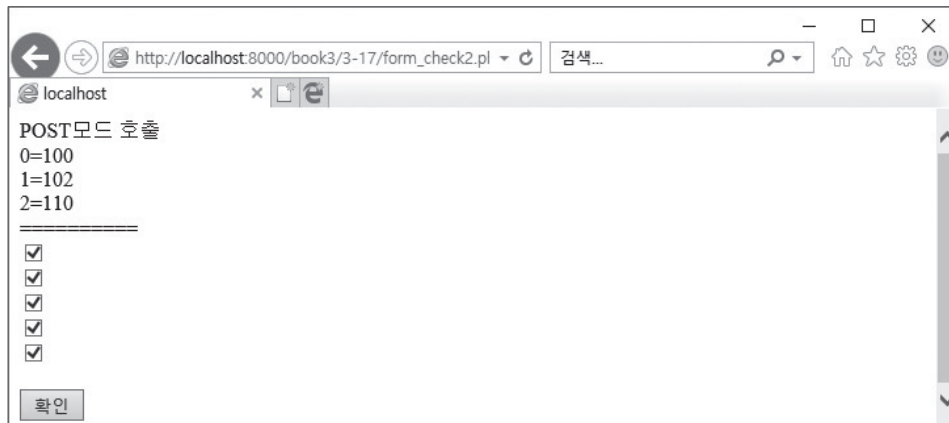


```

9         }
10
11         echo "=====<br>";
12     }
13
14     // HTML form을 출력합니다.
15
16     echo "<form name='jiny' method='post' action='".$_SERVER['PHP_
17     SELF']."'>";
18
19     // 체크 값의 이름이 배열 형태로 지정
20     echo "<input type='checkbox' name='TID[]' value='100' checked> <br>";
21     echo "<input type='checkbox' name='TID[]' value='102' checked> <br>";
22     echo "<input type='checkbox' name='TID[]' value='105' checked> <br>";
23     echo "<input type='checkbox' name='TID[]' value='107' checked> <br>";
24     echo "<input type='checkbox' name='TID[]' value='110' checked> <br>";
25
26     echo "<br>";
27     echo "<input type='submit' value='확인'>";
28     echo "</form>";
29     ?>

```

화면 출력



예제2에서 체크박스는 TID라는 이름의 배열로 작성되어 있습니다. 또한 각각의 체크박스는 value 속성을 통하여 고유값들을 가지고 있습니다.

서밋을 통하여 PHP로 값이 전달된 경우 배열로 체크박스는 배열 형태로 값을 전달받게 됩니다. 선택한 체크박스의 값만을 배열화하여 전달합니다.

이러한 유형은 리스트 목록 등에서 여러 개의 데이터를 선택하여 FORM으로 전달하는 용도로 많이 사용됩니다.

17.2.6 radio box

HTML의 라디오 요소는 여러 개의 선택 값 중에 하나만 입력받을 수 있는 요소입니다.

```
<input type="radio" name="sex" value="female">Female  
<input type="radio" name="sex" value="male">Male
```

라디오 버튼의 name="이름" 부분은 동일한 값을 가집니다. 하지만 속성 값은 각각의 다른 값을 갖게 됩니다.

예제 파일 | form_radio.php

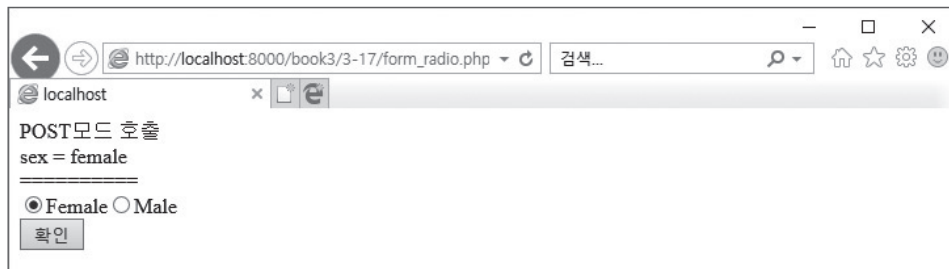
```
1  <?php  
2      // 서밋 처리 시 데이터 값 확인  
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {  
4          echo "POST모드 호출 <br>";  
5          echo "sex = " . $_POST['sex'] . "<br>";  
6          echo "=====<br>";  
7      }  
8  
9      // HTML form을 출력합니다.  
10  
11     echo "<form name='jiny' method='post' action='".$_SERVER['PHP_  
12         SELF']."'>";  
13  
14     if ($_POST['sex'] == "female") {  
15         echo "<input type=\"radio\" name=\"sex\" value=\"female\"  
16             checked>Female";  
17         echo "<input type=\"radio\" name=\"sex\" value=\"male\">Male";  
18     } else if ($_POST['sex'] == "male") {
```

```

18     echo "<input type=\"radio\" name=\"sex\" value=\"female\">Female";
19     echo "<input type=\"radio\" name=\"sex\" value=\"male\"
    checked>Male";
20
21 } else {
22     echo "<input type=\"radio\" name=\"sex\" value=\"female\">Female";
23     echo "<input type=\"radio\" name=\"sex\" value=\"male\">Male";
24 }
25
26 echo "<br>";
27 echo "<input type='submit' value='확인'>";
28 echo "</form>";
29
30 ?>

```

화면 출력



17.2.7 file

메일 작성이나 게시판에 글과 함께 첨부하는 파일 등을 지정하여 서버로 전송할 수 있습니다. 클라이언트 브라우저에서 서버로 파일을 전송하기 위해서는 form의 type="file"을 사용하면 됩니다.

```
<input type="file" name="upload">
```

HTML에서 파일 전송을 위해서는 전송 메서드 타입이 POST로 되어 있어야 합니다. 또한 바이너리 데이터를 함께 전송하기 위해서 속성 enctype="multipart/form-data"가 설정되어야 합니다.

HTML의 file 요소를 선택하면 컴퓨터는 서버로 전송할 파일을 선택하는 팝업 창을 브라우저가 실행시킵니다.

또한 PHP에서 파일을 업로드하여 전송하기 위해서는 약간의 php.ini 환경 설정이 필요합니다. php.ini 파일에서 file_uploads 부분을 ON으로 변경해야 합니다.

file_uploads = On

또한 파일 용량 제한도 가능합니다. 보다 자세한 것은 환경 설정 부분을 참조하면 됩니다.

예제 파일 | **form_upload.php**

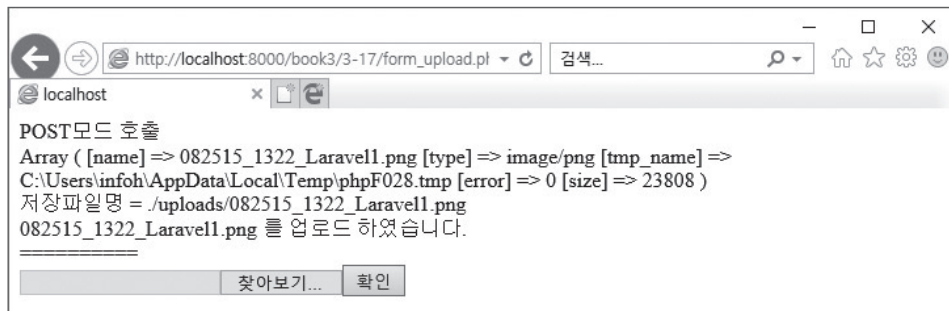
```
1  <?php
2      // 서밋 처리 시 데이터 값 확인
3      if ($_SERVER["REQUEST_METHOD"] == "POST") {
4          echo "POST 모드 호출 <br>";
5          print_r($_FILES['upload']);
6          echo "<br>";
7
8          // 업로드 디렉터리 지정
9          $target_dir = "./uploads";
10         if (!is_dir($target_dir)) {
11             mkdir($target_dir);
12         }
13
14         $target_file = $target_dir . "/" . basename($_FILES["upload"]
15         ["name"]);
16         echo "저장파일명 = ". $target_file . "<br>";
17
18         $uploadOk = true;
19
20         // 업로드할 파일이 존재하는지 검사
21         if (file_exists($target_file)) {
22             echo "이미 존재하는 파일명입니다.";
23             $uploadOk = false;
24         }
25
26         // 파일 사이즈 제한
27         if ($_FILES["upload"]["size"] > 500000) {
28             echo "파일 사이즈가 너무 큼니다.";
29         }
30     }
```

```

28         $uploadOk = false;
29     }
30
31     // 파일 등록 및 복사
32     if ($uploadOk) {
33         if (move_uploaded_file($_FILES["upload"]["tmp_name"], $target_
34             file)) {
35             echo basename( $_FILES["upload"]["name"]). "를 업로드
36                 하였습니다.";
37         } else {
38             echo "파일 업로드 오류";
39         }
40     } else {
41         echo "죄송합니다. 파일 등록을 할 수 없습니다.";
42     }
43
44     echo "<br>=====<br>";
45
46     // HTML form을 출력합니다.
47
48     echo "<form name='jiny' method='post' enctype='multipart/form-data\"
49         action='\"".$_SERVER['PHP_SELF']."'>";
50
51     echo "<input type='file\" name='\"upload\" id='\"upload\"\">";
52     echo "<input type='submit' value='확인'>";
53     echo "</form>";
54
55     ?>

```

화면 출력



위의 파일 업로드 실험은 클라이언트의 웹 브라우저에서 서버로 파일을 전송합니다.

form의 file 요소로 파일을 선택하여 전송합니다. 서버의 PHP에서는 \$_FILES 배열 변수로 파일의 정보를 받아들일 수 있습니다.

\$_FILES 배열에는 파일의 관련된 다양한 값의 정보를 가지고 있습니다. 또한 전송받은 파일은 임시폴더 안에 저장됩니다. move_uploaded_file() 함수를 통하여 실제 저장 장소로 이동합니다.

파일을 이동 시에는 파일의 정보와 저장 디렉터리, 용량 등을 사전에 체크하여 필터링할 수 있습니다.

17.3 다운로드

서버상에 있는 파일을 사용자 PC에 다운로드할 수 있도록 링크를 생성할 수 있습니다. 서버상의 파일을 간단하게 다운로드하는 방법은 header() 함수를 통하여 브라우저에 header를 설정하여 전송하는 방법입니다.

| 내장 함수 |

```
void header ( string $string [, bool $replace = true [, int $http_response_code ]] )
```

내장 함수 header() 함수는 사용자 브라우저로 데이터를 전송합니다. header() 함수는 다른 HTML 코드보다 우선적으로 실행되어야 합니다.

예제 파일 | download.php

```
1  <?php
2
3      $server_filename= "./sample.csv";
4
5      // $server_filename에서 파일명 부분만 추출합니다.
6      $filename = substr($server_filename, strpos($server_filename, '/') + 1);
```

```

7
8 // 서버의 파일명을 확인합니다.
9 // 파일이 존재하지 않거나 읽을 수가 없을 때 처리를 중단합니다.
10 if (!file_exists($server_filename) || !is_readable($server_filename)) {
11     die("파일을 읽을 수 없습니다.");
12 } else {
13     // 파일 사이즈를 체크합니다.
14     if (($filesize = filesize($server_filename)) == 0) {
15         die("파일 크기가 0 바이트 입니다.\n");
16     } else {
17         if (($fp = @fopen($server_filename, 'rb')) === false) {
18             die("파일을 읽을 수 없습니다.\n");
19         }
20     }
21 }
22
23
24 // UTF-8 파일명이 깨지지 않도록 하기 위해서
25 // 파일명에 사용할 수 없는 문자를
26 // 모두 제거합니다. 또는 다른 안전한 문자로 치환합니다. (RFC2231/5987 표준)
27 $illegal = array('\', '/', '<', '>', '{', '}', ':', ';', '|', '"',
28     '~', '`', '@', '#', '$', '%', '^', '&', '*', '?');
29 $replace = array('_', '(', ')', '(', ')', '-', ',', '-', ' ', '-',
30     '\', '-', '-', '-', '-', '-', '-', '-', '-', '-');
31 $filename = str_replace($illegal, $replace, $filename);
32 $filename = preg_replace('/([\x00-\x1f\x7f\xff]+)/', '', $filename);
33
34 // 유니코드 공백 문자를 일반 공백 문자(0x20)로 변경을 합니다.
35 $filename = trim(preg_replace('/[\pZ\pC]+/u', ' ', $filename));
36
37 // 중복된 점, 대체 문자를 정리합니다.
38 $filename = trim($filename, '._-');
39 $filename = preg_replace('/__+/', '-', $filename);
40 if ($filename === '') {
41     die("파일명 이름 변환처리 실패");
42 }
43
44 // 브라우저의 User-Agent 값 처리
45 USER_AGENT = isset($_SERVER['HTTP_USER_AGENT']) ? $_SERVER['HTTP_
46     USER_AGENT'] : '';

```

```

45 $old_ie = (bool)preg_match('#MSIE [3-8]\.#', $USER_AGENT);
46
47 // 브라우저별 헤더 파일명을 정리합니다.
48 if (preg_match('/^[a-zA-Z0-9_-]+$/ ', $filename)) {
49     // 파일명에 숫자와 영문으로만 된 경우
50     // 브라우저 상관없이 파일명을 헤더로 처리를 합니다.
51     $header = 'filename="' . $filename . '"';
52
53 } elseif ($old_ie || preg_match('#Firefox/(\d+)\.#', $USER_AGENT,
54 $matches)
55     && $matches[1] < 5) {
56     // IE 9 미만 , Firefox 5 미만의 경우 처리.
57     $header = 'filename="' . rawurlencode($filename) . '"';
58
59 } elseif (preg_match('#Chrome/(\d+)\.#', $USER_AGENT, $matches)
60     && $matches[1] < 11) {
61     // Chrome 11 미만의 경우.
62     $header = 'filename=' . $filename;
63
64 } elseif (preg_match('#Safari/(\d+)\.#', $USER_AGENT, $matches)
65     && $matches[1] < 6) {
66     // Safari 6 미만의 경우.
67     $header = 'filename=' . $filename;
68
69 } elseif (preg_match('#Android #', $USER_AGENT, $matches)) {
70     // 안드로이드 브라우저의 경우.
71     $header = 'filename="' . $filename . '"';
72
73 } else {
74     // RFC2231/5987 표준 준수
75     $header = "filename*=UTF-8'" . rawurlencode($filename) .
76     ' ; filename="' . rawurlencode($filename) . '"';
77 }
78
79 // 캐시 처리
80 // 캐시 처리를 위해서 Cache-Control, Expires 등의 헤더를 적용합니다.
81 if (!$expires) {
82     // 캐싱 금지
83     if ($old_ie) {
84         // 익스플로러 8 이하 버전
85         // SSL 사용시 no-cache 및 pragma 헤더 오류를 발생

```



```

85         header('Cache-Control: private,
86             must-revalidate,
87             post-check=0,
88             pre-check=0');
89         header('Expires: Sat, 01 Jan 2000 00:00:00 GMT');
90     } else {
91         header('Cache-Control: no-store,
92             no-cache,
93             must-revalidate,
94             post-check=0,
95             pre-check=0');
96         header('Expires: Sat, 01 Jan 2000 00:00:00 GMT');
97     }
98
99     } else {
100         // 캐싱 허용
101         header('Cache-Control: max-age=' . (int)$expires);
102         header('Expires: ' . gmdate('D, d M Y H:i:s', time() + (int)$expires)
103             ' GMT');
104     }
105
106     // 다운로드 이어받기 헤더 처리
107     // Range 헤더 자동 감지, Accept-Ranges 헤더 자동 생성
108     if (isset($_SERVER['HTTP_RANGE']) && preg_match('/^bytes=(\d+)-/',
109         $_SERVER['HTTP_RANGE'], $matches)) {
110         $range_start = $matches[1];
111
112         if ($range_start < 0 || $range_start > $filesize) {
113             header('HTTP/1.1 416 Requested Range Not Satisfiable');
114             return false;
115         }
116
117         header('HTTP/1.1 206 Partial Content');
118         header('Content-Range: bytes ' . $range_start . '-' . ($filesize - 1) .
119             '/' . $filesize);
120         header('Content-Length: ' . ($filesize - $range_start));
121     } else {
122         $range_start = 0;
123         header('Content-Length: ' . $filesize);

```

```

123
124     }
125
126     // 그 외 헤더를 전송한다.
127     header('Accept-Ranges: bytes');
128     header('Content-Type: application/octet-stream');
129     header('Content-Disposition: attachment; ' . $header);
130
131     // 출력 버퍼를 초기화.
132     // 대용량 파일 다운로드 시 메모리 누수를 방지합니다.
133     while (ob_get_level()) {
134         ob_end_clean();
135     }
136
137     // 파일을 64KB 단위로 분할 전송
138     // readfile() 함수의 메모리 누수 방지
139     $block_size = 16 * 1024;
140     $speed_sleep = $speed_limit > 0 ? round(($block_size / $speed_limit /
141     1024) * 1000000) : 0;
142
143     $buffer = '';
144     if ($range_start > 0) {
145         fseek($fp, $range_start);
146         $alignment = (ceil($range_start / $block_size) * $block_size) -
147         $range_start;
148
149         if ($alignment > 0) {
150             $buffer = fread($fp, $alignment);
151
152             echo $buffer;
153             unset($buffer);
154             flush();
155         }
156     }
157
158     while (!feof($fp)) {
159         $buffer = fread($fp, $block_size);
160         echo $buffer;
161         unset($buffer);
162         flush();
163         usleep($speed_sleep);

```

```
162     }  
163  
164     fclose($fp);  
165  
166 ?>
```

