

PHP tutorial

Observatie generala:

Scripturile php incluse în codul html sunt executate pe server. Asta are ca efect faptul ca daca încercati sa vizualizati codul sursa al unei pagini web dinamice o sa observati ca el contine doar tag-uri html, acestea fiind rezultatul obtinut în urma interpretarii codului php.

Sintaxa php

Un fisier php contine tag-uri normale de html si cod php. Marcarea începutului codului php se face cu „<? PHP” iar sfârșitul bocului de cod php cu „?>”

Exemplu:

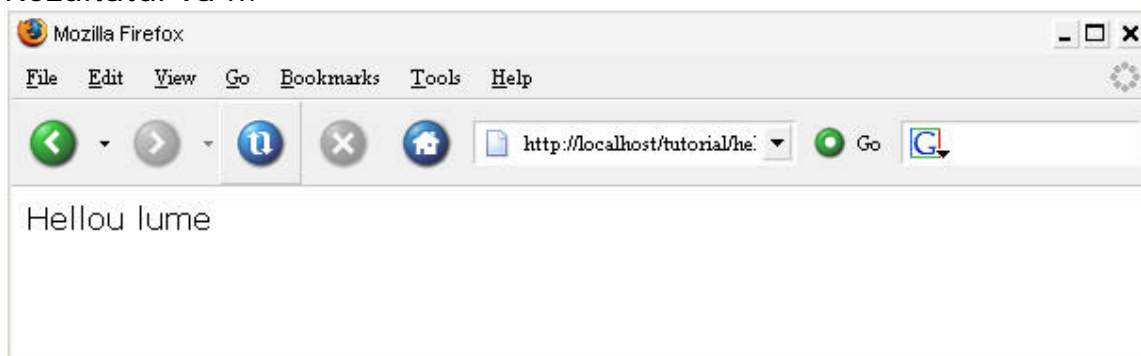
Clasicul exemplu cu „Hellou lume” transformat în script php:

```
<html>
<body>
    <?php echo "Hellou lume"; ?>
</body>
</html>
```

Observatii:

Fisierul de mai sus trebuie salvat cu extensia php pentru ca severul de web sa transmita fisierul interpretorului php

Rezultatul va fi:



iar codul sursa returnat (vizibil de altfel cu View Source) este urmatorul:

```
<html>
<body>
    Hellou lume</body>
</html>
```

Un bloc de script php poate fi inserat în orice loc a unui document html. Fiecare instructiune php trebuie separata de o alta instructiune cu ;

Afisarea în php

1. Afisarea cu ajutorul lui "echo"
2. Elemente de formatare a codului sursa
3. Afisarea elementelor HTML
4. Escap-ul pentru anumite semne
5. Afisarea semnelor speciale din PHP

1. Afisarea cu ajutorul lui "echo"

Pentru a afisa ceva în PHP (adica codul html pentru o pagina) se pot utiliza mai multe functii: **echo**, **print** und **printf**. Functia **printf()** se utilizeaza asemanator ca în c în sensul ca permite formatarea afisarilor. Noi vom folosi functia **echo** care este putin mai rapida ca **print**. Diferenta dintre echo si print este ca print returneaza întotdeauna valoarea 1 dupa ce a fost executata si deci poate fi utilizata în constructii mai complexe.

Exemplu 1

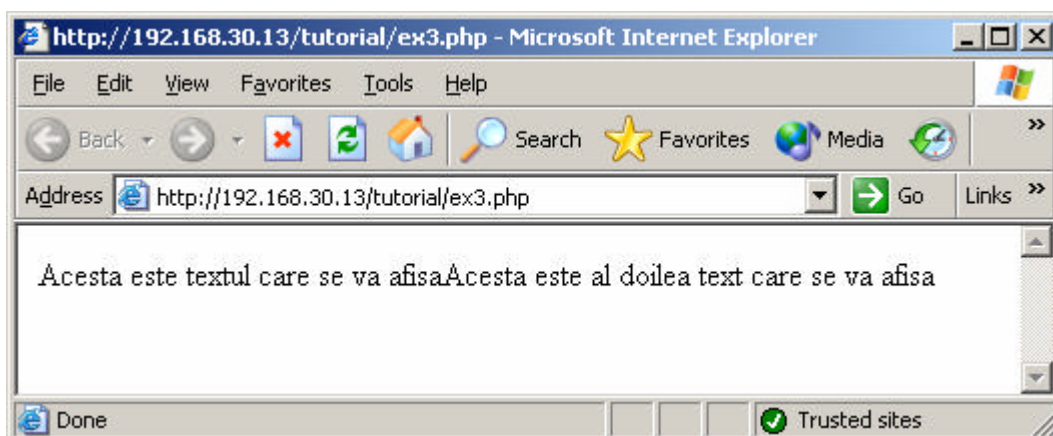
```
<?php
    echo "Acesta este textul care se va afisa";
?>
```

Dupa cum se observa lipsesc parantezele specifice functiilor. Asta se întâmpla deoarece echo nu este o functie ci doar un constructor al limbajului PHP. În schimb daca am folosi printf() atunci trebuie utilizate parantezele.

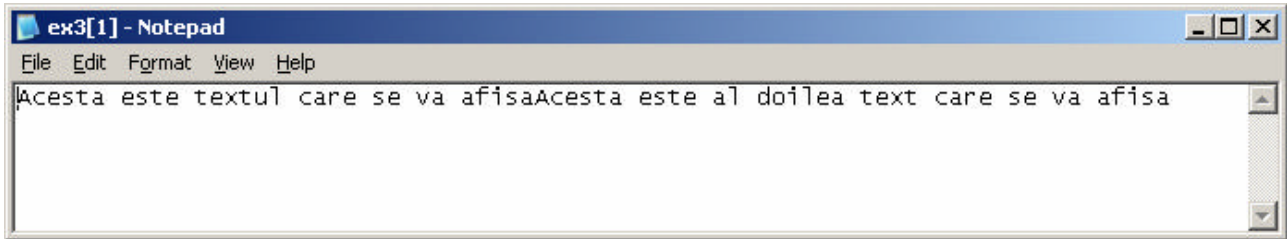
Exemplu 2

```
<?php
    echo "Acesta este textul care se va afisa";
    echo "Acesta este al doilea text care se va afisa";
?>
```

Daca salvam acest fisier si îl salvam pe server, obtinem urmatorul rezultat:



Codul sursa generat de php este:



```
ex3[1] - Notepad
File Edit Format View Help
Acesta este textul care se va afisaAcesta este al doilea text care se va afisa
```

Observam ca al doilea string a fost adaugat primului fara salt la rând nou cu toate ca am avut doua echo în codul php. Problema este ca dupa afisarea primului string cursorul imaginar al PHP-ului a ramas dupa primul string, al doilea echo fiind afisat începând cu acea pozitie.

Pentru a avea un salt la linie noua trebuie sa utilizam elemente de formatare

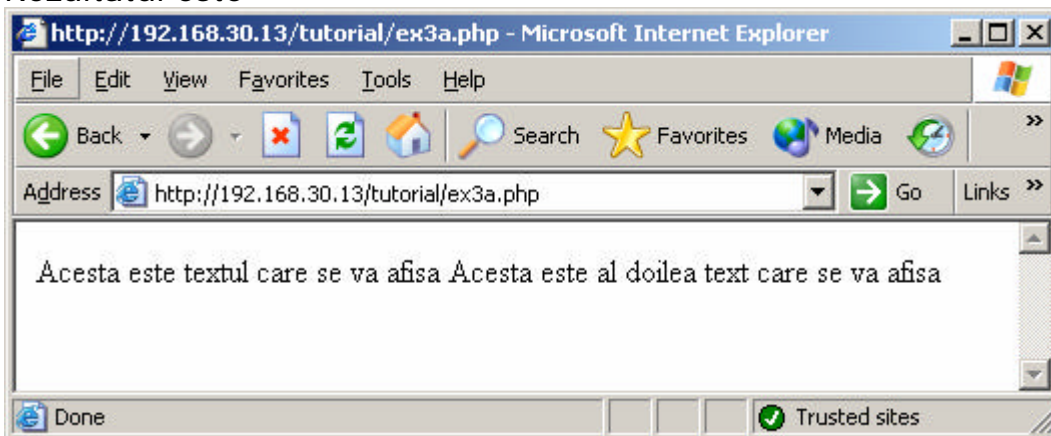
2. Elemente de formatare

Pentru a forta salt la rând nou în php utilizam string-ul `\n`

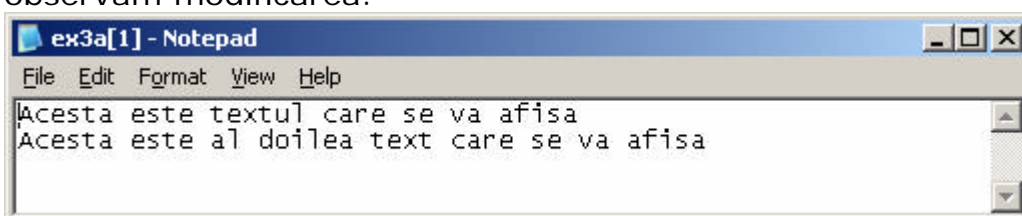
Exemplul 3

```
<?php
echo "Acesta este textul care se va afisa\n";
echo "Acesta este al doilea text care se va afisa";
?>
```

Rezultatul este



Adica cam tot ca la exemplul anterior. Totusi daca vedem codul generat de php observam modificarea.



```
ex3a[1] - Notepad
File Edit Format View Help
Acesta este textul care se va afisa
Acesta este al doilea text care se va afisa
```

În principiu utilizam elementele de formatare pentru lizibilitatea codului.

Totusi pentru a putea afisa si în fereastra browser+ului cele doua string-uri pe doua rânduri mai utilizam si elemente html.

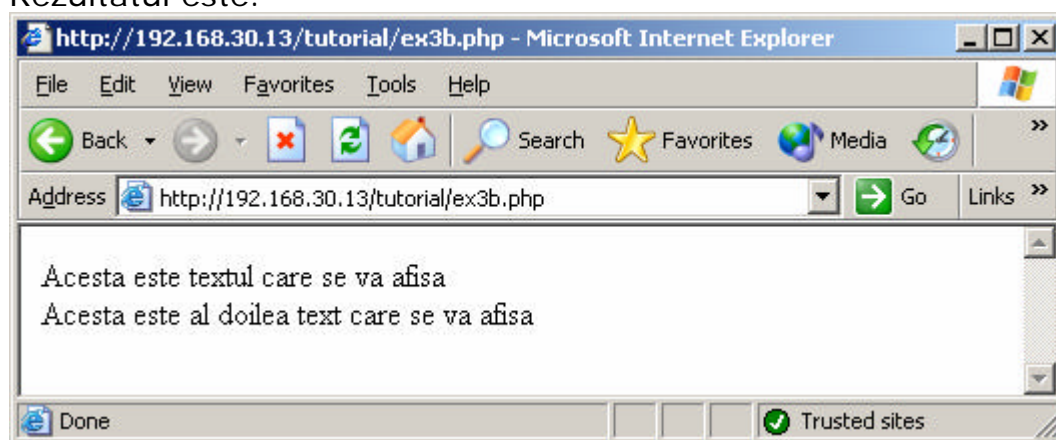
3. Afisarea elementelor html cu ajutorul lui php

Pentru a forta un salt la rând nou utilizam elementul html

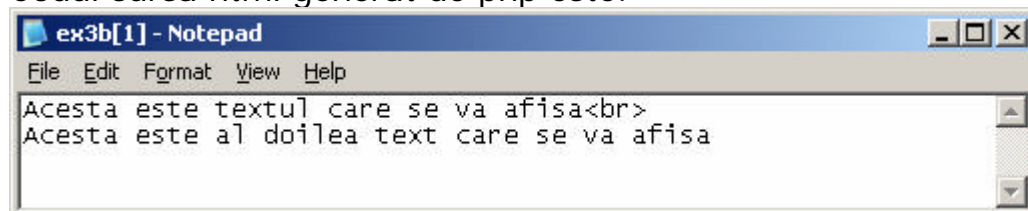
Exemplul 4

```
<?php
    echo "Acesta este textul care se va afisa<br>\n";
    echo "Acesta este al doilea text care se va afisa";
?>
```

Rezultatul este:



Codul sursa html generat de php este:



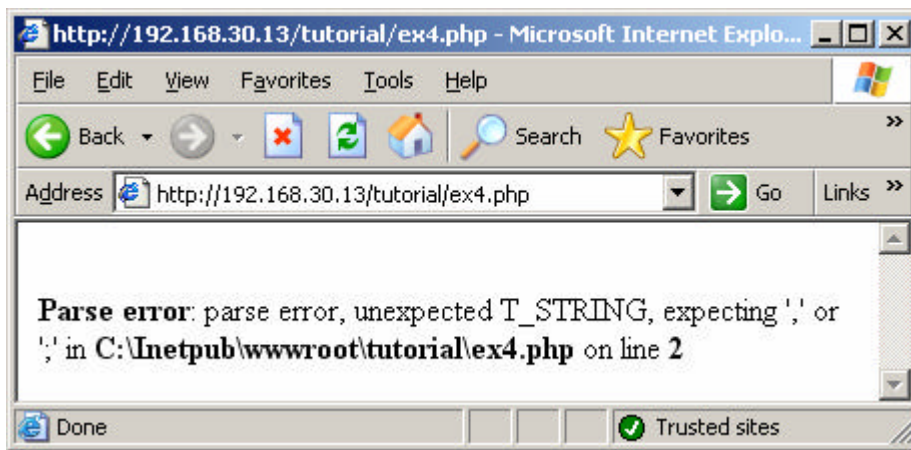
4. Escape-ul anumitor semne

Titul este ales nefericit. Ce vrea sa spuna este faptul ca anumite semne cum sunt ghilimelele, slash-ul etc sunt interpretate de catre php. Adica daca avem un text care ar contine ghilimelele am avea probleme la afisare sa.

Exemplu:

```
<?php
    echo "Acesta este \"textul\" care se va afisa<br>\n";
    echo "Acesta este al doilea text\scriis care se va afisa";
?>
```

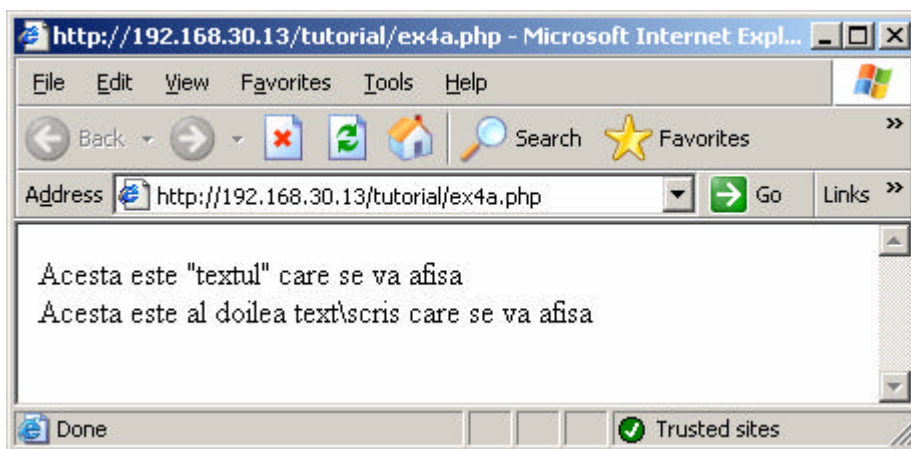
Daca vom lasa php-ul sa interpreteze acest script vom obtine un rezultat nesatisfacator:



Linia a doua contine mai mult de 2 ghilimele si acest lucru duce la eroarea afisata. Pentru a evita astfel de probleme trebuie sa „marcam” ghilimelele care sunt necesare în text si care nu fac parte din sintaxa. Acest lucru se face cu ajutorul escape-ului – se pune un \ înaintea ghilimelelor:

Codul ar trebui sa arate astfel:

```
<?php
    echo "Acesta este \"textul\" care se va afisa<br>\n";
    echo "Acesta este al doilea text\\s cris care se va afisa";
?>
```



Observam faptul ca pentru a afisa un \ a trebuit sa adaugam în codul sursa un al doilea \.

5. Afisarea semnelor speciale din PHP

Sunt cazuri în care am dori sa afisam semnul \$. În PHP acesta este un semn rezervat pentru variabile în sensul ca orice si de caractere care începe cu \$ este considerat nume de variabila.

Pentru a rezolva aceasta problema utilizam tot semnul de escape \

Exemplu:

```
<?php
    echo "Semnul de dolar \$ este o unitate monetara";
?>
```

Variabile in PHP

1. Declararea variabilelor
2. Atribuirea de valori
 - 2.1 Siruri de caractere
 - 2.1.1 Operator de concatenare
 - 2.2 Numere

1. Declararea variabilelor

Toate variabilele incep cu semnul de dolar \$ urmat de _ sau o litera. Variabilele sunt case-sensitive astfel incat \$ID nu este aceeași variabilă ca \$Id.

2. Atribuirea de valori unei variabile

Atribuirea unei valori (indiferent ca este vorba de string, integer, float, sau bool) se face cu operatorul de atribuire =.

2.1 Siruri de caractere

Exemplu 1:

```
<?php
    $xy = "Salutare!";
    $_10 = "Studenti";
?>
```

In ambele cazuri variabilelor \$xy si \$_10 li se atribuie cate un string. Acesta poate fi ulterior afisat cu echo.

2.1.1 Operator de concatenare

Pentru a uni doua siruri de caractere se utilizeaza operatorul "." (punct).

Exemplu 2:

```
<?php
    $variable = "Buna"." ziua!";
    echo $variable; // va afisa "Buna ziua!"
    echo "Salutare"."studenti"; // va afisa "Salutarestudenti"
?>
```

Se pot uni inclusive o variabilă de tip string cu un string sau doua variabile de tip string.

Exemplu 3:

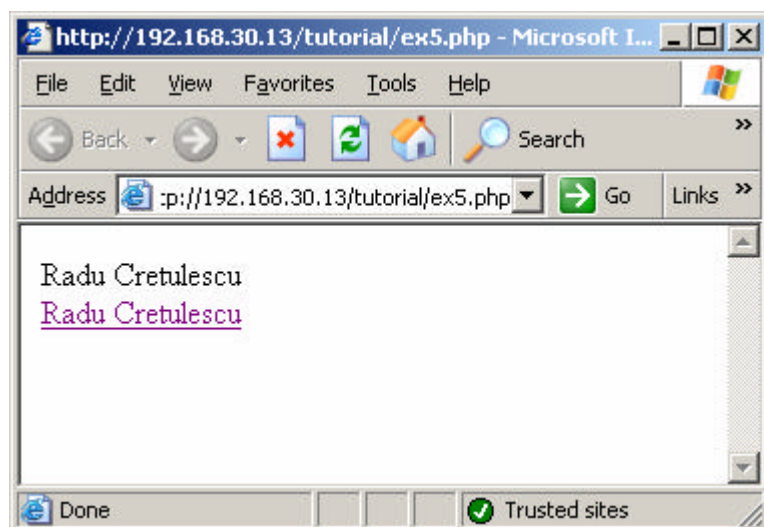
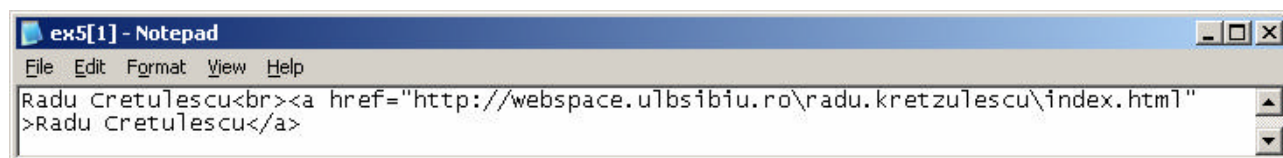
```
<?php
$bla = "Mama";
$name = $bla." mea"; // va uni continutul lui $bla + " mea" (cu spatiu)
echo $name;          // va afisa "Mama mea"
echo $name." si a ta";// va afisa "Mama mea si a ta"
?>
```

Intr-o expresie se pot utiliza mai multi operatori de concatenare. Oricum nu se vatine cont daca se unes string-uri sau variabile.

Exemplu 4:

```
<?php
$prenume = "Radu";
$nume = "Cretulescu";

$Nume = $prenume." ".$nume; // Variabilele contin siruri de caractere si
                             //trebuie sa adaugam manual un spatiu
echo $name;                  // Va afisa "Radu Cretulescu"
echo "<a href=\"http://webpace.ulbsibiu.ro\radu.kretzulescu\rindex.html\">"
.$prenume." ".$nume."</a>\n" ;
?>
```



2.2 Numere

2.2.1 Tipuri de variabile

2.2.2 Numere int

2.2.3 Numere float

2.2.1 Tipuri de variabile

In PHP se cunosc doua tipuri de numere: intergi (int) si in virgula mobila (float sau double)

2.2.2 Numere intregi

La atribuirea valorii unei variabile se stabileste automat tipul variabilei. Daca dorim sa definim o variabila int atunci ii atribuim direct o valoare intrega.

Exemplu 1:

```
<?php
    $valoare = 3;
    echo "Numarul ales este ".$valoare."\n";
    echo 100;
?>
```

2.2.3 Numere float

Numerele care contin parte zecimala sunt de tip float

```
<?php
    $g = 9.81;
    echo "Constanta gravitacionala g= ".$g." unitatea de
masura\n";
?>
```

La acest tip de numere este important sa retinem ca separatorul zecimal este semnul punct, semnul virgula se utilizeaza pentru separarea parametrilor unei functii

Exemplu 2:

```
<?php
    xyz(5.6); // 5.6 este un parametru de tip float al functiei xyz
    abc(5,6); // 5 si 6 sunt doi parametric ai functiei abc
    abc(5, 6); // ca si mai sus
?>
```

Mai apare inca o problema: Punctul este si operatorul de concatenare. Daca am dori sa unim doua numere avem urmatoarele posibilitati:

1. Atribuim mai intai numerele unor variabile si apoi unim acestea cu operatorul punct.

```
<?php
    $unitati = 4;
    $zeci = 5;
    echo $zeci.$unitati;
?>
```

2. Daca dorim sa unim doua valori intregi cu ajutorul operatorului punct atunci trebuie sa inseram cate un spatiu inainte si dupa operator.

```
<?php
    echo 4 . 5; // Afiseaza 45
```



```
echo 4.5; // Afiseaza 4.5
?>
```

Operatori PHP

Operatoii aritmetici

Operator	Descriere	Exemplu	Rezultat
+	Adunare	x=2 x+2	4
-	Scadere	x=2 5-x	3
*	Înmultire	x=4 x*5	20
/	Împartire	15/5 5/2	3 2.5
%	Modul (rest)	5%2 10%8 10%2	1 2 0
++	Incrementare	x=5 x++	x=6
--	Decrementare	x=5 x--	x=4

Operatori de atribuire

Operator	Exemplu	Corespunde
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Operatori de comparare

Operator	Descriere	Exemplu
==	este egal cu	5==8 returneaza false
!=	nu este egal cu	5!=8 returneaza true
>	este mai mare ca	5>8 returneaza false
<	este mai mic ca	5<8 returneaza true
>=	este mai mare sau egal ca	5>=8 returneaza false
<=	este mai mic sau egal ca	5<=8 returneaza true

Operatori logici

Operator	Descriere	Exemplu
&&	si	x=6 y=3 (x < 10 && y > 1) returneaza s true
	sau	x=6 y=3 (x==5 y==5) returneaza false
!	not	x=6 y=3 !(x==y) returneaza true

Probleme:

1. Dati un exemplu de script în care declaram un string
2. Dati un exemplu de utilizare a escape-ului pentru anumite semne