

Тема: «Работа со сложными формами»

Цель: Научиться обрабатывать данные с помощью сложных элементов форм (checkbox, radio, select)

Теоретическая часть

Работа с флажками (checkbox)

Флажок – это элемент управления, позволяющий представлять логическое значение в HTML форме, находясь в установленном или снятом состоянии. Синтаксис элемента управления выглядит так:

```
<input type="checkbox">
```

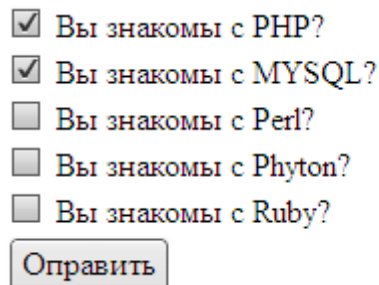
Помимо атрибута **type** флажок поддерживает атрибут **name** для уникального имени элемента управления, атрибут **value** для его значения и атрибут **checked**, наличие которого означает, что флажок установлен.

Создадим форму, содержащую пять флажков, два из которых являются предустановленными.

Пример 1.

```
<form method = 'post' action = 'handler.php'>  
<input type = 'checkbox' name = 'php' checked> Вы знакомы с PHP?<br>  
<input type = 'checkbox' name = 'mysql' checked> Вы знакомы с MySQL?<br>  
<input type = 'checkbox' name = 'perl' > Вы знакомы с Perl?<br>  
<input type = 'checkbox' name = 'python' > Вы знакомы с Python?<br>  
<input type = 'checkbox' name = 'ruby' > Вы знакомы с Ruby?<br>  
<input type = 'submit' value = 'Отправить'>
```

Внешний вид на рис. 1.



The image shows a web form with five checkboxes and a submit button. The checkboxes are arranged vertically, each followed by a question: 'Вы знакомы с PHP?', 'Вы знакомы с MySQL?', 'Вы знакомы с Perl?', 'Вы знакомы с Python?', and 'Вы знакомы с Ruby?'. The first two checkboxes are checked, while the others are unchecked. Below the checkboxes is a button labeled 'Отправить' (Submit).

Рис. 1. Использование флажков

При обработке данных в суперглобальный массив `$_POST` попадают только те элементы флажка, которые были отмечены.

Можно представить флажки в другой записи:

Работа со списками

Список позволяет выбрать одно или несколько значение из определенного набора и имеет следующий синтаксис.

Пример 2.

```
<select >  
  <option> Первый пункт</option>  
  <option>Второй пункт</option>  
  <option>Третий пункт</option>  
</select>
```

Между тегами `<select>` и `</select>` располагаются пункты списка, которые оформляются в виде `option` – тегов. В приведенном примере список имеет три пункта. Тег `<select>` может иметь атрибуты **multiple** и **size**. Параметр **multiple** позволяет выбрать несколько пунктов, отмечая их правой кнопкой мыши при одновременном удерживании клавиши `<Ctrl>`. Атрибут **size** определяет высоту списка в пунктах. Тег `<option>` может иметь атрибут **selected** для обозначения выделения текущего пункта.

Следующая форма содержит два выпадающих списка: первый – множественный список, который использует атрибут **multiple** и имеет высоту, равную 3, второй список является выпадающим и позволяет выбрать только одно значение.

Пример 3.

```

<form method = "post" action = "handler.php">
<table>
<tr valing = "top">
  <td> Выбор<br> нескольких значений</td>
  <td>
    <select name = "fst[ ]" multiple size = "3">
      <option value ="1" selected> Первый пункт</option>
      <option value ="2" selected> Второй пункт</option>
      <option value ="3" selected> Третий пункт</option>
    </select>
  </td>
</tr>
<tr valing = "top">
  <td> Одно значение</td>
  <td>
    <select name = "snd" >
      <option value ="1" selected> Первый пункт</option>
      <option value ="2" selected> Второй пункт</option>
      <option value ="3" selected> Третий пункт</option>
    </select>
  </td>
</tr>
<td></td>
  <td> <input type = "submit" value = "Отправить"></td>
</tr>
</table>
</form>

```

Внешний вид на рис. 2.

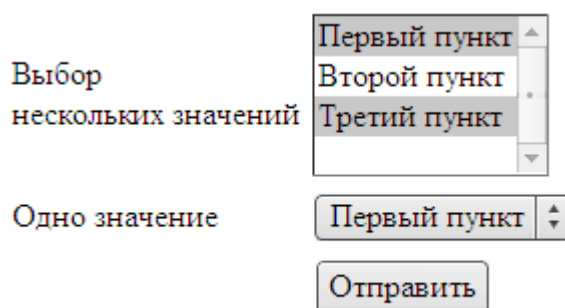


Рис. 2. Множественный и одиночный выпадающие списки

Если в качестве обработчика использовать следующий скрипт

```
<?php
```

```

echo "<pre>";
print_r($_POST);
echo "<pre>";
?>

```

то вывод данных из формы будет иметь вид:

```

[fst] => Array
(
    [0] => 1
    [1] => 3
)
[snd] => 1
}

```

Работа с переключателями

Переключатель, или иначе, радиокнопка – элемент управления, позволяющий выбрать из набора утверждений только одно. Он имеет следующий синтаксис:

```
<input type = "radio">
```

Для формирования набора утверждений используется несколько переключателей, которым присваивается одно и то же имя через атрибут name. Кроме этого переключатели могут иметь атрибут value для передачи значения и атрибут checked для того, чтобы отметить один из переключателей по умолчанию.

В следующем примере (4) переключатель используется для оценки сайта по пятибалльной системе.

Пример 4.

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

Оцените сайт

```

<input type = "radio" name = "mark" value = "1"> 1
<input type = "radio" name = "mark" value = "2"> 2
<input type = "radio" name = "mark" value = "3" checked>3
<input type = "radio" name = "mark" value = "4"> 4
<input type = "radio" name = "mark" value = "5"> 5
<input type = "submit" value = "Отправить">
</form>

```

Внешний вид на рисунке 3.

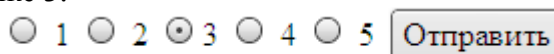


Рис. 3. Переключатели

Если в качестве обработчика использовать скрипт

```

<?php
echo "<pre>";
print_r($_POST);
echo "<pre>";
?>

```

то результатом отправки данных обработчику будет вывод следующего массива:

```

Array(
    [mark] => 3
)

```

Сохранение данных в форме после отправки

Практическая часть

Задание 1.

Работа с флажками (Checkbox)

1. Спросите у пользователя имя с помощью формы. Сделайте чекбокс: если он отмечен, то поприветствуйте пользователя, если не отмечен — попрощайтесь с пользователем.

2. Спросите у пользователя, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь. Если пользователь не отметил ни один язык — выведите на экран сообщение об этом.

Задание 2.

Работа с радиокнопками

1. Спросите у пользователя знает ли он PHP с помощью двух radio-кнопок. Выведите результат на экран. Сделайте так, чтобы по умолчанию один из вариантов был уже отмечен.

2. Спросите у пользователя его возраст с помощью нескольких radio-кнопок. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.

Задание 3.

Работа со списками

1. Спросите у пользователя его возраст с помощью select. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.

2. Спросите у пользователя с помощью мультиселекта, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.

Задание 4.

Дополнительные задания

1. Сделайте функцию, которая создает инпут type text или password. Функция должна иметь следующие параметры: type, name, value, placeholder.

2. Сделайте функцию, которая создает textarea. Функция должна иметь следующие параметры: name, value, placeholder.

3. Сделайте функцию, которая создает чекбокс. Если чекбокс не отмечен — функция должна отправлять 0 (то есть нужно сделать hidden инпут), если отмечен — 1.

4. Сделайте функцию, которая будет создавать селект. Функция должна принимать многомерный массив такого типа:

```
$arr = array(
    0=>array('value'=>'php', 'text'=>'Язык PHP'),
    0=>array('value'=>'html', 'text'=>'Язык HTML'),
)
```