web разработка

PHP функции

Съдържание

- Същност
- Функции с параметри
- Подготовка за проект - 2
  - Извикване на файл в друг файл
  - Организиране на файловете

Същност

Функции /методи/ - същност

Задача 1 Отпечатайте 10 пъти Hello!, в три различни тага.

Функции /методи/ - същност - 2

```php
<?php
//1

echo "<h1>";
for($i = 0; $i < 10; $i++){
    echo "Hello!";
}
echo "</h1>";

//2
echo "<p>";
for($i = 0; $i < 10; $i++){
    echo "Hello!";
}
echo "</p>";


//3
echo "<table border=1><tr>";
for($i = 0; $i < 10; $i++){
    echo "<td>"."Hello!"."</td>";
}
```

Дали има начин да оптимизираме решението на задачата –
да премахнем многократното повтаряне на един и същи код –
/в случая поне да избегнем повтарянето на цикъла for/?

Функции /методи/ - същност - 3

- ✓ Парчета код /изпълняващи някаква цел/, които можем да използваме многократно;
- ✓ Подреденост и организация на кода;
- ✓ Намаляват редовете код - няма нужда да копираме многократно едни и същи редове код, когато се налага да извършваме една и съща последователност от операции -> използваме функции.
- ✓ Ако се наложи да поменяме кода - променяме само на едно място - там, където е дефинирана функцията/метода/.

Функции /методи/ - същност - 4

Предназначението на функцията е да реши конкретен проблем.

Да разделим функциите условно на –

- Вградени функции в PHP - count(), rand(), sort(), date() …..http://php.net/
- Функции, които ние дефинираме за да решим конкретен проблем, който е с вероятност да се повтаря многократно.

Да се върнем към задача 1 и да трансформираме решението, използвайки функция ->

Функции /методи/ - същност - 5

Преди

```php
<?php
//1

echo "<h1>";
for($i = 0; $i < 10; $i++){
    echo "Hello!";
}
echo "</h1>";

//2
echo "<p>";
for($i = 0; $i < 10; $i++){
    echo "Hello!";
}
echo "</p>";


//3
echo "<table border=1><tr>";
for($i = 0; $i < 10; $i++){
    echo "<td>"."Hello!"."</td>";
}
```

След добавяне на функция

```php
<?php
//1 step - function declaration
function print_text(){
    for($i = 0; $i < 10; $i++){
        echo "Hello!";
    }
}

///2 step call function - 3 times in three different tags
//1 tag
echo "<h1>";
print_text();
echo "</h1>";

//2 tag
echo "<p>";
print_text()
echo "</p>";

//3 - !!!! won`t change this code for now!?
echo "<table border=1><tr>";
for($i = 0; $i < 10; $i++){
    echo "<td>"."Hello!"."</td>";
}
echo "</tr></table>";
```

Функции с параметри

Функции с параметри

Преди

```php
<?php
//1 step - function declaration
function print_text(){
    for($i = 0; $i < 10; $i++){
        echo "Hello!";
    }
}

///2 step call function - 3 times in three different tags
//1 tag
echo "<h1>";
print_text();
echo "</h1>";

//2 tag
echo "<p>";
print_text()
echo "</p>";

//3 - !!!! won`t change this code for now!?
echo "<table border=1><tr>";
for($i = 0; $i < 10; $i++){
    echo "<td>"."Hello!"."</td>";
}
echo "</tr></table>";
```

След добавянето на параметри

```php
<?php
//1 step - function declaration
function print_text($param1, $param2){
    echo $param1;
    for($i = 0; $i < 10; $i++){
        echo "Hello!";
    }
    echo $param2;
}

///2 step call function - 3 times in three different tags
//1 tag
$a = '<h1>';
$b = '</h1>';
print_text($a, $b);


//2 tag
$a = '<p>';
$b = '</p>';
print_text($a, $b);


//3 - !!!! won`t change this code for now!?
echo "<table border=1><tr>";
for($i = 0; $i < 10; $i++){
    echo "<td>"."Hello!"."</td>";
}
```

Функции с параметри - 2

Броя на параметрите при дефиниране трябва да са еднакви с тези, които използваме. Какво ще стане ако броя е различен?

3. Дефинирайте функция, която генерира и отпечатва матрица MxN със стойност на елементите, започващи от X и увеличаващи се с 1.
4. Дайте възможност промяната на стойността на елементите на масива да се задава като параметър във функцията.

Функции с параметри - 2

Броя на параметрите при дефиниране трябва да са еднакви с тези, които използваме. Какво ще стане ако броя е различен?

```php
//1 step - function declaration
function print_text($param1, $param2, $text){
    echo $param1;
    for($i = 0; $i < 10; $i++){
        echo $text;
    }
    echo $param2;
}
```

Броя на параметрите е 3

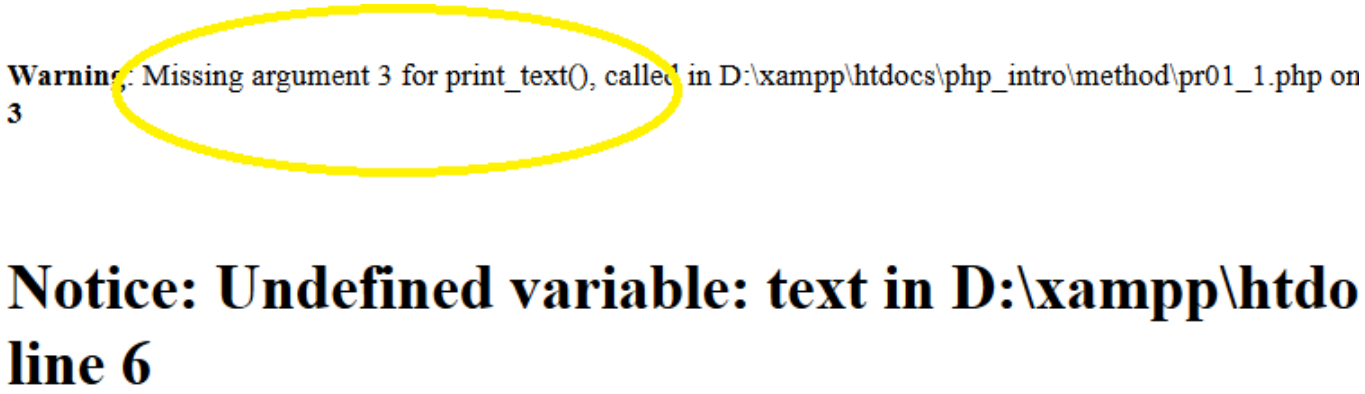
```php
$a = '<h1>';
$b = '</h1>';
print_text($a, $b);


//2 tag
$a = '<p>';
$b = '</p>';
print_text($a, $b);
```

Но извикваме пак с 2 параметъра

Функции с параметри - 2

Броя на параметрите при дефиниране трябва да са еднакви с тези, които използваме. Какво ще стане ако броя е различен?



Получаваме съобщение за липсващ трети параметър на функцията, която извикваме.

Функции с параметри - 3

Трансформираме
и третия код –
отпечатваме с
извикване на
функцията, която
дефинирахме ->

Забележка – като параметър при извикването
на функцията може
да подадем и стринг –
например ‘Hello’ или др.
в зависимост от контекста
на задачата.

```php
<?php
//1 step - function declaration
function print_text($param1, $param2, $text){
    echo $param1;
    for($i = 0; $i < 10; $i++){
        echo $text;
    }
    echo $param2;
}

///2 step call function - 3 times in three different tags
//1 tag
$a = '<h1>';
$b = '</h1>';
$c = 'Hello';
print_text($a, $b, $c);


//2 tag
$a = '<p>';
$b = '</p>';
//$c = 'Hello' - no need to write this again
print_text($a, $b, $c);


//3 tag
$a = '<table border=1><tr>';
$b = '</tr></table>';
$d = "<td>Hello!</td>";
print_text($a, $b, $c);
```

Функции с параметри - 3

Тъй като вече сте луди фенове на масивите – запазете стойностите, които подавате като параметри в многомерен масив.

Извикайте пак три пъти дефинираната от вас функция, но този път !!! обърнете внимание как ще опишете необходимите параметри.

Функции - задачи

Подготовка за проект 2

Извикване на файл в друг файл

Обикновено, дефинираните от нас функции се запазват в един файл – наречен напр. functions.php. Този файл извикваме там, където ще използваме функциите посредством една от готовите функции -

- include(),
- include_once()
- require(),
- require_once()
- Детайлно обяснение на употребата им - http://php.net/manual/en/function.require.php

Синтаксис – include_once('functions.php')

Извикване на файл в друг файл - 2

functions.php

```php
<?php
//DESCRIPTION OF PRINT_TEXT - WHAT IT IS DESIGNED FOR
//BECAUSE WE FORGET SO VERY QUICKLY:))

function print_text($param1, $param2, $text){
    echo $param1;
    for($i = 0; $i < 10; $i++){
        echo $text;
    }
    echo $param2;
}
```

файла, в който използваме функциите

```php
<?php
//1 step - include the file, where the function`s
//been deckared.
//MIND THE WAY TO THAT FILE!!!
include_once('functions.php');

$params = array (
            array('<h1>','</h1>', 'Hello!'),
            array('<p>','</p1>', 'Zdr!'),
            array('<table border=1><tr>','</tr></t
            );
//1 tag
```

Проект – 2, структуриране

Начални препоръки

1. Уеднаквете имената на променливите, които използвате – вие и вашия колега, с който работите върху проекта – може да направете списък с имената на променливите, които използвате и коя - каква информация съхранява.
2. Направете скици на страниците на приложението си – как ще изглеждат, каква информация ще съдържат, какво ще прави потребителя на съответната страница. Каква ще бъде навигацията между страниците, какъв ще бъде футъра.

Hint – използвайте мисловни карти - https://www.youtube.com/watch?v=jz5_K-a4j4kp, и mockups https://moqups.com/ или листа /многоа листа/ и цветни моливи и пишете, рисувайте страниците на вашето уеб-приложение.

Не забравяйте –
всичко, което ви хрумне
може да бъде описано
с код
– просто трябва
да откриете
КАК?!

структуриране

Основни файлове –
index.php – файла, който се зарежда при стартиране на приложението ви
…..php – файлове, реализиращи логиката на приложението ви.

header.php – съдържа, повтарящото се във всяка страница, начално съдържание;
footer.php – съдържа, повтарящото се във всяка страница финално съдържание;
functions.php – дефинираните от вас функции, които използвате в кода на проектите си.
Папки – css, images, includes /в която поместваме файловете, които вмъкваме в други файлове – напр. functions.php, header.php, footer.php/

структуриране – header

header.php – съдържа,
повтарящото се
във всяка страница,
начално съдържание.
„Отрязваме“ повтарящото
се начало на
страницата и
го запазваме в header.php.

```php
<?php
include_once('functions.php');
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>app structure</title>
</head>
<body>
<?php
someGreeting();
$params = array (
            array('<h1>','</h1>', 'Hello'),
            array('<p>','</p1>', 'Zdr!'),
            array('<table border=1><tr>','</tr></table>', '<td>Hi!!</td>')
            );
//1 tag
print_text($params[0][0], $params[0][1], $params[0][2]);
//2 tag
print_text($params[1][0], $params[1][1], $params[1][2]);
//3 tag
print_text($params[2][0], $params[2][1], $params[2][2]);
?>
</body>
</html>
```

header.php

index.php/.....php

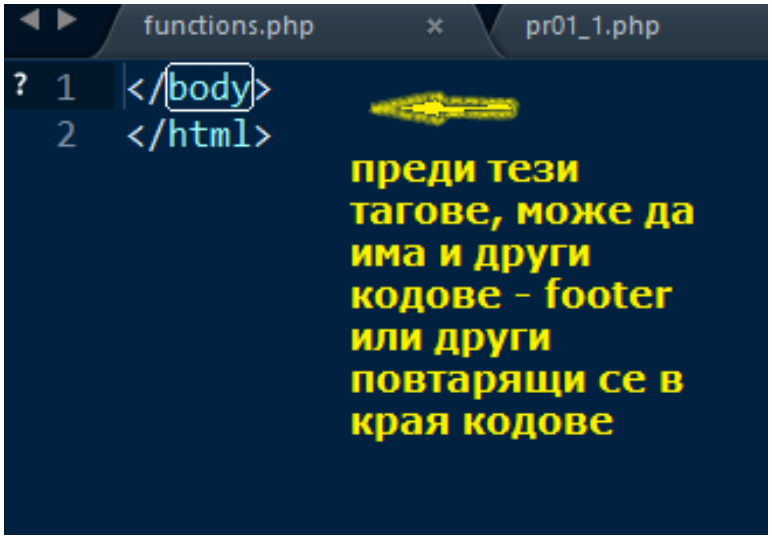
footer.php

структуриране – header - 2

header.php

структуриране – footer.php

footer.php



структуриране – index.php

index.php, …..php

```php
<?php
include_once('header.php');
someGreeting();
$params = array (
            array('<h1>','</h1>', 'Hello!'),
            array('<p>','</p1>', 'Zdr!'),
            array('<table border=1><tr>','</tr></table>', '<td>Hi!!</td>')
            );
//1 tag
print_text($params[0][0], $params[0][1], $params[0][2]);
//2 tag
print_text($params[1][0], $params[1][1], $params[1][2]);
//3 tag
print_text($params[2][0], $params[2][1], $params[2][2]);
include_once('footer.php');
?>
```

структуриране – проверка

Ако сме свързали правилно частите на всяка страница, освен логиката, която искаме нашето уеб приложение да изпълнява и която трябва да работи правилно, с F12 трябва да видим един правилно структуриран HTML код.