Лекція 4: Функції в PHP

План:

1. Визначення функцій в PHP
2. Функції із змінним числом аргументів
3. Повернення посилання функцій.

Внутрішні (вбудовані) функції

Луцьк 2015

1. Визначення функцій в PHP

У програмуванні, як і в математиці, функція є відображення безлічі її аргументів на безліч значень. Тобто, функція для кожного набору значень аргумента повертає відповідні значення, що є результатом її роботи. Функція може бути визначена за допомогою наступного синтаксису:

function Ім'я_функції (параметр1, параметр2 ... параметрN) { Блок_дій return "Значення, повернене функцією";}

Ім'я_функції **і** ***імена параметрів*** **функції (*параметр 1*, *параметр 2* і так далі) повинні відповідати правилам найменування в PHP. Імена функцій нечутливі до регістру. Параметри функції – це змінні мови, тому перед назвою кожного з них повинен стояти знак $. Після ключового слова *return* повинен йти коректний php-вираз. Загалом, у функції може і не бути параметрів та значення, яке вона повертає.**

1. Визначення функцій в PHP

Для виклику функції вказується ім'я функції і в круглих дужках список значень її параметрів, якщо такі є:

```
<?php Ім'я_функції("значення_для_параметра1",
"значення_для_параметра2" ...);?>
```

У PHP3 виклик функції можливий тільки після її визначення, тобто в будь-якому рядку програми нижче за блок *function f_name(){...}*. В PHP4 такої вимоги немає. Єдиний виняток становлять функції, визначені умовно (усередині умовних операторів або інших функцій). Коли функція зазначена таким чином, її визначення повинне передувати її виклику.

Якщо функція уже була визначена в програмі, то перевизначити або видалити її пізніше не можна. Не дивлячись на те, що імена функцій нечутливі до регістру, краще викликати функцію по тому ж імені, яким вона була задана у визначенні.

1. Визначення функцій в PHP

Аргументи функцій. **У кожної функції може бути список аргументів. Кожен аргумент є змінною або константою.**

За допомогою аргументів дані у функцію можна передавати трьома різними способами – це передача аргументів за значенням (використовується за замовчуванням), за посиланням та заданням значення аргументів за замовчуванням. Розглянемо ці способи докладніше.

Коли аргумент передається у функцію за значенням, зміна значення аргумента усередині функції не впливає на його значення поза функцією. Щоб дозволити функції змінювати її аргументи, їх потрібно передавати за посиланням. Для цього у визначенні функції перед ім'ям аргумента слід написати знак амперсанд «&».

1. Визначення функцій в PHP

```
<?php
/* напишемо функцію, яка б додавала до рядка слово checked */
function add_label(&$data_str)
{
$data_str .= "checked";
}
$str = "<input type=radio name=article "; /* нехай є такий рядок */
echo $str .="><br>"; /* виведе елемент форми – не відмічену радіо кнопку */
add_label($str);  // викличемо функцію
echo $str."><br>"; // виведе вже відмічену радіо кнопку
?>
```

1. Визначення функцій в PHP

У функції можна визначати значення аргументів, використовувані за замовчуванням. Саме значення повинне бути константним виразом, а не змінною і не представником класу або викликом іншої функції.

Якщо у функції декілька параметрів, то ті аргументи, для яких задаються значення за замовчуванням, повинні бути записані після решти всіх аргументів у визначенні функції. Інакше з'явиться помилка, якщо ці аргументи будуть опущені при виклику функції.

Наприклад, ми хочемо внести опис статті до каталогу. Користувач повинен ввести такі характеристики статті, як її назва, автор і короткий опис. Якщо користувач не вводить ім'я автора статті, вважаємо, що це Іванов Іван.

1. Визначення функцій в PHP

```
<?php
function Add_article($title, $description, $author = "Іванов Іван")
{echo "Заносимо в каталог статтю: $title,";
echo "автор $author";
echo "<br>Короткий опис: ";
echo "$description <hr>";
}  Add_article("Інформатика і ми", "Це стаття про інформатику ...", "Петров Петро");
Add_article("Хто такі хакери", "Це стаття про хакерів ...");?>
```

Результат роботи скрипта:

Заносимо в каталог статтю: Інформатика і ми, автор Петров Петро.

Короткий опис: Це стаття про інформатику...

Заносимо в каталог статтю: Хто такі хакери, автор Іванов Іван.

Короткий опис: Це стаття про хакерів...

2. Функції із змінним числом аргументів

У PHP4 можна створювати функції із змінним числом аргументів. Тобто ми створюємо функцію, не знаючи заздалегідь, з скількома аргументами її викличуть. Для написання такої функції ніякого спеціального синтаксису не потрібно. Все робиться за допомогою вбудованих функцій *func_num_args(), func_get_arg(), func_get_args()*.

Функція func_num_args() **повертає число аргументів, переданих в поточну функцію. Ця функція може використовуватися тільки усередині визначення функції, призначеного для користувача. Якщо вона з'явиться поза функцією, то інтерпретатор видасть попередження.**

2. Функції із змінним числом аргументів

```
<?php
function DataCheck()
{
$n = func_num_args();
echo "Число аргументів функції $n";
}
DataCheck(); /* виведе рядок "Число аргументів
функції 0 " */
DataCheck(1, 2, 3); /* виведе рядок "Число аргументів
функції 3" */
?>
```

2. Функції із змінним числом аргументів

Функція func_get_arg () **повертає аргумент із списку переданих у функцію аргументів, порядковий номер якого заданий параметром *номер_аргумента*. Аргументи функції нумеруються, починаючи з нуля. Як і *func_num_args()*, ця функція може використовуватися тільки усередині визначення якої-небудь функції.**

Номер_аргумента **не може перевищувати числа аргументів, переданих у функцію. Інакше згенерується попередження, і функція *func_get_arg()* поверне значення *False*.**

Функція func_get_args() **повертає масив, що складається із списку аргументів, переданих функції. Кожен елемент масиву відповідає аргументу, переданому функції. Якщо функція використовується поза визначенням призначеної для користувача функції, то генерується попередження.**

2. Функції із змінним числом аргументів

Комбінації функцій func_num_args(), func_get_arg() і func_get_args() використовуються для того, щоб функції могли мати змінний список аргументів. Ці функції були додані в PHP 4.

Використання змінних усередині функції. Щоб використовувати усередині функції змінні, задані поза нею, ці змінні потрібно оголосити як глобальні. Для цього в тілі функції слід перерахувати їх імена після ключового слова global:

2. Функції із змінним числом аргументів

```
<?php
$a = 1;
function Test_g()
{ global $a;
$a = $a*2;
echo "в результаті роботи функції $a=".$a;
}
echo "поза функцією $a=".$a;
Test_g();
echo "<br>";
echo "поза функцією $a=".$a;
Test_g();?>
```

Результат роботи скрипта:

поза функцією $a=1, в результаті роботи функції $a=2

поза функцією $a=2, в результаті роботи функції $a=4

2. Функції із змінним числом аргументів

Коли змінна оголошується як глобальна, фактично створюється посилання на глобальну змінну. Тому такий запис еквівалентний наступному (масив GLOBALS містить всі змінні, глобальні щодо поточної області видимості):

```
$var1 = &$GLOBALS["var1"];
$var2 = &$GLOBALS["var2"];
```

Це означає, наприклад, що видалення змінної $var1 не видаляє глобальної змінної $_GLOBALS["var1"].

2. Функції із змінним числом аргументів

Щоб використовувати змінні тільки усередині функції, при цьому зберігаючи їх значення і після виходу з функції, потрібно оголосити ці змінні як статичні. Статичні змінні видно тільки усередині функції і не втрачають свого значення, якщо виконання програми виходить за межі функції. Оголошення таких змінних проводиться за допомогою ключового слова static:

```
static $var1, $var2;
```

Статичній змінній може бути привласнене будь-яке значення, але не посилання.

2. Функції із змінним числом аргументів

Коли функція повертає декілька значень для їх обробки в програмі, зручно використовувати мовну конструкцію list(), яка дозволяє однією дією привласнити значення відразу декільком змінним. Наприклад, обробити повернені функцією значення можна було так:

```
<?php
  // завдання функції Full_age()
  list($day, $month,$year)= Full_age("07", "08", "1993");
  echo “Ви народились $year року $month місяця і $day дня";
?>
```

Взагалі конструкцію *list()* можна використовувати для привласнення змінним значень елементів будь-якого масиву.

3. Повернення посилання функцій. Внутрішні (вбудовані) функції.

Повернення посилання. **У результаті своєї роботи функція також може повертати посилання на яку-небудь змінну. Це може стати в нагоді, якщо потрібно використовувати функцію для того, щоб визначити, якій змінній повинне бути привласнене посилання. Щоб отримати з функції посилання, потрібно під час оголошення перед її ім'ям написати знак амперсанд (&) і кожного разу при виклику функції перед її ім'ям теж писати амперсанд (&). Зазвичай функція повертає посилання на яку-небудь глобальну змінну, посилання на статичну змінну або посилання на один з аргументів, якщо він був також переданий за посиланням.**

3. Повернення посилання функцій. Внутрішні (вбудовані) функції.

```
<?php
$a = 3;
$b = 2;
function & ref($par)
{
  global $a, $b;
  if ($par % 2 == 0) return $b;
  else return $a;
}
$var =& ref(4);
echo $var, " і ", $b"<br>";   // виведе 2 і 2
$b = 10;
echo $var, " і ", $b"<br>";  // виведе 10 і 10
?>
```

Під час використання синтаксису посилань в змінну *$var* не копіюється значення змінної *$b* поверненою функцією *$ref*, а створюється посилання на цю змінну.

3. Повернення посилання функцій. Внутрішні (вбудовані) функції.

Внутрішні (вбудовані) функції. **З деякими з вбудованих функцій, такими як echo(), print(), include(), date() ми вже познайомилися. Насправді всі перераховані функції, окрім date(), є мовними конструкціями. Вони входять в ядро PHP і не вимагають ніяких додаткових налаштувань і модулів. Функція date() теж входить до складу ядра PHP і не вимагає налаштувань. Але є і функції для роботи з якими потрібно встановити різні бібліотеки та підключити відповідний модуль. Наприклад, для використання функцій роботи з базою даних MySql слід скомпілювати PHP з підтримкою цього розширення. Останнім часом найбільш поширені розширення і їх функції спочатку включають до складу PHP так, щоб з ними можна було працювати без будь-яких додаткових налаштувань інтерпретатора.**

Дякую за увагу!