

Лабораторна робота №4

Тема: Отримання даних з форми

Мета: Визначити способи отримання даних з форми

ТЕОРЕТИЧНІ ВІДОМОСТІ

ФОРМИ

Форми - це частина мови HTML. Форми потрібні для передачі даних від клієнта на сервер. Найчастіше форми використовуються для реєстрації користувачів, заповнення анкет, оформлення замовлення в інтернет магазині, і так далі. Через форми можна відправляти як просту текстову інформацію, так і файли. Велику частину часу програмування на PHP ви будете так чи інакше працювати з формами і даними з них.

HTML описує те, з яких елементів складається форма, і як вона виглядає. Але без приймаючої сторони, тобто сервера, який приймає ці дані і обробляє їх потрібним чином, створювати форми немає ніякого сенсу.

PHP містить безліч засобів для роботи з формами. Це дозволяє дуже просто вирішувати типові завдання, які часто виникають у веб-програмуванні:

- Реєстрація та аутентифікація користувача;
- Відправлення коментарів на форумах і соціальних мережах;
- Оформлення замовлень.

Практично будь-який сучасний сайт містить як мінімум кілька різних HTML-форм.

Відправлення форми

Розглянемо один типовий приклад - форма зворотного зв'язку. Для зв'язку користувачів з авторами сайту, як правило, використовуються форми зворотного зв'язку, де людина вказує ім'я, пошту для зворотного зв'язку і текст свого повідомлення.

Така форма в HTML може виглядати наступним чином:

```
<form name="feedback" method="POST" action="form.php">

  <label>Ваше ім'я: <input type="text" name="name"></label>

  <label>Ваш email: <input type="text"
name="email"></label>

  <label>Повідомлення: <textarea
name="message"></textarea></label>

  <input type="submit" name="send" value="Відправити">

</form>
```

Це дуже проста форма, що складається з трьох полів і однієї кнопки відправки. Майже весь наведений код описує зовнішній вигляд і зміст форми, але слід звернути увагу на два атрибути тега `<form>`, які потрібні для вказівки на спосіб обробки даних:

- `method`- цей атрибут використовується для визначення методу HTTP, який буде використаний для передачі даних на сервер. Ви вже знайомі з HTTP-методом `GET`, яке вказує сервера просто повернути певний документ.

Метод `POST` повідомляє про намір передати на сервер деяку інформацію, що, втім, не скасовує подальше отримання контенту.

- `action` - містить адресу PHP-скрипта, який повинен обробити цю форму. Після натискання на кнопку «відправити», браузер виконує `POST` запит з введеними даними на адресу, вказану в атрибуті `action`.

Обробка форми

Після відправки форми управління передається PHP-скрипту, який повинен отримати передані дані, виконати з ними будь-які дії (наприклад, зберегти в базі даних) і показати результат.

Результатом може бути будь-яке повідомлення про успішне завершення операції, наприклад, «ваші дані успішно відправлено».

Тому потрібно в першу чергу навчитися отримувати дані з форми в сценарії.

У PHP це робиться легко - **всі дані з форми знаходяться в глобальному асоціативному масиві `$_POST`**.

Цей масив завжди буде неявно присутній в сценарії, якщо він був завантажений за методом `POST`.

Кожне поле з форми буде знаходитися в масиві, де ключем буде значення атрибута `name`, а значенням вміст поля.

Наприклад, щоб вивести з форми всю інформацію на екран, можна написати такий сценарій:

```
<?php
if (isset($_POST)) {
    print("Ім'я: " . $_POST['name']);
    print("<br>Email: " . $_POST['email']);
    print("<br>Повідомлення: " . $_POST['message']);
}
```

Функція `isset` служить для визначення, чи існує передана їй змінна. Так ми перевіряємо, що сценарій завантажений методом `POST`, тобто була відправлена форма.

Як правило, після обробки форми в PHP, сценарій повинен переадресувати користувача на іншу сторінку. Це пов'язано з тим, що якщо форма була відправлена через метод `POST`, то після оновлення сторінки дані будуть відправлені ще раз, а це, в більшості випадків, небажану поведінку.

Відправка файлів

Крім текстової інформації, існує можливість відправляти на сервер файли будь-яких типів.

Приклад форми для завантаження файлу:

```
<form name="file_upload" method="POST" action="form.php"
enctype="multipart/form-data">

    <label>Ваш аватар: <input type="file"
name="avatar"></label>

    <input type="submit" name="send" value="Відправити файл">
```

```
</form>
```

Тут є дві важливі відмінності від першого прикладу:

- Додався новий атрибут `enctype`, який завжди повинен мати значення `multipart/form-data`. Якщо його не буде, то файл не відправиться.
 - Сам файл завантажується за допомогою поля з типом «file».
- У PHP долучення буде доступний в іншому спеціальному масиві - `$_FILES`.

```
<?php
```

```
if (isset($_FILES['avatar'])) {  
    $file = $_FILES['avatar'];  
  
    print("Завантажено файл с іменем " . $file['name'] . "  
та розміром" . $file['size'] . " байт");  
}
```

PHP автоматично зберігає всі завантажені файли в тимчасову папку на сервері. Але зберігати там файли можна, тому що ця директорія періодично очищається, і посилання на такий файл не можна дати на сайті. Рішення тут тільки одне - перемістити завантажений файл в іншу папку. Переміщення файлу завжди виконують відразу після завантаження.

Для початку потрібно переконатися, що в робочій директорії проекту існує папка для зберігання завантажених файлів. Нехай вона називається `uploads`.

Переміщення завантаженого файлу

Для переміщення файлу потрібно знати, де він знаходиться зараз, і адреса папки, в яку він буде переноситися. З поточним адресою все вкрай просто - він вже знаходиться в масиві `$_FILES`. Нова адреса файлу, в свою чергу, складається з шляху до папки і імені файлу. Так як папка `uploads` знаходиться там же, де і поточний сценарій, отримати шлях до неї можна так: `dirname(__FILE__)`.

Код для переміщення файлу в нову папку:

```
<?php
```

```
$current_path = $_FILES['avatar']['tmp_name'];  
$filename = $_FILES['avatar']['name'];  
$new_path = dirname(__FILE__) . '/' . $filename;  
  
move_uploaded_file($current_path, $new_path);
```

Функція `move_uploaded_file()` виконує дві дії:

1. Перевіряє, що файл дійсно завантажений через форму.
2. Переміщує завантажений файл за новою адресою.

Валідація форми

Валідація форми - це перевірка вмісту її полів. Завдання такої перевірки - переконаватися, що необхідні поля заповнені, а значення в них відповідають

очікуваному формату. Так, наприклад, при реєстрації користувача на сайті, він повинен заповнити поля з адресою електронної пошти та придумати собі пароль. Обидва поля є обов'язковими для заповнення, але значення з поля `email` також має бути коректним email-адресою. Крім текстових значень форми, можна перевіряти формат і розмір файлів.

Загальний підхід до валідації

При виконання валідації будь-якої форми порядок дій буде завжди одним:

1. Сформувати масив з іменами полів, обов'язкових для заповнення.
2. Сформувати масив з правилами для валідації формату полів.
3. Отримати значення всіх полів.

Обробка форми в одному файлі PHP

У попередньому уроці наша форма була розміщена на одній сторінці, а відправлялася на іншу. Насправді це не обов'язково. Якщо залишити атрибут `action` порожнім або прибрати його зовсім, то форма буде відправлятися на цю ж сторінку.

Як це буде працювати: при першому заході на сторінку ми заповнимо форму даними і натиснемо на кнопку. Після цього сторінка оновиться і її код виконається знову, але вже з даними форми.

Давайте подивимося на прикладі. Нехай у нас в одному файлі є форма і її обробка:

```
<form action="" method="GET">
    <input name="test1">
    <input name="test2">
    <input type="submit">
</form>

<?php
    var_dump($_GET);
?>
```

При першому заході на сторінку функція `var_dump` виведе порожній масив. А після відправки форми він виведе вже дані форми. Тобто перший раз `$_GET` буде порожній, а другий раз - буде містити дані форми.

Це може привести до проблем. Нехай, наприклад, ми в форму будемо вводити числа і хочемо вивести суму цих чисел на екран:

```
<form action="" method="GET">
    <input name="test1">
    <input name="test2">
    <input type="submit">
```

```
</form>
```

```
<?php
```

```
    echo $_GET['test1'] + $_GET['test2'];
```

```
?>
```

В цьому випадку при першому заході на сторінку ми побачимо помилки PHP, пов'язані з тим, що масив \$_GET порожній, а ми звертаємося до її елементів.

Тут слід сказати, що у вас помилки можуть і не з'явитися в браузері. В цьому випадку пров перевірте, що у вас включений висновок помилок PHP, а також переконайтеся, що у вас перший візит на сторінку і в адресному рядку немає даних форми.

Давайте виправимо проблему. Для цього додамо умова, в якому будемо перевіряти те, що форма була відправлена.

Наприклад, можна перевіряти \$_GET на не порожнечі. Якщо \$_GET не порожній - то форма була відправлена і можна виконувати підсумовування. В іншому випадку у нас ще перший візит на сторінку і підсумовування виконано не буде. Отже, ось виправлений код:

```
<form action="" method="GET">
```

```
    <input name="test1">
```

```
    <input name="test2">
```

```
    <input type="submit">
```

```
</form>
```

```
<?php
```

```
    if (!empty($_GET)) {
```

```
        echo $_GET['test1'] + $_GET['test2'];
```

```
    }
```

```
?>
```

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

Завдання 1. Створити форму даних за зразком. Отримати дані на серверній стороні.

Увійти до Навчально-
інформаційний портал
Відокремленого підрозділу
Національного університету
біоресурсів і
природокористування України
"Бережанський агротехнічний
інститут"

Увійти

[Забули пароль?](#)

На деякі курси передбачено гостьовий
доступ

Доступ як гість

Українська (uk) ▾

Повідомлення про файли cookie

Завдання 2. Створити форму для відправки повідомлення. Отримати дані.

Контактна форма

* Обов'язкове поле

Ім'я *

Електронна адреса *

Тема *

Повідомлення *

Надіслати

Завдання 3. Створити форму та вивести у файл передані дані

Створення нового обговорення документа

Назва проєкту

Назва проєкту...

Контрольні дати

Дата створення обговорення*

дд.мм.рррр



Завершення обговорення*

дд.мм.рррр



Файл проєкту (тільки файли pdf)*

Вибрати файли

Файл не вибрано

Ініціатор обговорення

☐ Адміністрація інституту ☐ Навчальний відділ ☐ Виховний відділ ☐ Вчений секретар ☐ Факультет ЕЕТ ☐ Факультет ЕІТ ☐ Факультет АІ

Додати проєкт до обговорення

Очистити форму