

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: ИНФОРМАЦИОННИ СИСТЕМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 1

ТЕМА: Основни елементи на PHP. Синтаксис на езика

ЦЕЛ:

Целта на упражнението е студентите да се запознаят със основните елементи на езика PHP. След упражнението студентите би следвало да могат да боравят с променливи и константи, да пишат кратки изрази чрез PHP скриптове.

ВАЖНО! За упражненията по дисциплината е нужен следния софтуер: Уебсървър, PHP интерпретатор, бази от данни MySQL, интерфейс за работа с Бази от данни (PhpMyAdmin и Workbench), Уеб-браузър и текстов или скриптов редактор. На компютрите в лаборатория 3408, за текстов или скриптов редактор инсталиран "Notepad++", а за останалия необходим софтуер е инсталиран пакет WAMP (съкращение от Windows, Apache(уебсървър), MySQL и PHP) версия 3.2.0.

I. ТЕОРЕТИЧНА ЧАСТ

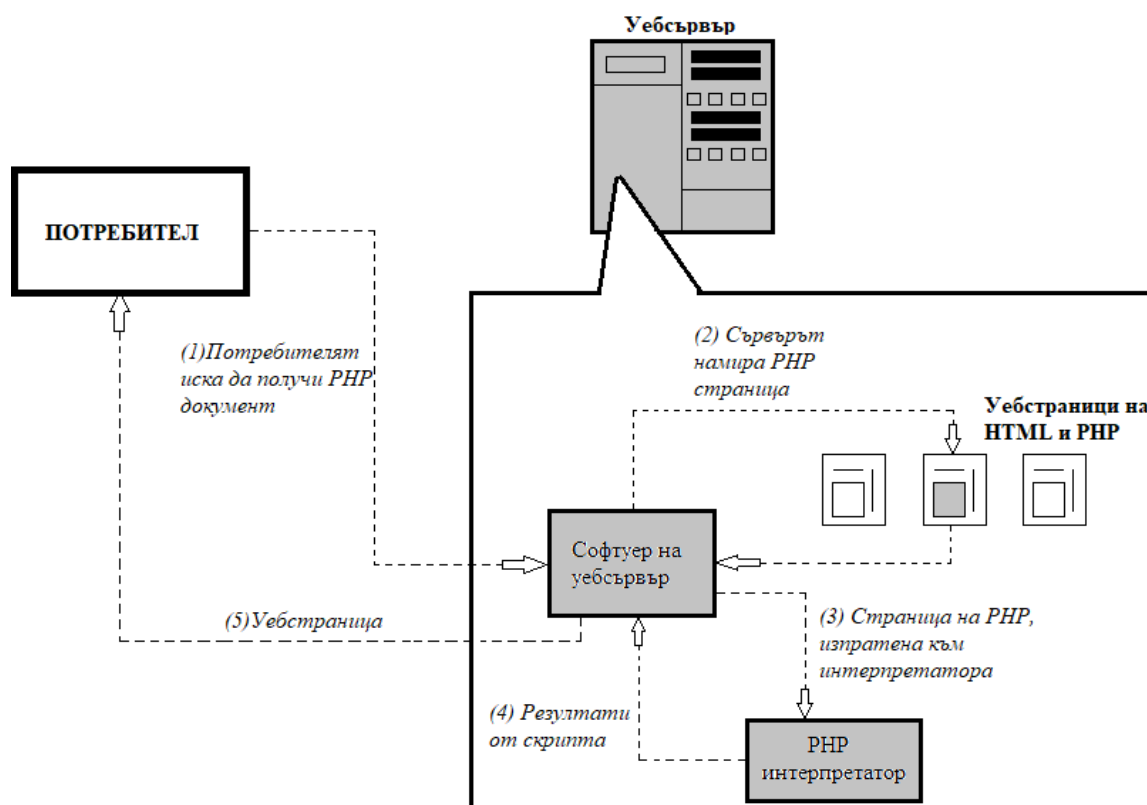
1. Въведение.

PHP е много популярен и широко приложим скриптов език, който може да бъде вграден в текст на HTML и особено подходящ за уеб-разработки.

PHP е скриптов език, който в комбинация с (X)HTML позволява създаването на динамични уеб-системи. Чрез него може да се преодолее едно от най-сериозните ограничения на (X)HTML документите, а именно тяхната естествена статичност. Стандартният (X)HTML документ се съхранява на сървъра и когато бъде поискан със заявка от клиента, се предоставя на потребителя, който може да види съдържанието му в брауъра си. Самият документ не се променя. Той може да съдържа код на JavaScript или приставка, които осигуряват известно взаимодействие с потребителя, но не могат да променят основното съдържание или да осъществят връзка с базата от данни, така че да отразят едно динамично актуализиране на съдържанието при всяко зареждане.

PHP обработва хипертекстови документи, преди те да бъдат подадени към брауъра. Той е сървърен език и всичко което извършва се случва на сървъра, преди резултата да бъде изпратен на клиента (фиг.1.).

На фиг.1. се вижда как потребителя е заявил да получи PHP документ от уебсървър. Тъй като документа е написан на PHP сървъра го подава първо към PHP интерпретатор. Той анализира текста на документа за да открие в него командите на PHP. Когато открие такава команда интерпретаторът я замества с резултата, който тя генерира. Когато прегледа всичко, готовият документ с премахнат PHP код се изпраща към брауъра на потребителя.



Фиг.1. Начин на работа на PHP.

За създаването на PHP скрипт се използва текстов или скриптов редактор в който се пише кода. След което е необходимо този скрипт да се запише като файл с .php разширение. За изпълнението на php файла е необходимо той да се постави в директорията на уебсървъра. За WAMP пакета това е:

C:\wamp\www

В тази директория могат да се създават отделни поддиректории в които да се съхраняват файловете от отделни проекти. За изпълнението на файла е необходимо в брауъра да се зареди следния адрес:

localhost/(поддиректория на съответния проект)/(име на файла)

2. Вграждане на PHP в HTML код

Това е много добра практика за осигуряване на качество и постоянство на уеб-документите. PHP кода е огражда от специални отварящи и затварящи тагове, които маркират началото и края на скрипта:

```
<php
...
?>
```

Той може да се вгради многократно в един HTML документ, както е показано в зад.1. в практическата част. Също с помощта на апострофи или кавички, би могло в самия PHP скрипт да се вградят HTML тагове.

3. Команди echo и print()

Това са езикови конструкции а не функции, въпреки че print() се държи като функция. И двете се използват за да връщат стойност. За техния синтаксис са важни следните условия:

- Когато връщаната стойност е символен низ, то той се поставя в кавички или апостроф. Ако се използват кавичките, то може да се включат и имена на променливи в съответния низ. Ако низа е ограден с апостроф и се включат имената на променливите, то резултата ще разпечата съответното име на променливата.
- Когато връщаната стойност е от числов формат то кавички и апострофи не се поставят.
- Точката (.) в операторите echo и print е оператор за конкатениране

В символните низове може да се използват и скоби заедно с кавичките и апострофите.

Примери:

```
echo "Hello World";           // ще изведе съобщението "Hello World" на екрана
echo ("10-7="). (10-7);       // ще изведе 10-7=3
```

4. Операции в РНР

Означенията на някои от по основните операции в РНР са следните:

- аритметични – +, -, *, /, %;
- логически:
 - ✓ логическо И – && или and;
 - ✓ логическо ИЛИ – || или or;
 - ✓ логическо отрицание – !;
- за сравнение – ==, !=, >, <, >=, <=;
- за присвояване – =.

5. Константи в РНР

Константите в РНР се декларираат с функция define(). Тя има следната структура:

```
define("име", "стойност", [чувствителност към регистъра]),    където:
```

- име – това е низ отговарящ за името на съответната константа.
- Стойност – валиден РНР израз, но не масив или обект, отговарящ за стойността на съответната константа
- Чувствителност към регистъра – Това е незадължителен аргумент. Той се използва за да покаже дали константа е чувствителна към големи и малки букви. Ако се пропусне този аргумент по подразбиране е "false" което означава че чувствителността към регистъра не е включена т.е. ще има разлика между големи и малки букви

ЗАБЕЛЕЖКА! Чувствителността към регистъра е спряна след версия РНР 7.3.0 и в по горни версии е напълно премахната. Затова използвайте свикнете да използвате функцията само в първите два аргумента и приемайте че има разлика между главни и малки букви.

За проверка дали дадена константа е дефинирана се използва функцията defined(„име“). Тя има само един аргумент отговарящ на името на константата. Функцията връща булев „true“ или "false" в зависимост от това дали константата е дефинирана или не.

Примери:

```
define("PASSWORD", "querty");
echo "PASSWORD=".PASSWORD;    //ще изведе PASSWORD=querty
echo "PASSWORD=".password     //ще изведе грешка понеже няма такава константа
```

6. Променливи в PHP

Променливите в PHP не се декларират явно. Типа им зависи от присвоената стойност. Пред променливите в PHP се използва знака \$ и могат да започват с долна черта или буква, като могат да съдържат букви, цифри и знака за долна черта.

Инициализацията на променливите не е задължителна, но се счита за добра практика. Стойността на не инициализираните променливи зависи от контекста в който се използват. Така по подразбиране за променливите от логическите тип подразбиращата се стойност е „false“. За числов тип (integer, float) – 0, за тип string – празен низ, за масив (тип array) – празен масив. Инициализиране на променливите може да се прави:

- със стойност (\$y=5;)
- с адрес(\$y=&\$x)
- индиректно – съдържанието на променливата се използва като име на променлива (\$x="Peter"; \$\$x="John"; - така се създават две променливи първата е с име x и стойност Peter, а втората е с име Peter и стойност John)

Последната възможност не се среща в други езици и е едно от големите предимства на PHP.

7. Проверка за типа на дадена променлива

Типа може да се провери чрез функцията `gettype(име)`. Тя има само един аргумент, който е името на променливата за която се проверява типа. Функцията връща като резултата типа на съответната променлива:

Пример:

```
$my_var="Georgi"; Echo gettype(my_var); //ще изведе string
$my_var=88.9; Echo gettype(my_var); //ще изведе double
$my_var=8; Echo gettype(my_var); //ще изведе integer
$my_var=null; Echo gettype(my_var); //ще изведе NULL
```

8. Управление на променливи

В PHP съществуват някои специални функции за управление на променливите. Това са:

- `isset(var)` – връща „true“ ако променливата `var` е установена и не е NULL, в противен случай връща „false“;
- `empty(var)` – връща „true“ ако променливата `var` не е установена, т.е. ако няма стойност. За такива случаи се броя и когато променливата приема някоя от следните стойности: „0“, „0.0“, „“, FALSE, NULL, празен масив.
- `unset(var1,var2,...)` – унищожава се стойността на специфицираната променлива. Поведението на функцията варира в зависимост от това какъв тип променлива ще се унищожава. Ако глобална променлива се унищожава вътре във функция, то ще се загуби само локалната и стойност, а променливата във викащата функция не губи стойността си.

Примери за използването на тези функции ще видите в някои от следващите упражнения.

9. Променливи в обкръжението

Чрез тези променливи може да се изведе важна информация за посетителя на една страница, като: от кой уеб адрес е дошъл потребителя, какъв е IP адреса му, каква

операционна система и браузър използва и т.н. Някои от най използваните променливи в обкръжението са следните:

- `$_SERVER['HTTP_USER_AGENT']` - дава информация за браузъра и ОС
- `$_SERVER['REMOTE_ADDR']` - дава информация за IP адреса
- `$_SERVER['SERVER_SOFTWARE']` - дава информация за сървъра
- `$_SERVER['HTTP_REFERER']` - дава информация за URL, откъдето идва user-а.

II. ПРАКТИЧЕСКА ЧАСТ

В практическата част са показани примери чрез които са обяснени нагледно отделните елементи в PHP. За целта са решени следните задачи:

ЗАДАЧА1: Да се създаде една PHP страница, със стандартни HTML тагове `<head>`, `<title>` и `<body>` tags. Съхранете файла в директорията на уебсървъра с разширение `.php`. Изведете следното заглавие на страницата "Hello Boys and Girls!" и следното съобщение „Hello World“ чрез PHP код.

КОД:

```
<html>
  <head>
    <title><?php echo "Hello, Boys and Girls!"; ?></title>
  </head>
  <body>
    <?php
      echo "Hello, World!";
    ?>
  </body>
</html>
```

ПОЯСНЕНИЕ

За изпълнението на кода е необходимо да го запишем във файл с `.php` разширение, например `example1.php`. Файла трябва да бъде поставен в уеб-директорията на сървъра. Ако е използван пакет WAMP това е `C:\wamp\www`. За да го изпълним ще е необходимо да заредим в браузъра следния адрес: `localhost/example1.php`.

Горния код е пример за вграждане на php скрипт в една html страница. Първия php код извежда следното заглавие на страницата "Hello, Boys and Girls!", а вторият извежда съобщението „Hello, World!“. Този код е и пример как чрез команда `echo` се извежда символен низ на екрана.

ЗАДАЧА2: Създайте php скрипт с две променливи `$x=10` и `$y=7`, който изчислява сумата, разликата, произведението, частното и остатъка от делението им в прозореца на браузъра по следния начин:

Сумата е: $10+7=17$

Разликата е: $10-7=3$

Произведението е: $10*7=70$

Частното е: $10/7=1.4285714285714$

Остатъка е: $10\%7=3$

Запишете файла в уеб-директорията на сървъра с наименование *example2.php* и го изпълнете.

КОД:

```
<?php
    $x=10;
    $y=7;
    $result=$x+$y;
    echo "<br> Сумата е: $x+$y=$result";
    $result=$x-$y;
    echo "<br> Разликата е: $x-$y=$result";
    $result=$x*$y;
    echo "<br> Произведението е: $x*$y=$result";
    $result=$x/$y;
    echo "<br> Частното е: $x/$y=$result";
    $result=$x%$y;
    echo "<br> Остатъка е: $x%$y=$result";
?>
```

ПОЯСНЕНИЕ

За да изпълним горния файл е необходимо да заредим в браузъра следния адрес:

localhost/example2.php

Задачата демонстрира как се използват променливите и операциите. Как се разпечатват стойности на променливи чрез оператор `echo`.

ЗАДАЧА3: Напишете PHP скрипт който да извлече следната информация за потребителя: име на сървъра, ip адреса му, информация за уеб сървъра и за неговия браузър и операционна система.

КОД

```
<?php
    echo "SERVER_NAME=".$_SERVER['SERVER_NAME'];
    echo "<br>";
    echo "SERVER_ADDR=".$_SERVER['SERVER_ADDR'] ;
    echo "<br>";
    $x=$_SERVER['SERVER_SOFTWARE'];
    echo "Вашия уеб сървър е: $x";
    echo "<br>";
    echo "Вашият браузър и операционната система са:".$_SERVER['HTTP_USER_AGENT'];
    echo "<br>";
?>
```

ПОЯСНЕНИЕ

За да изпълните задача 3 съхранете горния код във файл *example3.php* в директорията на уебсървъра, след което в адресната лента на браузъра напишете:

localhost/example2.php

Като резултат в браузъра ще се разпечата информация за вашия сървър, ip адрес, уебсървъра който използвате, както и информация за браузъра и операционната ви система.

III. Задача за самостоятелна работа.

1. Тествайте примерите задачите в практическата част.
2. Създайте PHP скрипт който да дефинира константа „AUTHOR“ чиято стойност е „Simon Stibart“. Пробвайте и да достъпите константата с малки и с главни букви. Нека при достъпването им да се разпечата стойността им. Анализирайте резултата.
3. Създайте PHP скрипт в който се декларира една променлива с име num и стойност 8. Нека върху тази променлива да се извършват необходимите операции, така че да се получат посочените стойности на изхода. Нека скрипта да е така реализиран че да визуализира по следния начин на екрана в браузъра:

Value is now 8.

Add 2. Value is now 10.

Subtract 4. Value is now 6.

Multiply by 5. Value is now 30.

Divide by 3. Value is now 10.

Increment value by one. Value is now 11.

Decrement value by one. Value is now 10.