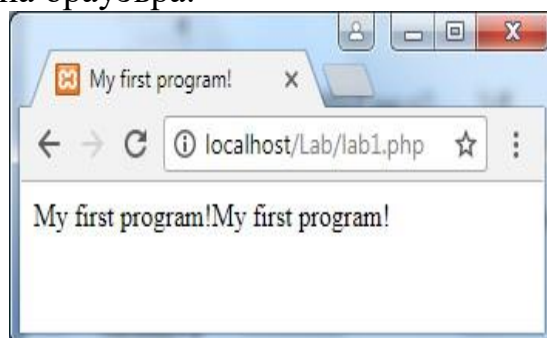


Упражнение 1. Основни елементи на PHP – синтаксис на езика, константи, променливи, изрази и оператори

Задача 1. В тялото на една HTML страница, създайте PHP секция, която извежда съобщение „ My first program!“ в прозореца на брауъра.

Примерна реализация:

```
<html>
  <head>
    <title>My first program!
  </title>
</head>
<body>
  <?php
    echo "My first program!";
    // Вместо echo може да използвате print.
    print "My first program!";
  ?>
</body>
</html>
```



Пример 1. Да се изведе заглавието на страницата от първа задача чрез PHP код:

```
<html>
  <head>
    <title><?php echo "My first program!"; ?></title>
  </head>
<body>
<?php echo "My first rogram!";
?>
</body>
</html>
```

Пример 2. Да се реализира първа задача, като се вгради HTML код в PHP скрипт.

```
<?php
echo "<HTML>
  <HEAD>
  <TITLE>My first program!</TITLE>
  </HEAD>
  <BODY>
  <b>My first program!</b>
  </BODY>
</HTML>";
```

?>

Символни низове

Пример 3. Символните низове в PHP могат да се оградят с кавички и с апострофи:

```
<html>
  <head>
    <title><?php echo "My first program!"; ?></title>
  </head>
  <body>
<?php
  echo "Hi!";      //в кавички
  echo ("Hi!");   //можете да използвате и скоби
  echo 'Hi!';     //в апостроф
  echo ('Hi!');   //можете да използвате и скоби
// Вместо echo може да използвате print.
  print "Hi!";
  print ("Hi!");
  print 'Hi!';
  print ('Hi!');
?>
</body>
</html>
```

Параметри:

```
echo "Hi!","Hi!"; // ok
echo ("Hi!"), ("Hi!"), ("Hi!!"); // ok
echo ("Hi!", "Hi!"); // fail
print ("Hi!"); // ok, print() приема само един параметър
// (израз или стринг)
print ("Hi!"), ("Hi!"), ("Hi!"); // fail
```

Задача 2. Изведете в брауъра използвайки само echo/print:

$10 + 7 = 17$

$10 - 7 = 3$

$10 * 7 = 70$

$10 / 7 = 1.4285714285714$

$10 \% 7 = 3$

Примерна реализация:

```

<HTML>
<HEAD>
<TITLE>echo/print</TITLE>
</HEAD>
<BODY>
<?php
  echo "10+7=", 10+7, "<br>";      //ok (с кавички)
  echo '10-7=', 10-7, "<br>";    //ok (с апостроф)
  print "10*7=".(10*7)."<br>";  //ok (с кавички)
  print '10/7='.(10/7)."<br>";  //ok (с апостроф)
  print "10%7=".(10%7). '<br>'; //ok(с кавички и апостроф)
  print "10%7=", 10%7, "<br>";  //fail
?> </BODY>

```

Константи

Константа се нарича идентификатор с единствена стойност, която не може да се променя по време на изпълнение на скрипта. Прието е имената на константите в PHP да се задават с главни букви.

Дефиниране на константи

define(“име_на_константа”, стойност[, чувствителност_към_регистъра])

- “име_на_константа” - низ;
- стойност - валиден PHP израз, но не масив или обект;
- чувствителност_към_регистъра - булева стойност (true/ false) и е незадължителна. Подразбиращата стойност е false (чувствителен).
- Проверка за дефинирането на константа може да стане чрез функция defined("име на константа").
- Стойността на една константа се извлича чрез името и, но също така и чрез функция constant("име на константа").
- Логическите константи имат стойности TRUE и FALSE.

Пример 4. Разгледайте всички деклариращи константи

```

<pre>
<?php echo
  "PHP_VERSION=".PHP_VERSION."<br>";
  print_r(get_defined_constants());
?> </pre>

```

Пример 5. Дефиниране на потребителски константи

```

<?php

```

```

//дефиниране на константа PASSWORD
define("PASSWORD","123456");
//дефиниране на константа независима от регистъра PI
//със стойност 3.14
define("PI","3.14", True);
// извеждаме стойността на PASSWORD,
echo "<br>";
echo "PASSWORD=".PASSWORD;
echo "<br>PASSWORD =";
echo constant("PASSWORD");
echo "<br>";

// извежда предупреждение, че password е регистър-
//зависима константа

    if (defined("password"))
        echo "password=".password;
    else
        echo "Constant password is not defined!";
    echo "<br>";

// извежда 3.14 тъй като константа PI е независима от
//регистъра
echo "pi=".pi;
?>

```

Променливи в PHP

Променливите в PHP не се декларират явно, типа им зависи от присвоената стойност. Пред променливите в PHP се поставя знака \$ и могат да започват с долна черта или буква, като могат да съдържат букви, цифри и знак за подчертаване. Чувствителни са към главни и малки букви. По време на изпълнение на програмата променливата може да променя типа си, т.е. да приема стойности от различни типове. Инициализацията на променливите не е задължителна. Стойността на не инициализираните променливи зависи от контекста в който се използват. Така по подразбиране за променливите от логически тип подразбиращата се стойност е FALSE, за числов тип (integer, float) – 0, за тип string – празен низ, за масив (тип array) – празен масив. Променливите в PHP се обявяват по следния начин:

\$име_на_променлива = стойност;

Инициализиране на променливи може да се прави:

- със стойност (\$y = \$x);

- с адрес (`$y = &$x`);
- индиректно;

Пример 6. Деклариране и инициализация на променливи

```
<?php
//инициализация със стойност
$a = 'Nora';
$y=$a;
$name="Maria";
//инициализация с адрес
$z = &$name;
echo "a=", $a, " y=", $y, " z=", $z;
$x = 'Peter';
//индиректна инициализация - съдържанието на
//променливата $x се използва като име на променлива
$$x = 'John'; //създава се $Peter='John'
echo " x=".$x." Peter=".$Peter; //$Peter ще изведе
//John
?>
```

Управление на променливите:

- **isset (var)** - връща TRUE, ако променливата var е установена и не е NULL и FALSE - в противен случай. Може да се използва и за повече от една променлива - връща TRUE, ако всички променливи са установени;
- **empty (var)** – връща TRUE, ако променливата няма стойност (променливата не е установена или има стойност: 0,0.0,"0","", FALSE, NULL, празен масив).
- **unset (var1, var2,...)** – унищожава се стойността на специфицираната променлива. Поведението на unset() във функция варира в зависимост от това какъв тип променлива искате да унищожите. Ако глобална променлива се унищожава с unset() вътре във функцията, само локалната променлива се разрушава, а променливата във викащата функция не губи стойността си:

Пример 7. isset/ unset/ empty

```
<?php
$my_name = 'Иван Иванов';
if (isset($my_name))
    echo "Променливата my_name има стойност $my_name!<br>";
    // унищожава се стойността на променливата $my_name
unset($my_name);
```

```

if (isset($my_name))
    echo "Променливата my_name има стойност $my_name!<br>";
else
    echo "Променливата my_name няма вече стойност!!!<br>";

if (empty($my_name))
    echo "Променливата my_name е празна<br>";
else
    echo " Променливата my_name не е празна<br>";
?>

```

Основни типове данни

Пример 8. Примерът съдържа PHP код, в който е зададена променлива за всеки един тип данни, както и преобразуване от един тип в друг. Оператори за преобразуване на типове : (int), (float), (string), (bool):

```

<HTML>
<HEAD>
<TITLE>First PHP Script</TITLE>
</HEAD>
<BODY>
<?php
    // boolean: true or false
    $validEmail = true;
    // integer
    $number = 30;
    // Floating point number -(float)
    $temp = 25.7;
    //string
    $name = 'Valeria';
    //Empty type(NULL)
    $second_number = NULL;
    echo "email=$validEmail, number=$number, temp=$temp,
name=$name and second_number=$second_number";
    //email=1, number=30, temp=25.7,
    //name="Valeria", second_number=
    $speed = 503.789;
    //създаваме една променлива с плаваща запетая
    echo "<p>$speed</p>"; //503.789

```

```

    $intSpeed = (integer)$speed; //( int )
    //сега ще преобразуваме $speed в цяло число
    echo $intSpeed;           //503
?>
</BODY>
</HTML>

```

Проверка на типа на дадена променлива: `gettype()`, `is_type(var)`

Пример 9.

```

<?php
    $x = 'George';
    echo "x= ".gettype($x)."<br/>\n";
    $x = 88.9;
    echo "x= ".gettype($x)."<br/>\n";
    $x = true;
    echo "x= ".gettype($x)."<br/>\n";
    $x = 8;
    echo "x= ".gettype($x)."<br/>\n";
    $x = null;
    echo "x= ".gettype($x)."<br/>\n";
?>

```

Оператори на PHP

- аритметични - +, -, *, /, %
- логически: and, or, not, xor;
- за сравнение: ==, !=, >, <, >=, <=, ===(идентично Пример: \$a=== \$b True ако \$a е = \$b и са от един и същи тип (въведено от PHP4), !== (неидентично Пример: \$a!== \$b True ако \$a не е = \$b или ако не са от един и същи тип (въведено от PHP4));
- Оператор за забраняване на извеждане на грешка - @;

Пример 10. Даден е скрипт, който създава две променливи \$x=10 и \$y=7 и изчислява и извежда сумата, разликата, произведението, частното и остатъка от делението им код по следния начин:

```

10 + 7 = 17
10 - 7 = 3
10 * 7 = 70
10 / 7 = 1.4285714285714
10 % 7 = 3

```

Реализация:

```

<?php

```

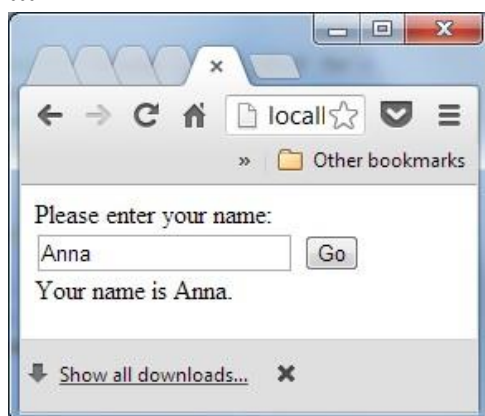
```

$x=10;
$y=7;
$result=$x+$y;
echo "$x + $y = $result<br />";
$result=$x-$y;
echo "$x - $y = $result<br />";
$result=$x*$y;
echo "$x * $y = $result<br />";
$result=$x/$y;
echo "$x / $y = $result<br />";
$result=$x%$y;
echo "$x % $y = $result<br />";

```

?>

Пример 11. Създайте скрипт, който използва HTML форма за въвеждане на име. След натискане на бутон се извежда следния отговор „Your name is ... ”



Примерна реализация:

```

<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=utf-8">
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <form method="post" action="<?php $_PHP_SELF ?>">

```

Или: <FORM name="form" method="post" action="<?php \$_SERVER['PHP_SELF']; ?>" >

Или: <FORM name="form" method="post" action="#" >

```
Please enter your name: <br />
<input type="text" name="ime" />
<input type="submit" name="subm" value="Go" />
</form>
<?php
if (isset($_POST['subm']))
{
//Retrieve string from form submission.
$name = $_POST['ime'];
echo "Your name is $name.";
}
?>
</body>
</html>
```

Забележка:

`$PHP_SELF` е сървърна променлива, която връща името и пътя на текущия файл, тоест както `$_SERVER['PHP_SELF']`. Но, ако промените `register_globals` на `off`, то `$PHP_SELF` няма да е достъпна. Затова се препоръчва:

```
<FORM name="form" method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>" >
```

Пример 11. Създайте скрипт, който използва HTML форма за въвеждане на име и възраст. След натискане на бутон се извежда следния отговор „Здравейте,! Вие сте на ...години.“

Здравейте! Моля попълнете следната информация :

Вашето име :

Възраст :

здравейте,Иван
вие сте на Петров години!

```
<HTML>
<HEAD>
<HTML>
<HEAD>
```

```

<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=utf-8">
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <title></title>
  </head>
<body>
<p>Здравейте! Моля попълнете следната информация : <br>
<FORM name="form" method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>" >
Вашето име : <input type="text" name="ime"><br>
Възраст : <input type="text" name="godini"><br>
<input type="submit" name="subm" value="Изпрати!">
</form></p>
<?php
if (isset($_POST['subm']))
  {
  // $ime = $_POST['ime']; $godini=$_POST['godini'];
  // echo "<p>Здравейте, $ime, вие сте на $godini години! ";
  echo "здравейте, " . $_POST['ime'] . "<br>". "вие сте на ".
  $_POST['godini'] . " години! ";
  }
?>
</body>

```

Забележка:

Вместо:

```

<FORM name="form" method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>" >

```

Може да се напише:

```

<form method="post" action="#">
<form method="post" action="<?php $_PHP_SELF ?>">
<form method="post" action="<?php $_SERVER['PHP_SELF'];
?>"

```

Полезни сайтове:

<http://tutor-bg.hit.bg/Html%20Files/phptut.html>

<http://vladilovech.site50.net/>

http://www.tutorialspoint.com/php/php_web_concepts.htm