jinyPHP 3권 - 17.HTML Form

**17. HTML & FORM**

====================

HTML은 화면을 구성하는 태그들과 사용자의 입력을 받아 서버로 전송을 하는 FROM 구성요소로 크게 구분을 할 수 있습니다.

FROM 구성요소들은 서버로 데이터를 전송을 하는데, PHP는 이런 전송된 데이터를 받아서 처리를 할 수 있습니다.

**17.1 FORM**

====================

HTML의 FORM 태그는 폼요소들을 감싸고 있는 바디 부분과 그 안에 들어가는 요소들의 항목으로 구분을 할 수 있습니다.

**17.1.1 FORM 태그**

====================

HTML에서 <form></form>으로 마크업이 되어진 코드를 form 테그라고 합니다. 폼테그는 폼요소들을 포함하는 바디와 같습니다.

|  |
| --- |
| <form>  폼요소들...  </form> |

폼 요소들은 폼 태그로 둘러 쌓여져 있어야 합니다.

폼 태그는 몇개의 속성 설정을 같이 설정할 수 있습니다.

**name**

폼의 이름을 지정합니다.

**method**

매소드 속성은 입력된 데이터의 전송방식을 설정합니다. POST 방식 또는 GET 방식으로 지정할 수 있습니다. 만일 POST 방식으로 설정을 하였다면, 서버쪽 PHP에서는 슈퍼변수 $\_POST 배열을 통해서 데이터를 가지고 올 수 있습니다.

GET 방식으로 설정을 하였다면 슈퍼변수 $\_GET 배열을 통해서 데이터를 읽어 올 수 있습니다.

**action**

엑션 속성은 해당 폼을 처리하는 PHP 스크립트의 url과 파일명을 지정합니다. 폼이 서브밋 처리가 되면 액션에서 지정한 속성으로 페이지가 이동이 되면서 데이터를 처리합니다.

**enctype="multipart/form-data"**

파일 업로드와 같이 바이너리 데이터를 전송할 경우에 추가되는 속성입니다.

**17.1.2 Action**

====================

폼 태그의 속성중 action 은 서브밋 클릭시 처리되는 스크립트 페이지 입니다. 전형적인 HTML 코딩에서는 직접 url과 스크립트 명을 입력합니다.

|  |
| --- |
| <form method="post" action="**./test.php**"> |

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

이런경우 슈퍼변수 $\_SERVER["PHP\_SELF"] 값을 통하여 아래와 같이 편리하게 사용을 하기도 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.

acrion=”” 부분에 $\_SERVER["PHP\_SELF"]를 입력하는것은 서브밋 버튼 클릭시 다른 페이지로 이동하지 말고, 다시 자기 자신으로 다시 호출하여 데이터를 처리하라는 의미 입니다.

하지만 이렇게 편리하게 사용한 $\_SERVER["PHP\_SELF"] 테크닉은 요즘들어 보완적으로 문제가 발생되고 있습니다. 이런한 셀프페이지 방식은 동일한 페이지 form에서 오류 메시지를 출력하기 좋습니다.

웹페이지는 불특정 다수의 사람들에게 노출되는 특성이 있습니다. 그중에서는 정상적인 사용자 이외에 해커 및 블랙유저 또한 있을 것입니다. PHP의 서버 전역변수 $\_SERVER["PHP\_SELF"] 의 값은 해커의 이해 노출될 수 있습니다.

PHP\_SELF를 사용하면 해커들을 슬래시 (/)를 입력 한 후에 XSS (Cross Site Scripting) 명령을 입력하여 정상 실행을 방해할 수 있습니다. XSS (Cross-Site Scripting)는 웹 응용 프로그램에서 일반적으로 발견되는 컴퓨터 보안적 취약점 유형입니다. XSS를 사용하면 공격자가 다른 사용자가 본 웹 페이지에 클라이언트 측 스크립트를 삽입 할 수 있습니다.

예를들어 현재의 스크립트 test.php에서 폼 액션 값을 다음과 같이 설정하였다고 생각해 봅시다.

|  |
| --- |
| <form method="post" action="<?php echo $\_SERVER["PHP\_SELF"];?>"> |

이 페이지가 정상적으로 서비및을 하였다면 현재의 페이지 test.php 일것입니다.

|  |
| --- |
| <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 입력을 했다면 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 들어가 있습니다. 변경된 폼 요소는 자바크립트를 실행하고 페이지가 해킹이 될 수도 있습니다.

$\_SERVER["PHP\_SELF"]의 보안적인 부분 때문에 직접 변수를 사용하지 않고 htmlspecialchars() 함수를 같이 이용하여 처리를 합니다.

|  |
| --- |
| <form method="post" action="<?php echo **htmlspecialchars($\_SERVER["PHP\_SELF"]);**?>"> |

htmlspecialchars() 함수는 HTML의 특수문자들을 변환처리 해줍니다. 특수문자란 < , > 그리고 &lt; 와 &gt; 를 말합니다.

이런 특수문자는 HTML 또는 자바스크립트 코드등에서 인젝션 공격을 당할 수 있습니다. 또한 사이트가 위험에 노출 될 수 있습니다.

다시한번 예를 들어

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

htmlspecialchars() 함수를 같이 사용을 하면

<form method="post" action="test.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">

형태로 변경이 되기 때문에 좀더 안전한 실행이 가능할 것입니다.

**17.1.2 서브밋**

====================

서브밋은 form에서 입력된 데이터를 action에서 지정한 url로 전송을 처리하는 승인 버튼입니다. 즉, 클라이언트 브라우저에서 사용자 데이터 값을 입력 후에 서브밋 버튼을 클릭하면 서버 쪽으로 데이터를 전송합니다.

기본적으로 서브및을 지정하는 양식은 다음과 같습니다.

|  |
| --- |
| <input type='submit' value='확인'> |

input 태그로 타입을 **submit**으로 설정만 하면 됩니다. 또한 버튼의 이름을 value 속성값으로 설정을 하면 됩니다.

또한 서브밋 버튼을 클릭시 자바스크립트를 실행하여 처리를 할 수도 있습니다.

|  |
| --- |
| <input type='submit' value='확인' **onclick=\"javascript:check\_submit()\"** > |

서버로 데이터값을 전달하기 전에 자바스크립트를 통하여 필수항목이나 유효성등을 확인한 후에 처리를 할 수 있습니다.

타입 형식을 image로 설정할 경우 이미지 버튼으로 사용할 수 있습니다.

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

**17.1.3 리셋**

====================

리셋은 서브밋의 반대로 입력한 폼의 값들을 모두 초기화 합니다.

|  |
| --- |
| <input type='reset' value='지우기'> |

**17.2 요소**

====================

HTML은 폼 태그 안에 다양한 요소들을 삽입하여 데이터를 입력 받을 수 있습니다.

폼의 요소값을 입력받에 처리를 할때는 프로그램 동작오류를 줄이기 위해서 필수체크와 유효성 값을 같이 처리하는 것이 필요합니다.

**필수체크**

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

이런경우, 다음에 처리할 유효성 체크 부분과 정상적인 프로그램 로직 처리가 불가능해 질 수 도 있습니다. 물론, 폼을 처리하기전에 javascript 등으로 체크를 한다고 해도 외부에서 잘못된 경로로 처리 PHP로 접근하여 오류를 발행하거나, 해킹이 될수도 있을 것입니다.

따라서 폼으로 입력을 받은 데이터값 중에 반드시 필요로 하는 값등은 필수 체크를 중복으로 해주시는 것이 좋습니다.

값을 입력 받을때 다음과 같이 empty() 함수를 응용하여

if (empty($\_POST["name"])) {

참: 비어있습니다...

} else {

거짓: 비어 있지 않습니다...

}

패턴으로 필수체크를 해주시는 것이 좋습니다.

**유효성**

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

따라서 정확한 처리로직과 보안상을 위해서 반드시 데이터를 처리하기 전에 PHP 에서도 한번더 데이터를 검사하는 작업을 하는 것이 좋습니다.

PHP의 필터링 함수의 기능을 이용하면 좀더 복잡한 데이터의 유효성을 쉽게 확인을 할 수 있습니다.

**17.2.1 input 텍스트**

====================

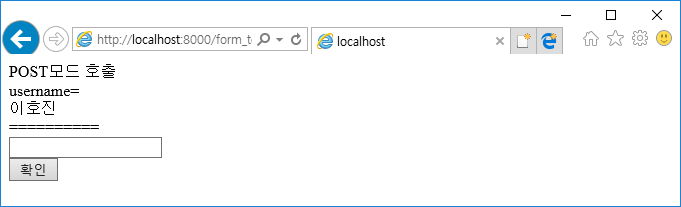
HTML의 input Text 요소는 폼 요소로 가장 사용 빈도가 높은 요소입니다. TEXT는 사용자로 부터 다양한 문자열을 입력을 받을 수 있습니다. 입력되는 값이 문자열일 수도 있고, 또한 문자를 가장한 숫자일 수도 있습니다. 그외에도 이메일, 이름, url등 모두 문자열에 속하기 때문에 다양한 입력값들이 들어 올 수 있습니다.

|  |
| --- |
| <input type="text" name="name"> |

**예제파일) form\_text.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  echo "username= <br>" . $\_POST['username'];  echo "<br>==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  echo "<input type=\"text\" name=\"username\">";  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";    ?> |

**화면출력)**



**다양한 타입속성**

HTML5의 input은 좀더 많은 다양한 타입속성을 지원합니다. 타입 속성을 이용하면 데이터 입력시 발생할 수 있는 오류들을 먼저 브라우저 차원에서 필터링 할 수 있습니다.

email

url

search

tel

number

date

datetime

month

week

time

range

color

**필수체크**

HTML5 에는 폼의 필수 입력을 확인하는 속성인 **required** 가 입니다. 하지만 이 속성은 브라우저에서만 동작하는 속성입니다. 브라우저 이외의 방식으로 접속을 할경우에는 필수체크를 할 수 없습니다.

이런경우 PHP 소스상에서 다시한번 필수 체크해 주는 것이 중요합니다.

**입력값 제한**

HTML5는 브라우저에서 입력 문자열의 길이를 제한할 수 있습니다. input 테그의 속성으로 maxlength=”값” 을 추가하시면 됩니다.

maxlength의 입력 제한 동작은 브라우저에서만 동작을 합니다. 브라우저를 제외한 방법으로 접속하여 데이터를 전송할때는 maxlength 적용을 받지 않습니다.

**입력패턴**

HTML5의 입력패턴 속성을 통하여 데이터 형식을 제한할 수 있습니다. 패턴 속성과 정규표현방법을 통하여 입력되는 데이터를 필터링 합니다.

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

**최대, 최소**

HTML 5에서는 min 또는 max 속성값을 통하여 입력되는 데이터의 최소치와 최대치를 지정할 수도 있습니다.

**초기값지정**

input 의 텍스트 속성을 사용시 사용자에게 보여주는 초기값 및 기본값을 미리 출력해서 보여줄 수 있습니다. 초기값은 약간 흐린 회색계열의 문자로 출력함으로서 사용자 입력값과 구분을 합니다.

|  |
| --- |
| <input type="text" name="username" **placeholder=”이름을 입력해 주세요”**> |

**자동완성 추천**

검색엔진이나 특정 값을 입력할때 하단에 추천하는 단어들을 나타내는 경우를 보신적이 있을 것입니다. HTML5의 autocomplete 기능을 이용하면 쉽게 구현을 하실 수 있습니다.

|  |
| --- |
| <input type="text" name="country" **placeholder=”국가를 입력 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 전송을 하는 값 이지만, 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.

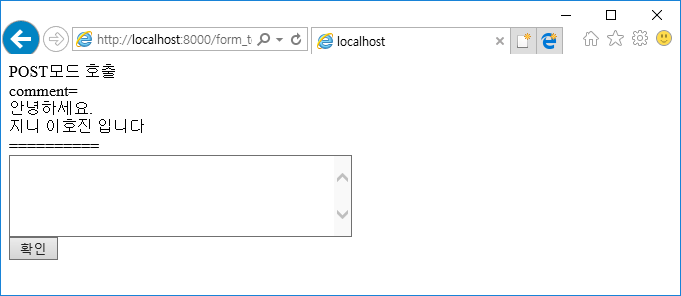
|  |
| --- |
| <textarea name="comment" rows="5" cols="40"></textarea> |

textarea는 웹페이지에서 글을 작성하는 본문등 많이 이용을 합니다. textarea는 많은 양과 다양한 입력을 고려하여 데이터안에 숨겨진 기호나 패턴등을 복합적으로 판별을 해야 합니다.

**예제파일) form\_textarea.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  // 다음줄 nl을 <br>로 변환하여 출력함.  echo "comment= <br>" . nl2br($\_POST['comment']);  echo "<br>==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  echo "<textarea name=\"comment\" rows=\"5\" cols=\"40\"></textarea>";  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";    ?> |

**화면출력)**

****

**17.2.4 select**

====================

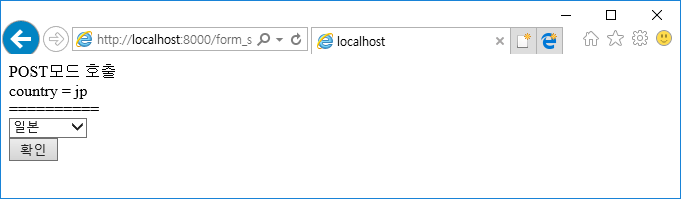
select 요소는 input 요소 다음으로 가장 많이 사용하는 html form 구성입니다. select는 드롭다운 형태로 여러개의 데이터 값중에 한개 또는 다수의 데이터를 선택할 수 있습니다.

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

**예제파일) form\_select1.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  echo "country = " . $\_POST['country'] ."<br>";  echo "==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  $body = "<select name=\"country\">";  $body .= "<option value=\"kr\">대한민국</option>";  $body .= "<option value=\"jp\">일본</option>";  $body .= "<option value=\"cn\">중국</option>";  $body .= "</select>";  // 기본 선택값을 selected 로 처리  if ($\_POST['country'])  $body = str\_replace(  "value=\"".$\_POST['country']."\"",  "value=\"".$\_POST['country']."\" selected",  $body);  echo $body;  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";    ?> |

**화면출력)**

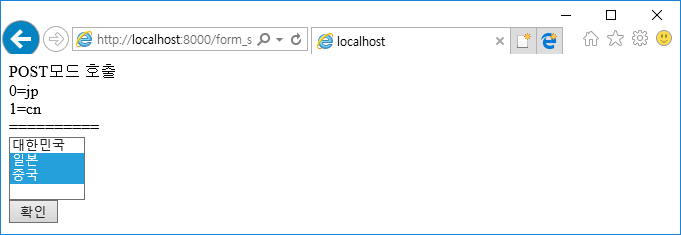


select는 기본적으로 선택한 한개의 값을 지정할 수 있습니다. 하지만 여러개의 항목을 선택이 필요한 경우에는 <select **multiple** name=”이름[]”> 형태로 이름을 **배열 형태**로, **multiple** 속성을 사용하면 됩니다.

**예제파일) form\_select2.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  $COUNTRY = $\_POST['country'];  for($i=0;$i<count($COUNTRY);$i++){  echo $i ."=". $COUNTRY[$i] ."<br>";  }  echo "==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  $body = "<select name=\"country[]\" multiple>";  $body .= "<option value=\"kr\">대한민국</option>";  $body .= "<option value=\"jp\">일본</option>";  $body .= "<option value=\"cn\">중국</option>";  $body .= "</select>";  // 기본 선택값을 selected 로 처리  if ($\_POST['country']){  $COUNTRY = $\_POST['country'];  for($i=0;$i<count($COUNTRY);$i++){  $body = str\_replace(  "value=\"".$COUNTRY[$i]."\"",  "value=\"".$COUNTRY[$i]."\" selected",  $body);  }  }    echo $body;  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";    ?> |

**화면출력)**



**17.2.5 checkbox**

====================

폼요소중 체크박스는 어떠한 항목에 동의등을 구할때 많이 사용을 하는 구성요소 입니다.

|  |
| --- |
| <input type='checkbox' name='이름'> |

위와 같이 HTML 코드입력하면 form의 데이터는 슈퍼변수 $\_GET[] 또는 $\_POST[] 배열 변수를 통해서 값을 가지고 올 수 있습니다.

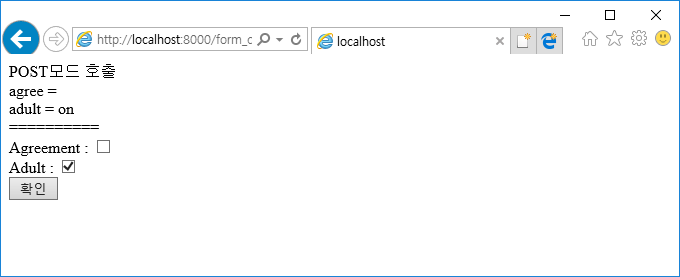
이때 지정한 name=”이름” 값의 배열의 선택 키값으로 사용이 됩니다. 만일 체크박스의 초기값을 “선택된” 으로 하고자 할때는 **checked** 키워드를 추가해 주면 됩니다.

체크박스가 선택된 값은 **on** 문자열 값으로 전달됩니다.

**예제파일) form\_check1.php**

|  |
| --- |
| <?php  // 서브 및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  echo "agree = " . $\_POST['agree'] ."<br>";  echo "adult = " . $\_POST['adult'] ."<br>";  echo "==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  echo "Agreement : ";  if ($\_POST['agree']){  echo "<input type='checkbox' name='agree' checked>";  } else {  echo "<input type='checkbox' name='agree'>";  }  echo "<br>";  echo "Adult : ";  if ($\_POST['adult']){  echo "<input type='checkbox' name='adult' checked>";  } else {  echo "<input type='checkbox' name='adult'>";  }  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";  ?> |

**화면출력)**

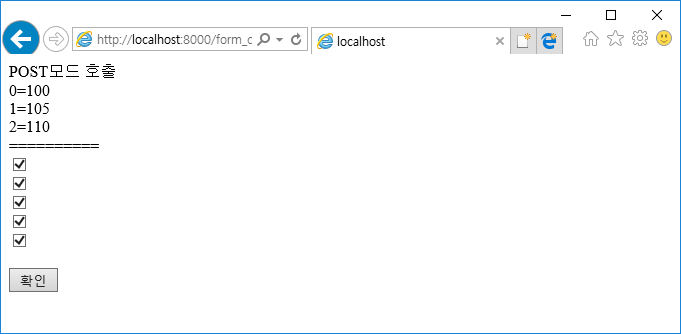


위의 실험1은 체크박스가 하나에 한개의 이름으로 각각의 할당되어 있습니다. 다음 예제는 체크박스 이름을 배열형태로 지정후에 value 속성을 이용하여 값을 할당합니다.

**예제파일) form\_check2.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  $TID = $\_POST['TID'];  for($i=0;$i<count($TID);$i++){  echo $i ."=". $TID[$i] ."<br>";  }  echo "==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  // 체크값의 이름이 배열형태로 지정  echo "<input type='checkbox' name='TID[]' value='100' checked> <br>";  echo "<input type='checkbox' name='TID[]' value='102' checked> <br>";  echo "<input type='checkbox' name='TID[]' value='105' checked> <br>";  echo "<input type='checkbox' name='TID[]' value='107' checked> <br>";  echo "<input type='checkbox' name='TID[]' value='110' checked> <br>";  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";  ?> |

**화면출력)**



실험2에서 체크박스는 TID라는 이름의 배열로 작성되어 있습니다. 또한 각각의 체크박스는 value 속성을 통하여 고유 값들을 가지고 있습니다.

서브및 을 통하여 PHP로 값이 전달되어진 경우, 배열로 체크박스는 배열 형태로 값을 전달 받게 됩니다. 선택한 체크박스의 값만을 배열화 하여 전달합니다.

이러한 유형은 리스트목록등에서 여러개의 데이터를 선택하여 FORM 으로 전달하는 용도로 많이 사용을 합니다.

**17.2.6 radio box**

====================

HTML의 라디오 요소는 여러개의 선택값중에 하나만 입력을 받을 수 있는 요소입니다.

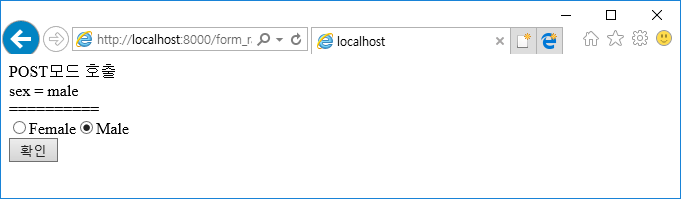
|  |
| --- |
| <input type="radio" name="sex" value="female">Female  <input type="radio" name="sex" value="male">Male |

라디오 버튼의 name=”이름” 부분은 동일한 값을 가집니다. 하지만 속성값은 각각의 다른 값을 가지게 됩니다.

**예제파일) form\_radio.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  echo "sex = " . $\_POST['sex'] ."<br>";  echo "==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' action='".$\_SERVER['PHP\_SELF']."'>";  if ($\_POST['sex'] == "female"){  echo "<input type=\"radio\" name=\"sex\" value=\"female\" checked>Female";  echo "<input type=\"radio\" name=\"sex\" value=\"male\">Male";  } else if ($\_POST['sex'] == "male"){  echo "<input type=\"radio\" name=\"sex\" value=\"female\">Female";  echo "<input type=\"radio\" name=\"sex\" value=\"male\" checked>Male";    } else {  echo "<input type=\"radio\" name=\"sex\" value=\"female\">Female";  echo "<input type=\"radio\" name=\"sex\" value=\"male\">Male";  }  echo "<br>";  echo "<input type='submit' value='확인'>";  echo "</form>";    ?> |

**화면출력)**



**17.2.7 file**

====================

메일 작성이나 계시판에 글과 함께 첨부하는 파일등을 지정하여 서버로 전송을 할 수 있습니다. 클라이언트 브라우저에서 서버로 파일을 전송하기 위해서는 form 의 type=”file”을 사용하면 됩니다.

|  |
| --- |
| <input type="**file**" name="upload"> |

HTML에서 파일전송을 위해서는 전송 매소드 타입이 POST로 되어 있어야 합니다. 또한, 바이너리 데이터를 같이 전송하기 위해서 속성 enctype="**multipart/form-data**" 가 같이 설정되어야 합니다.

HTML의 file요소를 선택하시면, 컴퓨터는 서버로 전송할 파일을 선택하는 팝업창을 브라우저가 실행을 시킵니다.

또한 PHP에서 파일을 업로드하여 전송을 하기 위해서는 약간의 php.ini 환경설정이 필요로 힙니다.

php.ini 파일에서 file\_uploads 부분을 ON 으로 변경을 해주어야 합니다.

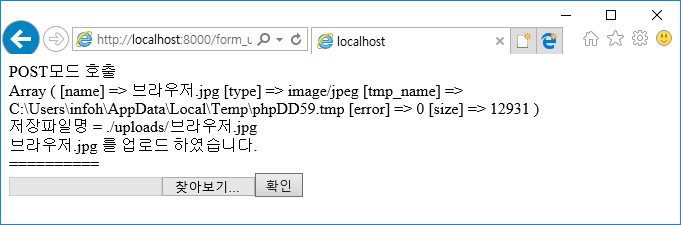
**file\_uploads = On**

또한 파일 용량도 제한도 가능합니다. 보다 자세한 것은 환경설정 부분을 참조해 주세요.

**예제파일) form\_upload.php**

|  |
| --- |
| <?php  // 서브및 처리시 데이터값 확인  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  echo "POST모드 호출 <br>";  print\_r($\_FILES['upload']);  echo "<br>";  // 업로드 디렉토리 지정  $target\_dir = "./uploads";  if(!is\_dir($target\_dir)){  mkdir($target\_dir);  }  $target\_file = $target\_dir . "/". basename($\_FILES["upload"]["name"]);  echo "저장파일명 = ". $target\_file ."<br>";  $uploadOk = true;  // 업로드 할 파일이 존재하는지 검사  if (file\_exists($target\_file)) {  echo "이미 존재하는 파일명입니다.";  $uploadOk = false;  }  // 파일 사이즈 제한  if ($\_FILES["upload"]["size"] > 500000) {  echo "파일 사이즈가 너무 큽니다.";  $uploadOk = false;  }  // 파일 등록 및 복사  if ($uploadOk) {  if (move\_uploaded\_file($\_FILES["upload"]["tmp\_name"], $target\_file)) {  echo basename( $\_FILES["upload"]["name"]). " 를 업로드 하였습니다.";  } else {  echo "파일 업로드 오류";  }  } else {  echo "죄송합니다. 파일등록을 할 수 없습니다.";  }    echo "<br>==========<br>";  }  // HTML form을 출력합니다.  echo "<form name='jiny' method='post' enctype=\"multipart/form-data\" action='".$\_SERVER['PHP\_SELF']."'>";  echo "<input type=\"file\" name=\"upload\" id=\"upload\">";  echo "<input type='submit' value='확인'>";  echo "</form>";  ?> |

**화면출력)**

****

위의 파일 업로드 실험은 클라이언트의 웹브라우저에서 서버로 파일을 전송합니다.

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

$\_FILES 배열에는 파일의 관련된 다양한 값의 정보를 가지고 있습니다. 또한 전송받는 파일은 임시폴더 안에 저장이 됩니다. move\_uploaded\_file() 함수를 통하여 실제 저장 장소로 이등을 합니다.

파일을 이동시에는 파일의 정보와 저장 디렉토리, 용량등을 사전에 체크하여 필터링할 수 있습니다.

**17.3 다운로드**

====================

서버상에 있는 파일을 사용자 PC에 다운로드 할 수 있도록 링크를 생성할 수 있습니다. 서버상의 파일을 간단하게 다운로드 받는 방법은 header() 함수를 통하여 브라우저에 header 를 설정하여 전송하는 방법입니다.

**|내부함수|**

|  |
| --- |
| void **header** ( string $string [, bool $replace = true [, int $http\_response\_code ]] ) |

내부함수 header() 함수는 사용자 브라우저로 데이터를 전송합니다. header() 함수는 다른 HTML 코드보다 우선적으로 실행이 되어야 합니다.

**예제) download.php**

|  |
| --- |
| <?php  **$server\_filename= "./sample.csv";**  // $server\_filenam에서 파일명 부분만 추출합니다.  $filename = substr($server\_filename, strrpos($server\_filename,'/') + 1);  // 서버의 파일명을 확인합니다.  // 파일이 존재하지 않거나 읽을 수가 없을때 처리를 중단합니다.  if (!file\_exists($server\_filename) || !is\_readable($server\_filename)) {  die("파일을 읽을 수 없습니다.");  } else {  // 파일 사이즈를 체크합니다.  if (($filesize = filesize($server\_filename)) == 0) {  die("파일크기가 0 바이트 입니다.\n");  } else {  if (($fp = @fopen($server\_filename, 'rb')) === false) {  die("파일을 읽을 수 없습니다.\n");  }  }  }  // UTF-8 파일명이 깨지지 않도록 하기 위해서  // 파일명에 사용할 수 없는 문자를  // 모두 제거합니다. 또는 다른 안전한 문자로 치환합니다.(RFC2231/5987 표준)  $illegal = array('\\', '/', '<', '>', '{', '}', ':', ';', '|', '"', '~', '`', '@', '#', '$', '%', '^', '&', '\*', '?');  $replace = array('', '', '(', ')', '(', ')', '\_', ',', '\_', '', '\_', '\'', '\_', '\_', '\_', '\_', '\_', '\_', '', '');  $filename = str\_replace($illegal, $replace, $filename);  $filename = preg\_replace('/([\\x00-\\x1f\\x7f\\xff]+)/', '', $filename);  // 유니코드 공백 문자를 일반 공백 문자(0x20)로 변경을 합니다.  $filename = trim(preg\_replace('/[\\pZ\\pC]+/u', ' ', $filename));  // 중복된 점, 대체문자를 정리합니다.  $filename = trim($filename, ' .-\_');  $filename = preg\_replace('/\_\_+/', '\_', $filename);  if ($filename === '') {  die("파일명 이름 변환처리 실패");  }  // 브라우저의 User-Agent 값 처리  USER\_AGENT = isset($\_SERVER['HTTP\_USER\_AGENT']) ? $\_SERVER['HTTP\_USER\_AGENT'] : '';  $old\_ie = (bool)preg\_match('#MSIE [3-8]\.#', $USER\_AGENT);  // 브라우저별 해더 파일명을 정리합니다.  if (preg\_match('/^[a-zA-Z0-9\_.-]+$/', $filename)) {  // 파일명에 숫자와 영문으로만 된경우  // 브라우저 상관없이 파일명을 해더로 처리를 합니다.  $header = 'filename="' . $filename . '"';  } elseif ($old\_ie || preg\_match('#Firefox/(\d+)\.#', $USER\_AGENT, $matches)  && $matches[1] < 5) {  // IE 9 미만 , Firefox 5 미만의 경우 처리.  $header = 'filename="' . rawurlencode($filename) . '"';    } elseif (preg\_match('#Chrome/(\d+)\.#', $USER\_AGENT, $matches)  && $matches[1] < 11) {  // Chrome 11 미만의 경우.  $header = 'filename=' . $filename;  } elseif (preg\_match('#Safari/(\d+)\.#', $USER\_AGENT, $matches)  && $matches[1] < 6) {  // Safari 6 미만의 경우.  $header = 'filename=' . $filename;  } elseif (preg\_match('#Android #', $USER\_AGENT, $matches)) {  // 안드로이드 브라우저의 경우.  $header = 'filename="' . $filename . '"';  } else {  // RFC2231/5987 표준 준수  $header = "filename\*=UTF-8''" . rawurlencode($filename) .  '; filename="' . rawurlencode($filename) . '"';  }  // 캐쉬처리  // 캐쉬 처리를 위해서 Cache-Control, Expires 등의 헤더를 적용합니다.  if (!$expires) {  // 캐싱 금지  if ($old\_ie) {  // 익스플로러 8 이하 버전  // SSL 사용시 no-cache 및 pragma 헤더 오류를 발생  header('Cache-Control: private,  must-revalidate,  post-check=0,  pre-check=0');  header('Expires: Sat, 01 Jan 2000 00:00:00 GMT');  } else {  header('Cache-Control: no-store,  no-cache,  must-revalidate,  post-check=0,  pre-check=0');  header('Expires: Sat, 01 Jan 2000 00:00:00 GMT');  }  } else {  // 캐싱 허용  header('Cache-Control: max-age=' . (int)$expires);  header('Expires: ' . gmdate('D, d M Y H:i:s', time() + (int)$expires) . ' GMT');  }  // 다운로드 이어받기 해더처리  // Range 헤더 자동 감지, Accept-Ranges 헤더 자동 생성  if (isset($\_SERVER['HTTP\_RANGE']) && preg\_match('/^bytes=(\d+)-/', $\_SERVER['HTTP\_RANGE'], $matches)) {  $range\_start = $matches[1];  if ($range\_start < 0 || $range\_start > $filesize) {  header('HTTP/1.1 416 Requested Range Not Satisfiable');  return false;  }  header('HTTP/1.1 206 Partial Content');  header('Content-Range: bytes ' . $range\_start . '-' . ($filesize - 1) . '/' . $filesize);  header('Content-Length: ' . ($filesize - $range\_start));  } else {  $range\_start = 0;  header('Content-Length: ' . $filesize);  }    // 그외 헤더를 전송한다.  header('Accept-Ranges: bytes');  header('Content-Type: application/octet-stream');  header('Content-Disposition: attachment; ' . $header);  // 출력 버퍼를 초기화.  // 대용량 파일 다운로드시 메모리 누수를 방지합니다.  while (ob\_get\_level()) {  ob\_end\_clean();  }  // 파일을 64KB 단위로 분할 전송  // readfile() 함수의 메모리 누수 방지  $block\_size = 16 \* 1024;  $speed\_sleep = $speed\_limit > 0 ? round(($block\_size / $speed\_limit / 1024) \* 1000000) : 0;  $buffer = '';  if ($range\_start > 0) {  fseek($fp, $range\_start);  $alignment = (ceil($range\_start / $block\_size) \* $block\_size) - $range\_start;  if ($alignment > 0) {  $buffer = fread($fp, $alignment);  echo $buffer;  unset($buffer);  flush();  }  }  while (!feof($fp)) {  $buffer = fread($fp, $block\_size);  echo $buffer;  unset($buffer);  flush();  usleep($speed\_sleep);  }  fclose($fp);    ?> |