

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 4

ТЕМА: Действие с формуляри

ЦЕЛ:

Целта на упражнението е студентите да се запознаят, как да създават формуляри и как PHP взаимодейства с тях. След упражнението студентите би следвало да могат да създават формуляри от различни типове и да обработват данните чрез PHP скрипт.

!ВАЖНО: За да можете да тествате упражнението ви е необходим следния софтуер. Web server, Php интерпретатор, MYSQL база данни. Всеки един от тези софтуери ги има в пакетите WAMP или XAMP. Настоящото упражнение е тествано върху WAMP Version 3.2.0.

I. ТЕОРЕТИЧНА ЧАСТ

В упражненията по дисциплината за създаването на формуляри ще използваме HTML документи. Както бе споменато в първо упражнение PHP се вгражда лесно в HTML документи и това дава възможност за гъвкавото взаимодействие между потребителя и сървъра.

За създаването на формуляри се използва елемента „form“, който се състои от отварящ и затварящ таг и обхваща всички останали елементи на формуляра:

```
<form action="име_на_файл" method="тип">  
... ..  
</form>
```

Началният таг на елемента „form“ съдържа два атрибута, които определят действието, което трябва да се извърши при изпращане на скрипт, както и метода, според който данните да бъдат изпратени към скрипта. PHP поддържа метод „post“, затова и това е метода който ще използваме в нашите формуляри.

В елемента „form“ се поставят редица други елементи, които изграждат самата структура на формуляра. Някои от най често използваните са:

- за етикети;
- за текстови полета;
- за радиобутони;
- за полета за отметка;
- за списъци;
- за многоредови текстови полета;
- за полета от тип изображение ;

Етикетът се поставя чрез таг „label“ и се използва най – често за поставянето на етикети за текстовите полета. Препращането на етикета към съответното текстово поле става чрез атрибут „for“ на таг „label“. Синтаксиса му е следния:

```
<label for="идентификатор_на_полето">Етикет</label>
```

За създаването на текстови полета се използва таг „*input*“ с задължителен атрибут „*type*“, който трябва да приема стойност „*text*“. Към тага биха могли да се добавят и други препоръчителни атрибути като „*name*“ и „*id*“. Синтаксисът на таг „*input*“ е следния:

```
<input type="тип" name="име_на_полето" id="идентификатор_на_полето"/>
```

Чрез атрибут „*id*“ представлява идентификатор към който препраща елементът „*label*“. Чрез този идентификатор съответния етикет се свързва към конкретното поле. Атрибут „*name*“ служи за уникално име на въведеното поле. То се използва при подаването на информация към скрипта. Чрез атрибут „*type*“ се разбира какъв е типа на полето. Някои от най – често използваните са:

- *text* – за създаване на текстово поле;
- *password* – за създаване на текстово поле, чиято информация се скрива със определен знак при въвеждането ѝ;
- *radio* – за създаването на радиобутони;
- *checkbox* – за създаването на полета за отметка;
- *image* – за създаването на полета от тип изображение;
- *hidden* – за създаването на скрито текстово поле. Това поле е невидимо за потребителя;
- *submit* – за създаването на бутон за изпращане.

Типът „*radio*“ на атрибут „*type*“ за тага „*input*“ създава друг вид поле във формуляра, с което потребителя може да взаимодейства. То не позволява въвеждането на текст, а предлага на потребителя няколко радиобутона, от които може да направи своя избор. В даден момент само един от няколко групирани заедно радиобута може да бъде избран. За да предостави информация на потребителя от радиобутоните се използва атрибут „*value*“. Важно е да се отбележи че групирани радиобутони трябва да имат един и същи атрибут „*name*“. Така формуляра ще знае кои от тях формират една група.

Типът „*checkbox*“ на атрибут „*type*“ създава полета за отметка. Те позволяват на потребителя да избере едновременно произволен брой от тях. Всяко поле за отметката трябва да притежава уникален атрибут „*name*“.

Типът „*image*“ на атрибут „*type*“ за тага „*input*“ позволява на формулярите да обработват изображения като входни данни. В такъв случай към тага „*input*“ трябва да се добави още един атрибут „*src*“ в който се посочва файла със самото изображение. В този случай „*input*“ би имал следния синтаксис:

```
<input type='image' src='файл_с_изображение' name='име'/>
```

Когато с мишката се кликне върху самото изображение, елементът „*image*“ връща координатите на мястото което е посочено. Координатите се получават като две отделни стойности с имена „*име_x*“ и „*име_y*“, където „*име*“ е стойността на атрибута „*name*“.

Типът „*hidden*“ на атрибут „*type*“ за таг „*input*“ предоставя възможност на програмиста да се предават данни които потребителя да не вижда. Тук може би възниква въпросът защо е необходимо това? Възможно е програмиста да иска чрез формуляр да предаде информация от един скрипт към друг или отново към същия скрипт, но така че потребителя да не може да види или оперира с предаваната информация. Примерно за подаването на така информация е, когато е необходимо скриптът да отчита колко пъти е натиснат бутона за изпращане на формуляра. Това би могло да стане чрез скрито поле, чиято стойност трябва да се подаде към атрибут „*value*“.

Тип „*submit*“, позволява данните от формуляра да бъдат изпратени към сървър. Този елемент се изобразява на екрана във вид на бутон, който може да бъде натиснат. При натискане на бутона информацията се изпраща към скриптовия файл посочен в атрибут

“action” на тага “form”. За прочете подадената информация от формуляра се използва метода на предаване. Като бе споменато по-рано PHP поддържа метод “post”, затова и него ще използваме в нашите формуляри.

Когато формулярът бъде създаден, към него се създава асоциативен масив от променливи. Той се нарича \$_POST. Всеки елемент от даден HTML формуляр, който притежава уникално име (атрибут „name“ на таговете), се съхранява в този масив и може да бъде достигнат от скрипт на PHP. Синтаксисът е следният:

`$_POST[‘име’]`

На мястото на „име“ се поставя стойността на атрибута „name“ за елемента който искаме да достигнем.

PHP скриптът обработващ данните подадени от формуляра, може да бъде отделно от html документа. Но в такъв случай ако се наложи евентуално редактиране на формуляра би се получило затруднение. Както се запознахме в първото упражнение, PHP може да бъде вграден и в самия html документ, т.е. скриптът и формуляра да бъдат в един общ файл. Тогава когато бутонът за изпращане на формуляра бъде натиснат той трябва да вика сам себе си. Това може да стане като в атрибут action на елемента „form“ се посочи името на същия файл в който е записан самия html документ и скрипт. Това би затруднило програмиста ако в процеса на разработка той поради някаква причина преименува самия файл. Затова в PHP съществува сървърна променлива (някои от тези променливи са споменати в 1-во упражнение), която съдържа името на текущия скрипт. Това е:

`$_SERVER[“PHP_SELF”]`

Ако използвайки команда echo вмъкнем нейната стойност в елемента „form“, това ще гарантира че скриптът винаги ще извика себе си, независимо от името, под което се съхранява. В този случай заглавния ред на елементът “form” би изглеждал по следния начин:

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

Когато PHP скриптът и формуляра са в един файл при първото зареждане на страница би се изпълнил и самия скрипт, който би извел на екрана незавършени изречения или предупреждения, че изброените променливи (в предупрежденията) все още не са създадени. Обикновено тези предупреждения са изключени и е възможно да не се покажат. Допускането на подобни проблеми обаче не говори добре за стила на програмиране и затова е необходимо да се направи проверка дали данните от формуляра вече са били изпратени или не. Това би могло да стане като целия PHP скрипт се постави в конструкция “if” която да извършва съответната проверка. За целта би могло да се използва функцията “isset()” (описана в първо упражнение) по следния начин:

`if(isset($_POST[“име_на_бутона”])){
.....}`

За „име_на_бутона“ се поставя уникалното име от атрибут „name“ на бутона за изпращане. Така се получава сигурност че PHP скрипта би се изпълнил единствено ако бутона е натиснат.

За създаването на елементи от тип списъци се използват двата тага “select” и “option”, като вторият е вграден в първия. Създават се толкова на брой тагове “option” колкото искаме да е голям списъка. Формулировката на конструкцията е следната:

`<select name='име'>
 <option value='стойност'>съдържание</option>
`

</select>

Атрибутът „name“ в тага „select“ е уникалното име за този елемент, а атрибутът „value“ на тага „option“ съдържа избраната стойност за съответния ред от списъка. Тя ще се изпрати към сървъра, при евентуално нейно избиране от потребителя. „Съдържание“ е стойността което потребителя вижда на екрана за съответния ред от списъка.

За създаването на многоредови текстови полета се използва тага „textarea“. Той има следната формулировка:

```
<textarea name='име' rows='брой_редове' cols='брой_колони'>
</textarea>
```

Атрибутите в тага не са задължителни. „Name“ отново представлява уникално име на за полето, а атрибутите „rows“ и „cols“ съдържат броя на редовете и броя на колоните които да съдържа това поле.

II. ПРАКТИЧЕСКА ЧАСТ

В практическата част е показан пример, чрез които се демонстрира взаимодействието на формуляри с PHP скриптове.

ЗАДАЧА1: Да се създаде форма предоставяща възможност за регистрация на потребител чрез въвеждането на име, фамилия, потребителско име и парола. Също така нека чрез формата на падащ списък потребителя да избере своята рождена дата. Във формуляра да се предвиди възможност и за избор на пол. При натискането на бутон за изпращане да се извиква PHP скрипта от същия файл който извежда поздравително съобщение за новия потребител и съобщава какво е неговото потребителско име и парола:

КОД:

```
<html>
<body>
<h2> Please enter your personal details:</h2>
<form action='<?php echo $_SERVER["PHP_SELF"]?'>' method='post'>
<p>
<label for="strFirstName">Firstname</label>
<input type="text" name="strFirstName" id="strFirstName"/>
</p>
<p>
<label for="strSurName">Surname</label>
<input type="text" name="strSurName" id="strSurName"/>
</p>
<p>
<label for="strUserName">Username</label>
<input type="text" name="strUserName" id="strUserName"/>
</p>
<p>
<label for="strPassword">Password</label>
<input type="password" name="strPassword" id="strPassword"/>
</p>
<p>
```

```

<label for="date">My Birthday is</label>
<select id="date">
    <?php    for($i=1;$i<32;$i++){        ?>
        <option value="<?php echo $i;?>"><?php echo $i;?></option>
    <?php    }        ?>
</select>
<select id="date">
    <?php    for($i=1;$i<13;$i++){        ?>
        <option value="<?php echo $i;?>"><?php echo $i;?></option>
    <?php    }        ?>
</select>
<select id="date">
    <?php    for($i=2015;$i<2100;$i++){        ?>
        <option value="<?php echo $i;?>"><?php echo $i;?></option>
    <?php    }        ?>
</select>
</p>
<p>
    <label for="gender">My gender is</label>
    <input type="radio" name="gender" id="gender" value="male"/>
    <input type="radio" name="gender" id="gender" value="female"/>
</p>
<p>
    <input type="submit" name="Submit"/>
</p>
</form>
</body>
</html>
<?php
if(isset($_POST['Submit'])){
    $strFirstName=$_POST['strFirstName'];
    $strSurName=$_POST['strSurName'];
    $strUserName=$_POST['strUserName'];
    $strPassword=$_POST['strPassword'];
    echo "<p>Greetings $strFirstName $strSurName</p>";
    echo "<p>Your username is $strUserName and your password is $strPassword</p>";
}
?>

```

ПОЯСНЕНИЕ

Горния пример създава регистрационен формуляр който в отделни параграфи са отделени двойките елементи „input“ и „label“ създаващи съответните текстови полета и етикети към тях. В текстовите полета потребителят може да напише своите данни. Чрез елемента „select“ и вложения в него елемент „option“ се създават падащ списък чрез който потребителя да избере своята рождена дата, месец и година. За да не е необходимо да поставяме доста на брой елементи „option“ той е конструиран в един вграден php скрипт който създава цикъл „for“. Чрез него може да се поставят толкова на брой „option“ елемента колкото са необходими, а именно за дата 31 на брой, за месец 12 на

брой и за година е предвидено от 2015 до 2100 година. Чрез елемент *“input”* с атрибут *„type=radio“* се създават два радио бутони чрез които потребителят може да избере своя пол. Обърнете внимание че и двата радио бутона имат едно и също име в атрибут *“name”*.

При натискане на бутона за изпращане информацията се подава на скрипта от същия файл, благодарение на системната променлива на PHP вградена в таг *„form“* за атрибут *„action“*. При изпълнението на скрипта първо се проверява дали бутонът с атрибут *„name= ‘Submit’“* е натиснат. Това се извършва благодарение на функцията *„isset()“*. Тя ще върне *„true“* ако в масива *\$_POST* е декларирана такава променлива отговаряща на името на бутона. Ако бутонът е натиснат то променливата ще бъде декларирана и функцията ще върне стойност *“true”* кода в конструкцията *„if“* да се изпълни, а ако ли не то функцията ще върне *„false“* и кода няма да се изпълни. Скрипта присвоява стойностите на променливите от масива *\$_POST* в отделни променливи и ги използва за да распечата необходимите съобщения чрез команда *„echo“*.

III. Задача за самостоятелна работа.

1. Тествайте задачата в практическата част. Редактирайте кода така че да отпечата и дата на раждане на съответния потребител и от какъв пол е. Добавете текстово поле във формуляра позволяващо на потребителя да въвежда повече от един ред информация. Нека и тази допълнителна информация да се извежда при натискане на бутона.
2. Редактирайте задача две от предходното упражнение, така че скрипта да използва формуляр от който потребител да въвежда данни за правоъгълника.
3. Създайте формуляр представляващ форма за вход, чрез който потребител въвежда своето потребителско име и парола. При натискането на бутон за изпращане да се зарежда PHP скрипт който да проверява дали потребителското име и парола ги има в даден масив създаден от програмиста в самия скрипт.
4. Със софтуер за рисуване или обработка на изображение създайте такова в което се съдържа следните фигури, кръг, квадрат, триъгълник, петоъгълник. Създайте PHP скрипт, обработващ формуляр, състоящ се от новосъздаденото изображение, който проверява какво изображение е избрано извежда текст посочващ името на съответната фигура.

Упътване: За да разберете координатите на курсора мишката за избраната фигура е необходимо да знаете размера на самото изображение в пиксели. Можете да го видите от свойствата на файла с изображението. Като имате предвид че позиция (0,0) се намира в горния ляв ъгъл на изображението то вие бихте могли да определите в коя част от него се намира курсора. Бутон за изпращане не е необходим за такъв формуляр информацията се изпраща при избиране на фигура от изображението.