

Laboratorul 9

PHP - partea I

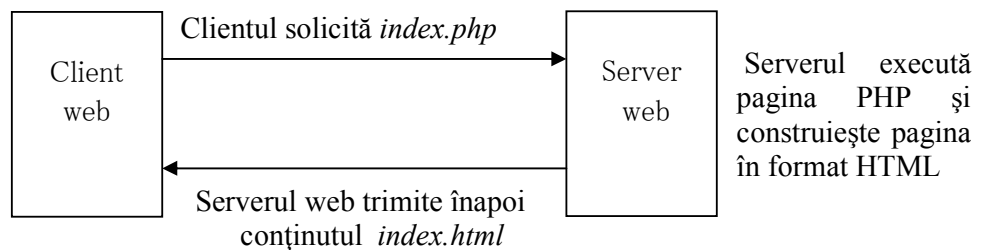
1. Introducere

Limbajul PHP

- PHP – **P**HP **H**ypertext **P**reprocessor
- Limbaj de scriptare de uz general
- Script-urile sunt rulate și executate pe server
- Alternativă gratuită pentru Microsoft ASP
- Permite lucrul cu diferite tipuri de baze de date

Caracteristici PHP

- **Familiaritatea** – sintaxa limbajului este foarte ușoară, combinând sintaxele unora din cele mai populare limbaje, precum Perl, Java sau C
- **Simplitatea** – sintaxa limbajului este destul de liberă. Nu este nevoie de includere de biblioteci sau de directive de compilare, codul PHP inclus într-un document executându-se între marcajele speciale
- **Eficiența** – PHP-ul se folosește de mecanisme de alocare a resurselor foarte necesare unui mediu multi-utilizator, așa cum este web-ul
- **Securitate** – PHP-ul pune la dispoziția programatorului un set flexibil și eficient de măsuri de siguranță
- **Flexibilitate** – PHP este modularizat. Nefiind legat de un anumit server web, PHP-ul a fost integrat pentru numeroasele servere web existente: Apache, IIS etc. Poate fi rulat pe diferite platforme (Windows, Linux, Unix)
- **Gratuitate** – Dezvoltarea PHP-ului sub licența open-source a determinat adaptarea rapidă a PHP-ului la nevoile web-ului, eficientizarea și securizarea codului.



Fișiere și sintaxă PHP

- Fișierele PHP pot conține text, elemente HTML și scripturi PHP
- Pot avea extensia **.php**, **.php3** sau **.phtml**
- Un script PHP este încadrat de **<?php** și **?>**
- Poate fi folosită și încadrarea **<? și ?>**
- Fiecare linie de cod PHP se termină cu caracterul “;”
- Comentariile sunt incluse cu **“//”** (comentarii pe o linie) sau sunt încadrate de **“/*” și “*/”** (comentarii pe mai multe linii)
- Pentru scrierea în documentul HTML corespunzător se pot folosi funcțiile **echo** și **print**

```
<html>
<body>

<?php
    echo "Test <br /><b>PHP</b>";
?>

</body>
</html>
```

2. Variabile, operatori, tipuri de date

Variabile în PHP

- Pot memora numere, șiruri de caractere, vectori
- O variabilă în PHP începe cu caracterul “\$”
- O variabilă nu trebuie declarată, PHP face automat conversia variabilei la tipul datei atribuite
- Numele unei variabile începe cu o literă sau “_” și poate conține caracterele “a”-“z”, “A”-“Z”, “0”-“9” și “_”

```
<html>
<body>

<?php
    $mesaj = "Mesaj PHP!";
    $val = 12;
    print($mesaj. ($val+$val));
?>

</body>
</html>
```

Operatori PHP

- Aritmetici – “+”, “-”, “*”, “/”, “%”, “++”, “--”
- De concatenare – “.”
- De atribuire – “=”, “+=”, “-=”, “*=”, “/=”, “.=”, “%=”
- De comparare – “==”, “!=”, “<”, “>”, “<”, “>=”, “<=”
- Logici – “&&”, “||”, “!”

Variabile de tip șir de caractere

- Există un singur operator specific șirurilor de caractere, cel de concatenare
- Există un set de funcții specifice (strlen(), strpos() etc.)

```
<?php
echo strlen("Hello world!");
?>
```

Vectori

- a. Vectori cu index numeric – există două metode de creare a vectorilor cu index numeric

```
<html>
<body>

<?php
    $meniu1 = array("File","Edit","Search","View");
    $meniu2[0]="Load";
    $meniu2[1]="Save";
    $meniu2[2]="Print";
    $meniu2[3]="Close";
    echo "Primele optiuni din meniuri sunt ".$meniu1[0]." si
    ".$meniu2[0].".";
?>

</body>
</html>
```

- b. Vectori asociativi – indexul ia anumite valori specifice

```
<html>
<body>

<?php
    $salarii = array("Ion"=>3200, "Vasile"=>1300,
    "Gheorghe"=>18200);
    $vechime['Ion'] = "30";
    $vechime['Vasile'] = "10";
    $vechime['Gheorghe'] = "21";
    echo "Vasile are salariul de ".$salarii['Vasile']."RON si
    vechimea de ".$vechime['Vasile']." de ani.";
?>

</body>
</html>
```

c. Vectori multi-dimensionali – conțin unul sau mai mulți vectori

```
<html>
<body>

<?php
    $angajati = array (
        "Ion"=>array("vechime"=>12, "salariu"=>2100),
        "Vasile"=>array("vechime"=>6, "salariu"=>9000));
    echo "Vasile are salariul de
        .$angajati['Vasile']['salariu']
        ."RON si vechimea de "
        .$angajati['Vasile']['vechime']." de ani.";
?>

</body>
</html>
```

3. Instrucțiuni de test

- **If**

```
if (conditie) secvență cod;
```

- **if ... else**

```
if (conditie)
    secvență cod1;
else
    secvență cod2;
```

- **if ... elseif ... else**

```
If (conditie1)
    secvență cod1;
elseif (conditie2)
    secvență cod2;
else
    secvență cod3;
```

- **switch**

```
switch (n)
{
case val1:
    secvență cod1;
    break;
case val2:
    secvență cod2;
    break;
default:
    secvență cod3;
}
```

Exemplu "switch"

```
<html>
<body>

<?php
    $x = 2;
    switch ($x)
    {
        case 1: echo "Numarul are valoarea 1";
                break;
        case 2: echo "Numarul are valoarea 2";
                break;
        case 3: echo " Numarul are valoarea 3";
                break;
        default: echo " Numarul nu are valoarea in 1, 2 sau 3";
    }
?>

</body>
</html>
```

4. Instrucțiuni ciclice

- **while**

```
while (conditie)
{
    secvență cod;
}
```

- **do ... while**

```
do
{
    secvență cod;
}
while (conditie);
```

```
<html>
<body>
```

```
<?php
    $i=1;
    do
    {
        $i++;
        echo "Numarul este " . $i . "<br />";
    }
    while ($i<=5);
?>
```

```
</body>
</html>
```

- **for** – parametrii pot fi vizi sau pot avea mai multe valori separate prin virgulă
(*initializare; conditie; increment*)

```
{
  secvență cod;
}
```

- **foreach**

```
foreach ($vector as $var)
{
  secvență cod;
}
```

```
<html>
```

```
<body>
```

```
<?php
```

```
  $x=array(1,2,3);
  foreach ($x as $value)
  {
    echo ($value+1)."<br />";
  }
}
```

```
?>
```

```
</body>
```

```
</html>
```

5. Funcții PHP

- Există funcții predefinite dar pot fi definite și funcții utilizator
- Numele funcției poate începe cu o literă sau “_”

```
function numeFuncție()
{
  secventa cod;
}
```

```
<html>
```

```
<body>
```

```
<?php
```

```
function adunare($x,$y)
{
  $suma = $x+$y;
  return $suma;
}
```

```
echo "32 + 14 = " . adunare(21,14);
```

```
?>
```

```
</body>
```

```
</html>
```