

PHP форма за връзка с клиенти

Евгения Христова (София)
info@shevitza.com

И така – имате сайт, по-вероятно статичен, хостинг пакета ви има поддръжка на PHP и се нуждаете от форма, в която вашите клиенти да попълват своите поръчки и кратки мнения и да ви ги пращат по електронната поща. Създаването на такава форма не е трудно, достатъчни са само познания по HTML. Що се отнася до PHP, там нещата съвсем не са толкова страшни, колкото изглеждат

PHP скрипт за имейл-форми може да получите понякога от фирмата, предоставяща ви хостинг. Това става или срещу допълнително заплащане или пък безплатно, с тълкуване като “бонус” – въпрос на маркетингова политика. Даже и когато получавате скрипта абсолютно безплатно, е добре да сте наясно как е направен.

Три в едно и на вашия компютър

За да може да правите проби на собствения си компютър, а не да качвате всички експерименти на мястото на реалния ви хостинг, свалете си някоя от многобройните пакети програми, които обединяват PHP, MySQL и сървъра Apache, настроени така, че да могат да работят съвместно, без допълнителни промени. Подобни пакети са EasyPHP, WAMP и много други. Те ще имитират средата на реалния хостинг, която в 90% от случаите включва PHP, MySQL, средство за администриране на базата, най-често PHPMyAdmin. За разнообразие ще ползваме един такъв пакет, фигуриращ сравнително отскоро на страниците на Download.BG – VertrigoServ. Другият вариант е да отделите няколко седмици или месеци, потъвайки в документацията на пакетите PHP, MySQL и Apache и накрая да научите как те се настройват за съвместна работа!

И така, да тръгнем по “мързеливия път”. Свалете си VertrigoServ, инсталирайте го по подразбиране, като си отбележите къде точно се инсталира – обикновено това е C:\Program Files\VertrigoServ

Преди стартирането на програма като Vertrigo, е добре да проверите колко сървъра Apache вече работят в паметта на вашата машина. Ако ползвате Skype ще установите, че “апахите” вече са тук, така че – спрете ги като с изход от Skype!

WWW е мястото на моя сайт

Интернет – страницата, която искате да изпробвате, трябва да разположите в папката www. PHP скриптове можете да стартирате през браузер, когато те са разположени в тази папка или в нейни подпапки. Създайте в тази папка подпапка, която се казва например mail_test и вътре една проста интернет – страница с име formpage.htm

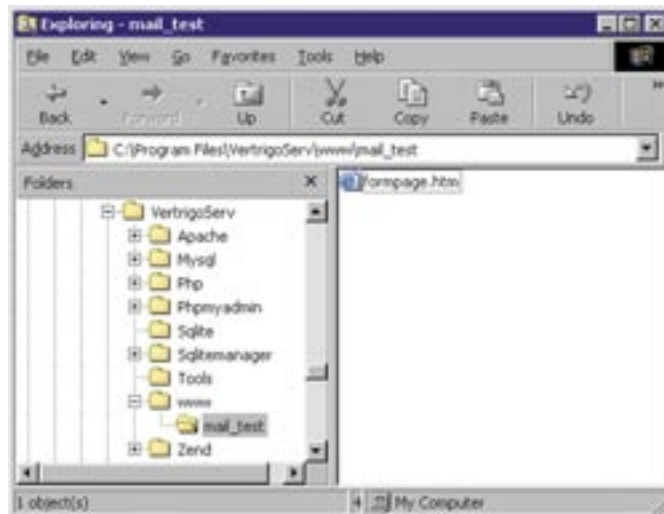
HTML страница с форма

В formpage.htm ще разположим една форма с три контроли – текстово поле, бутон за изпращане и бутон за изчистване. За по-добро подреждане, текстът на контролите и самите контроли ще разположим в таблица.

Ето кодът, който трябва да се разположи между таговете <body> </body>:

```
<form action="sendmail.php" method="POST">
<table>
<tr>
<td>Name</td>
<td><input type="text" name="name"
align="LEFT" size="20" maxlength="30" /></td>
</tr>
<tr>
<td>Send</td>
<td><input type="submit" value="Send"
/></td>
</tr>
<tr>
<td>Reset</td>
<td><input type="reset" value="Reset"
/></td>
</tr>
</table></form>
```

Тагът <form action="sendmail.php" method="POST"> дава началото на формата, като определя метода POST, по който ще бъдат изпратени до вас думите написани от клиента в текстовото поле с име name и действието action="sendmail.php", което ще свърши тази работа.

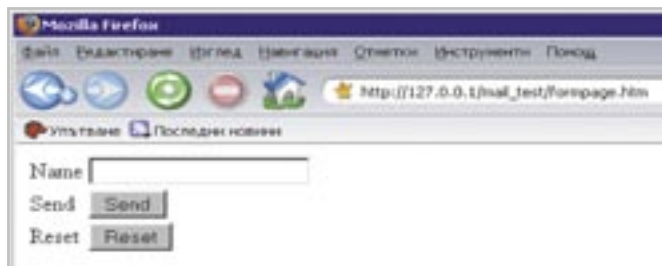


Освен върху страницата с формата, ще работим и върху файла sendmail.php за да постигнем целта си.

Да пуснем сървъра

Стартирайте програмата Vertrigo, след което стартирайте браузер и напишете в адресната лента `http://127.0.0.1/mail_test/formpage.htm`

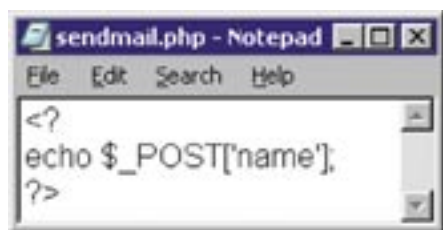
Ще видите вашата форма с полето за име:



тази форма все още “не умее” да прави нищо.

Първи PHP скрипт

Отворете бележника notepad и създайте с него следния файл:



Скриптът на php започва с `<?>` и завършва с `?>` Операторите завършват с точка и запетая. Съхранете файла като `sendmail.php`

На първия и третия ред са началото и края на скрипта, а втория разпечатва елемента на масива `$_POST`, който отговаря на съдържанието на текстовата кутия `name`. PHP е много интересен език, елементите на масивите се номерират с числа, както си е във всички други програмни езици, но заедно с това могат да идентифицират и със символни низове – точно това правим с този код.

Напишете нещо, например `Ivan` в полето `Name`, натиснете бутона `Send` и ще видите резултата.

В адресна лента виждате кой файл се показва, нали? Това е `sendmail.php`.

Взехме онова, което е написал потребителя и го показвахме в една, генерирана с php интернет страница. Е, сега ни трябва само да използваме функцията `mail`, за да изпращаме текста от потребителя в нашия е-mail. Това ще бъде нашата цел и тя е близко.

Сумиране на низове

Нека само още малко да се задържим тук, като покажем как към символния низ от текстовата кутия можем да добавим друг низ – за сумиране на символни низове в php се използва точка.

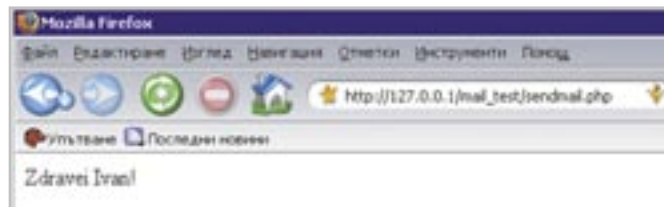
След първия ред на скрипта добавете

```
$_POST['name']="Zdravei ".$_POST['name'];
```

Съхранете, стартирайте и ще получите името с поздрава отпред.

Опитайте да оригирате кода с удивителна на края, не забравяйте да съхраните!

```
$_POST['name']="Zdravei ".$_POST['name']."!";
```



Сега Иван е доволен, а и ние заедно с него – също!

Променливи

Нуждаем се от една променлива, която да събира данните от контролите на нашата форма и да ни ги изпрати като писмо. В PHP имената на променливите започват със знак за долар - `$`, като може и да не обявяваме типа. Нашият код ще продължи да работи по същия начин, ако въведем променливата `$msg` и добавим “поздрава”

```
$msg=$_POST['name'];
```

```
$msg="Zdravei ".$msg."!";
```

По начина, по който добавихме поздрав, ще добавяме и стойностите на другите полета, попълнени от клиента. Но няма смисъл да хвърляме труд за оформление на полета, преди да сме се уверили, че ще изпратим имейл с това единствено поле.

Да помислим за котките и млякото

Да, не е зле още тук да помислим за сигурността. Добре е да ограничим възможността на Иван да “хвърли” в нашето текстово поле “зловреден код” и да ни “хакне” сайта L

По този повод винаги си мисля, че няма средство за сигурност, измислено от един човек, което да не може да бъде преодоляно от друг. Каквато и ключалка да си сложа, ако нямам късмет (по-малко вероятно) или ако с нещо предизвиквам злоба и завист у другите (доста вероятно!), все ще се намери някой да разбие вратата или поне да се опита. За това хората са измислили оная поговорка за покритото мляко и котките!

Ще използваме вградената функция `preg_match("маска", $_POST['name'])`, която ще върне лъжа, ако съдържанието на полето `name` съдържа различни от зададените в “маска” символи.

Ако става дума за име или съобщение, то трябва да съдържа само букви и евентуално и цифри, един телефонен номер има само цифри, имейлите имат латински букви, цифри, ниско тире, точка и знакът `@` и т.н.

Маската се подчинява на доста сложни правила, но все пак ще напишем няколко примера, които, изпробвани, ще ни създадат увереност, че тя работи добре.

Маската е символен низ, следователно се загражда в кавички. Започва с `/^` и завършва с `/`. В квадратни скоби се изброяват последователност от символи, които са разрешени.

```
- /^[a-zA-Z0-9_]/- тази маска ще позволи символът @,
```

малки и главни латински букви и ниско тире и точка. Тя е подходяща за поле, което ще приема e-mail адреси.

- /^[a-zA-Z0-9a-zA-Я_]/ - с тази маска може да въвеждаме кирилски и латински букви и цифри, тя е подходяща за имена или текст. Не забравяйте да поставите в главата на htm файла meta за кирилица!

- /^[0-9]{3,12}\$/ ще позволи въвеждане на 3 до 12 цифри и интервал и е подходяща за въвеждане на телефонен номер.

Дължината на текстовото поле във формата също може да стане ограничител за въвеждането на символи. В нашия случай текстовото поле приема до 30 символа.

Следващия код ще провери дали в полето са въведени от 3 до 12 цифри, ако да – ще прехвърли стойността на name в променливата за съобщение \$msg, ако не – ще изтрие съдържанието на елемента на масива \$_POST. След това съдържанието на двете променливи се печата за контрол.

```
if(preg_match("/^[0-9 ]{3,12}$/", $_POST['name'])) {
    $msg=$_POST['name'];
}
else {
    $_POST['name']="";
}
echo $msg;
echo $_POST['name'];
```

Смисълът на горните редове е в това, че ако въведеният от потребителя символен низ съдържа само разрешените символи, резултатът от въвеждането ще се разпечата, ако не – не се появява нищо, защото на елемента от масива \$_POST['name'] се присвоява празен низ.

Добре е да опитаме и другите маски, по зададените вече правила и да наблюдаваме резултата, за да сме сигурни, че работи както трябва. С това можем да смятаме проблема за съдържанието на текстови полета, попълнени от потребителя, за решен.

Изпращане на e-mail

Ще използваме функцията mail(a1,a2,a3, [a4] [,a5]). Тук a1,a2... са аргументите на функцията. Първите три са задължителни, останалите, заградени в квадратни скоби са по желание на потребителя.

- * a1 е e-mail адреса на получателя, в случай на собственика на сайта – например 'myaddress@myhost.com';
- * a2 е subject, темата на писмото – например 'e-mail test'
- * a3 е съдържанието на писмото. Тук ще сложим променливата \$msg

И така, преди края на скрипта добавете

```
mail('myaddress@myhost.com', 'e-mail test' $msg);
```

Ако се опитате да изпробвате кода от своя компютър, по начина, по който тествахме до сега, ще получите съобщение за грешка.

Има начин да се настрои работещия на компютъра ви сървър Арасче и клиентската програма за поща, така че да изпратите до себе си писмо. Но това ще отнеме доста от времето ви и

за това ще напишем функцията mail и после ще изпробваме скрипта направо от хостинга.

Ето как ще изглежда кода в sendmail.php, ако поставим поле за обикновен текст.

```
<?
$msg= "";
if(preg_match("/^[a-zA-Z0-9a-zA-Я_]/", $_
POST['name'])) {
$msg=$_POST['name'];
}
else {
$_POST['name']="";
}

echo $msg." \n";
echo $_POST['name'];
mail('myname@mydomain.com', 'e-mail test', $msg);
?>
```

Съхранете, като зададете вашият имейл-адрес и качете на хоста си файловете sendmail.php и formpage.htm

Попълнете нещо в полето и би трябвало да получите съобщение в кутията си.

Добавяне на други видове контроли

Тези контроли ще връщат готови стойности, отразяващи избора на клиента и няма да е необходима проверка на въведеното в тях. Използваме името на контролата по аналогичен на първия случай начин и \n, за да получим данната на нов ред.

Във таблицата на formpage.htm трябва да се добави ред

```
<tr><td>Избор от падащ списък</td>
<td><select name="myselect" size=1>
<option value = "Sel1">Първи избор
<option value = "Sel2">Втори избор
<option value = "Sel3">Трети избор
</select>
</td></tr>
```

В sendmail.php се добавя към \$msg

```
$msg=$msg." \n Избор: ". $_POST['myselect'];
```

Радиобутони

Във таблицата на formpage.htm трябва да се добави ред:

```
<tr><td>Начин на плащане:</td>
<td> cash <input type="radio" name = "payment" value="cash"
/>
bank<input type="radio" name = "payment" value="bank" /></
td></tr>
```

В sendmail.php се добавя към \$msg:

```
$msg=$msg." \n Плащане: ". $_POST['payment'];
```

Кутия за отметки

```
<tr><td>Искам доставка:</td>
<td><input type="checkbox" name="ship">
</td></tr>
```

В sendmail.php се добавя към \$msg

```
$msg=$msg."\n Доставка: " . $_POST['ship'];
```

Резултатът

След малко редактиране на файла с формата и няколко проби получаваме:

info@shevitza.com: **Четене на писмо**

ОТ:	web@wishmaster.com.bg
ПОЛУЧАТЕЛ:	info@shevitza.com
ТЕМА:	e-mail test
ДАТА:	07.06.2006 12:40

Име: Иван Иванов
 email: ivan@proba123.com
 Избор: Sel2
 Плащане: cash
 Доставка: on

Форма за поръчки

Име	<input type="text" value="Иван Иванов"/>
e-mail	<input type="text" value="ivan@proba123.com"/>
Продукт:	<input type="text" value="Втори избор"/>
Плащане:	в брой <input type="radio"/> банка <input checked="" type="radio"/>
Доставка:	<input checked="" type="checkbox"/>
<input type="button" value="Изпрати"/> <input type="button" value="Изчисти"/>	

Изчистване на формата

След като клиентът е попълнил формата и е изпратил писмото, той вижда контролния печат от файла sendmail.php. Разбира се, че бихме могли да му благодарим, но това вероятно ще направите по-късно, когато усъвършенствате своите умения. Сега проблемът ни е, че при натискане на бутона Back, формата остава попълнена и това може да го озадачи повече, отколкото поздравленията и благодарностите. Файлт sendmail.php ще завършим като поставим след последния ред на който има ?> още един ред:

```
<meta http-equiv="refresh"
content="0;URL=formpage.htm">
```

Така контролният печат от sendmail.php не се вижда, а след изпълнение на функцията mail се зарежда вече изчистената форма.

Когато човек има възможност да ползва поддръжка на PHP при хостинг и познанията му са малко, той в повечето случаи не изпитва нужда да преработи сайта си в php файлове. Той не усеща още предимството на този подход, но скрипт за връзки с клиенти всеки би искал да има. Решението с Java Script е по-трудно, това с PHP има сериозно предимство – простотата. Посланията на клиентите може, освен да се пращат по имейл, да се запишат във файл или в база данни – през всичко това редовия потребител ще мине през следващите месеци и години, но може би ще започне своето приключение в света на PHP точно с форма за връзка с клиенти!

