Інтернет програмування

лек

Лекція. PHP

Зав.каф.КН, PhD Граф М.С.

PHP-Сортування функцій для масивів

У цьому розділі ми проходитиме через такі функції сортування масиву PHP:

- `sort()`- Сортування масивів у порядку зростання
- `rsort()`- Сортування масивів у порядку зменшення
- `asort()`- Сортування асоціативних масивів у порядку зростання за значенням
- `ksort()`- Сортування асоціативних масивів у порядку зростання відповідно до ключа
- `arsort()`- Сортування асоціативних масивів у порядку зменшення за значенням
- `krsort()`- Сортування асоціативних масивів у порядку зменшення, відповідно до ключа

Обробка форм

Для збору даних форми використовуються PHP-глобальні $_get і $_post.

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

Для відображення надісланих даних можна просто повторити всі змінні. "Welcome. php" виглядає наступним:

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

Обидва Get і POST створити масив (наприклад, Array (ключ = > значення, Key2 = > value2, Key3 = > валуе3,...)). Цей масив містить пари «ключ-значення», де ключі – це імена елементів керування форми, а значення – вхідні дані користувача.
Як Get, і POST розглядаються як $ _жет і $ _пост. Це глобальні, що означає, що вони завжди доступні, незалежно від області видимості-і ви можете отримати до них доступ з будь-якої функції, класу або файлу без необхідності робити нічого особливого.

$ _жет-це масив змінних, що передаються поточному сценарію через параметри URL.
$ _пост це масив змінних, що передаються поточному сценарію за допомогою методу HTTP POST.

Коли використовувати Get?
Інформація, що надсилається з форми з методом GET, **видно всім** (усі імена змінних і значення відображаються в URL-адресі). Get також має обмеження на обсяг інформації, що відправляється. Обмеження складає близько 2000 символів. Однак, оскільки змінні відображаються в URL-адресі, можна закладати сторінку. Це може бути корисним у деяких випадках.
Get може використовуватися для надсилання не конфіденційних даних.

Коли використовувати POST?
Інформація, що надсилається з форми з методом POST, **невидима для інших** (всі імена та значення впроваджуються в тіло HTTP-запиту) і **не має обмежень** на обсяг інформації, що надсилається.
Крім того, POST підтримує розширені функціональні можливості, такі як підтримка багатокомпонентного двійкового введення під час завантаження файлів на сервер.

Form Validation

Перевірка форми PHP

Приклад перевірки форми PHP

* Обов'язкове поле

Ім'я: *

Електронна пошта: *

Веб-сайт:

коментар:

Стать: Жінка Чоловік Інше *

Надіслати

Ваші дані:

Правила перевірки цієї форми

Поле	Правила проверки
Name	Обязательно. + Должен содержать только буквы и пробелы
E-mail	Обязательно. + Должен содержать действительный адрес электронной почты (с @ и.)
Website	Дополнительные. Если он присутствует, он должен содержать допустимый URL-адрес
Comment	Дополнительные. Многострочный ввод (текстовое поле)
Gender	Обязательно. Необходимо выбрать один

Перевірка даних форми за допомогою PHP

Перше, що ми будемо робити, це передати всі змінні через PHP в хтмлспеціалчарс () функції.
При використанні функції хтмлспеціалчарс (); Потім, якщо користувач намагається надіслати наступне текстове поле:
< сценарій > розташування. href ('http://BBB.хаккед.ком')</script>
-Це не було б виконано, тому що воно було б збережено як код екранованого HTML, як це:
< скрипт & gt; Location. href ('http://BBB.хаккед.ком') & lt;/скрипт&АМП;ГТ;
Тепер код безпечний для відображення на сторінці або всередині електронної пошти.
Ми також робитимемо ще дві речі, коли користувач надсилає форму:
1.Прокладає непотрібні символи (зайвий простір, табуляцію, новий рядок) з вхідних даних користувача (за допомогою функції PHP Trim ())
2.Видалення зворотної косої межі (\) з вхідних даних користувача (за допомогою функції PHP stripslashes ())
Наступний крок полягає у створенні функції, яка робитиме всі перевірки для нас (що набагато зручніше, ніж писати ж код знову і знову).

Форми - обов'язкові поля

Поле	Правила проверки
Name	Обязательно. + Должен содержать только буквы и пробелы
E-mail	Обязательно. + Должен содержать действительный адрес электронной почты (с @ и.)
Website	Дополнительные. Если он присутствует, он должен содержать допустимый URL-адрес
Comment	Дополнительные. Многострочный ввод (текстовое поле)
Gender	Обязательно. Необходимо выбрать один

Форми. Перевірка електронної пошти та URL-адреси

PHP-перевірка імені

У наведеному нижче коді показаний простий спосіб перевірити, чи поле Name лише літери та пробіли. Якщо значення поля name неприпустиме, збережіть повідомлення про помилку:

```
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
  $nameErr = "Only letters and white space allowed";
}
```

Функція прег_матч () виконує пошук рядка для масиву, повертаючи значення true, якщо шаблон існує, і false інакше.

PHP-перевірка електронної пошти

Найпростіший і найбезпечніший спосіб перевірити, чи правильно сформована адреса електронної пошти - використовувати функцію PHP філтер_вар ().

У наведеному нижче коді, якщо адреса електронної пошти неправильно сформована, збережіть повідомлення про помилку:

```
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  $emailErr = "Invalid email format";
}
```

Форми. Перевірка електронної пошти та URL-адреси

PHP-перевірка URL

У наведеному нижче коді показаний спосіб перевірки допустимості синтаксису URL-адреси (це регулярне вираз також дозволяє тире в URL-адресі). Якщо синтаксис URL-адреси неприпустимий, збережіть повідомлення про помилку:

```
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-
9+&@#\/%=~_|]/i",$website)) {
  $websiteErr = "Invalid URL";
}
```

Дякую за увагу!