

Sesiuni – PHP

Sesiunile, reprezinta o functionalitate prin care anumite informatii sunt mentinute de la o pagina la alta.

O sesiune dureaza atat timp cat utilizatorul acceseaza un site si se incheie odata cu inchiderea browserului.

Sesiuni PHP - reprezinta perioada de timp in care mai multe scripturi PHP, accesate la momente diferite de timp, pot stoca si folosi informatii comune.

O sesiune incepe atunci cand un script apeleaza functia `session_start` si se termina cand utilizatorul inchide browserul.

Sesiunile sunt folosite pentru a transporta date de la o pagina la alta.

Sesiunile **nu sunt mai sigure ca si cookie-urile** (deoarece unele servere nu sunt securizate destul de bine si ceilalti utilizatori cu care se imparte serverul poate vedea id-ul sesiunilor).

Pentru securizarea sesiunilor putem utiliza un `hash...`

Exemplu salvare si afisare date cu ajutorul sesiunilor

`Ex_sesiune.php`

```
<?php
```

```
session_start();//functia trebuie apelata in primele linii din fisier pentru a nu se trimite nici macar un caracter la browser.
```

```
$_SESSION['curs'] = 'Tehnologii Web';//
```

```
$_SESSION['orar'] = '12-14';
```

```
$_SESSION['laborator'] = '14-16';
```

```
echo '<b>'.print_r($_SESSION,true).'
```

```
?>
```

Array ([curs] => Tehnologii Web [orar] => 12-14 [laborator] => 14-16)

Exemplu 2 pentru a afla valoarea unei variabile din sesiune folosind `$_SESSION[index]`

```
<?php
session_start();
if(isset($_SESSION['curs'])) ){
    // folosim $_SESSION[index] pentru a afla valoarea unei variabile din sesiune
    echo "Cursul <b>{$_SESSION['curs']}</b> este de la ora <b>{$_SESSION['orar']}</b> si
laboratorul este la ora <b>{$_SESSION['laborator']}</b>! ";
} else{ echo "Trebuie sa accesezi mai intai ex_sesiune.php!";
}
?>
```

functia **session_destroy()** permite **sergerea unei sesiuni**; Stergerea unei variabile din sesiune se realizeaza folosind **unset()**

```
1 <?php
2 session_start();
3 unset($_SESSION[curs]);
4 ?>
```

<https://www.php.net/manual/ro/function.session-destroy.php>

Ce este XML

Documentele XML sunt realizate din unitati de stocare numite entitati, ce contin date parsate sau neparsate. Datele parsate sunt realizate din caractere, unele dintre ele formand date caracter iar altele ca marcate. Marcatele codifica o descriere a schemei de stocare a documentului si structura logica. XML furnizeaza un mecanism pentru a impune constrangeri asupra schemei de stocare si a structurii logice.

XML - eXtensible Markup Language- este un limbaj pentru crearea altor limbaje.

Are o structura bine definite.

Poate fi utilizat pentru a crea limbaje de marcare precum HTML, XHTML

Scopul limbajului XML:

- XML a fost elaborat pentru:
- Separarea sintaxei de semantica pentru a furniza un cadru comun de structurare a informatiei
- construirea de limbaje de mark-up pentru aplicatii din orice domeniu structurarea informatiei in viitor
- asigurarea independentei de platforma si suport pentru internationalizare

XML vs. HTML

Exemplu HTML

```
<html>
<body>
<h2> Student</h2>
<p>Informatii
</p>
</body>
</html>
```

HTML -

- Specifica modul de randare a documentului, si nu ce tip de informatie este continuta in document
- Este dificil pentru o masina sa extraga informatia continuta in pagina. Este relativ simplu pentru om

XML vs. HTML

Sa consideram acum urmatoarea reprezentare

```
<contact>
  <name>Student</name>
  <address>Bucuresti</address>
  <web-page>http://www.utm.ro</web-page>
  <email>
    <email1>...</email2>
    <email2>...</email2>
  </email>
</contact>
```

In acest caz

- Informatia continuta este pusa in evidenta si nu modul de randare
- Continutul este separat de prezentare

- Informatia poate fi "inteleasa" atat de oameni cat si de masini

Concluzii:

- HTML este utilizat pentru a marca textul astfel incat el sa poata fi afisat
- XML este utilizat pentru **marcarea datelor** astfel incat ele sa poata fi procesate automat de calculator
- HTML descrie atat structura (ex. <p>, <h2>,) cat si modul de reprezentare (ex. , , <i>)
- XML descrie numai continutul sau "intelesul"
- HTML utilizeaza un set fix, neschimbat, de tag-uri
- In XML putem sa **definim propriile tag-uri**

Avantajele utilizarii limbajului XML

- Este un limbaj bazat standard open
- Este un limbaj extensibil
- Este transformabil
- Disponibilitatea unui numar mare de unelte de dezvoltare

XML a fost elaborat pentru:

- separarea **sintaxei** de **semantica** pentru a furniza un cadru comun de structurare a informatiei
- **construirea de limbaje de mark-up** pentru aplicatii din orice domeniu
- **structurarea informatiei** in viitor
- **asigurarea independentiei** de platforma si suport pentru **internationalizare**

Un document XML este un **arbore ordonat etichetat**:

- **date caracter** - noduri frunza ce contin datele
- noduri **elemente** etichetate cu
 - un nume (adesea numit si tipul elementului) si
 - o multime de attribute, fiecare din ele avand un nume si o valoare

acestea pot contine unu sau mai multi copii.

Structura documentelor XML

Un document XML este format din:

- marcaje (tag-uri)
- date caracter

Un *marcaj (tag)* este un sir de caractere delimitat de caracterele "<" si ">". *Datele caracter* reprezinta continutul marcajelor.

Un fisier XML cuprinde urmatoarele sectiuni:

- Prolog
- Definitia tipului de document (optionala)
- Elementul radacina

Exemplu: Fisierul mail.xml

```
<?xml version="1.0" encoding="UTF-8"?>//php
<!DOCTYPE MAIL SYSTEM "MAIL.DTD">
<MAIL id="10" date="10-05-2021">
  <FROM>nume_prof@prof.utm.ro</FROM>
  <TO>student@s.utm.ro</TO>
  <SUBJECT>Hello</SUBJECT>
  <MESSAGE>
    Hello Student
  </MESSAGE>
</MAIL>
```

Prologul:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Este o instructiune de procesare. Ea informeaza ca urmeaza descrierea unui fisier XML ce respecta versiunea de specificatie 1.0 iar setul de caractere utilizat este encodat UTF-8.

Definitia Tipului de Document:

```
<!DOCTYPE MAIL SYSTEM "MAIL.DTD">
```

Precizeaza ca fisierul MAIL.DTD contine declaratia tipului de document (DTD-ul), document ce are ca radacina tag-ul MAIL. Acesta este un set de reguli ce defineste structura unui fisier XML.

Elementul radacina:

```
<MAIL id="10" date="10-05-2020">  
  <FROM>profesor@prof.utm.ro</FROM>  
  <TO>student@s.utm.ro</TO>  
  <SUBJECT>Hello</SUBJECT>  
  <MESSAGE>  
    Hello Student  
  </MESSAGE>  
</MAIL>
```

Un document XML are un singur element radacina.

Un document XML poate contine urmatoarele tipuri de marcate:

- Elemente
- Atribute
- Comentarii
- Entitati
- Sectiuni CDATA
- Instructiuni de procesare
- Declaratia tipului de document

Elementele:

```
<nume_tag> ..... </nume_tag>
```

- sunt blocurile de baza ale unui document XML
- Retin informatii sau definesc structura
- Fiecare element are un tag de inceput si unul de sfarsit
- Elementele pot fi imbricate

Elemente vide:

```
<nume_tag/>
```


Exemplu

```
<?xml version="1.0"?>
<BIBLIOTECA>
<CARTE>
<TITLU>XML </TITLU>
<AUTOR>Eminescu</AUTOR>
<EDITURA>Art Educational</EDITURA>
<AN_APARITIE>2019</AN_APARITIE>
</CARTE>
</BIBLIOTECA>
```

Se observa ca

- elementele <TITLU>, <AUTOR>, <EDITURA>, <AN_APARITIE> contin informatii
- <BIBLIOTECA>, <CARTE> sunt folosite doar pentru a defini structura datelor

Atributele:

```
<nume_tag numeAtr1="val1" ... numeAtrN="valN">
    . . .
</nume_tag>
```

- sunt localizate in tag-ul de start al unui element
- au rolul de a descrie elementele
- adesea contin metadate, ex: id

```
<?xml version="1.0"?>
<BIBLIOTECA>
<CARTE cota="12345">
<TITLU>XML</TITLU>
<AUTOR>Eminescu</AUTOR>
<EDITURA>Art Educational</EDITURA>
<AN_APARITIE>2019</AN_APARITIE>
</CARTE>
</BIBLIOTECA>
```

- cota="12345" este un atribut

Comentarii:

```
<!-- comentariu -->
```

- sunt secvențe de caractere ce pot apărea oriunde în document ignorate la parsare
- Ele nu fac parte din datele caracter ale documentului

Exemplu

```
<?xml version="1.0"?>
<!-- Documentul retine cartile dintr-o biblioteca -->
<BIBLIOTECA>
<CARTE cota="12345">
<!-- titlul cartii -->
<TITLU>XML </TITLU>
<!-- Autorul Cartii -->
<AUTOR>Nume Autor</AUTOR>
<!-- Editura in care a aparut cartea -->
<EDITURA> Editura care a publicat cartea</EDITURA>
<!-- Anul de aparitie a cartii -->
<AN_APARITIE>2019</AN_APARITIE>
```

</CARTE>
</BIBLIOTECA>

Important

Tag-uri sunt *case sensitive*, adica se face distinctia intre litere mari si litere mici.

De exemplu, urmatoarele exemple de taguri sunt gresite:

<Student>NUME</STUDENT>
<StudentT>NUME</student>

Numele unui tag este o succesiune de caractere alfa-numerice ce *incepe obligatoriu cu o litera*. Astfel <7nume> este eronat.

Referinte la entitati

Referintele la entitati sunt de fapt pointeri catre entitati. În XML, entitatile sunt unitati de text, unde o unitate poate fi orice, de la un singur caracter la un intreg document sau chiar o referinta la un **alt document**.

Sintaxa referintelor la entitati este:

&*nume_entitate*;

‘&’, urmat de numele entitatii, urmat de ‘;’

Una dintre cele mai frecvente utilizari ale referintelor la entitati este atunci cand se doreste folosirea unor caractere care ar duce la aparitia unor confuzii pentru analizorul XML si deci care nu ar trebui sa apara in forma lor normala in text. În acest caz exista cinci entitati predefinite in XML:

Table 1. Entitati definite in XML

Entitate	Referinta la entitate
<	<
>	>
&	&
'	'
"	"e;

În momentul in care analizorul XML intalneste referinta la o entitate in document, el o va substitui cu datele pe care aceasta le refera si va returna documentul cu inlocuirile facute.

Exemplu:

```
<TITLE>Tom &amp; Jerry</TITLE>
```

dupa analizarea textului de catre analizorul XML, va rezulta:

Tom & Jerry

O alta utilizare frecventa a referintelor la entitati este in cazul in care avem in documentul XML fragmente de text care se repeta. Pentru a nu scrie aceste parti de text de mai multe ori vom defini o entitate care va avea ca valoare acea parte de text si de fiecare data cand fragmentul respectiv apare in document vom folosi referinta la entitate.

Prin folosirea referintelor la entitati se vor obtine documente mai scurte si se va scurta timpul de redactare.

Instructiuni de prelucrare

Instructiunile de prelucrare sunt un tip special de marcaj care contin informatii despre anumite aplicatii ce urmeaza a fi executate. Sintaxa generala a unei instructiuni de procesare ar fi urmatoarea:

```
<?aplicatieinstructiune="valoare" ?>
```

Încep cu <?, urmat de numele aplicatiei si de specificarea unor elemente ce tin de acea aplicatie si se incheie cu ?>. Numele aplicatiei trebuie sa fie diferit de xml sau XML, sau alte moduri de scriere a acestui cuvânt, deoarece cuvintele de acest tip sunt rezervate, urmand a fi standardizate intr-o versiune ulterioara.

Sectioni CDATA

Sectionile CDATA sunt utilizate pentru a include blocuri de text continand caractere care altfel ar fi recunoscute ca marcaje. Sectionile CDATA incep cu sirul <![CDATA[si se termina cu sirul]]>.

Sectionile CDATA sunt folosite in general atunci cand dorim ca datele incluse in interiorul lor sa nu fie interpretate de catre analizor, ci sa fie considerate date caracter.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<exemplu>
```

Un exemplu de creare a unui tabel in HTML:

```
<![CDATA[
  <table align="center">
    <tr>
      <td>Coloana 1</td>
      <td>Coloana 2</td>
    </tr>
  </table>
```

```
]]>
```

```
</exemplu>
```

Folosind secțiunea CDATA, analizorul va ignora conținutul acesteia și datele vor fi expuse utilizatorului exact în forma în care sunt, și datele nu vor fi interpretate drept marcaje, ci drept date caracter.

O restricție de sintaxă este faptul că în interiorul secțiunilor CDATA nu poate să apară șirul ']]'. Încă un lucru de reținut este că secțiunile CDATA nu pot fi incluse unele în altele.

```
<?xml version="1.0" encoding="UTF-8"?>
<exemplu>
Un exemplu de creare a unui tabel în HTML:
  <![CDATA[
    <table align="center">
      <tr>
        <td>Coloana 1</td>
        <td>Coloana 2</td>
      </tr>
    </table>
  ]]>;
</exemplu>
```

Parsarea XML-urilor este o parte integranta a dezvoltarii web,

In php exista o biblioteca cunoscuta sub numele de **simpleXML**, ale carei facilitati sunt mai mult decat suficiente pentru a acoperi procesarea de baza a XML-urilor.

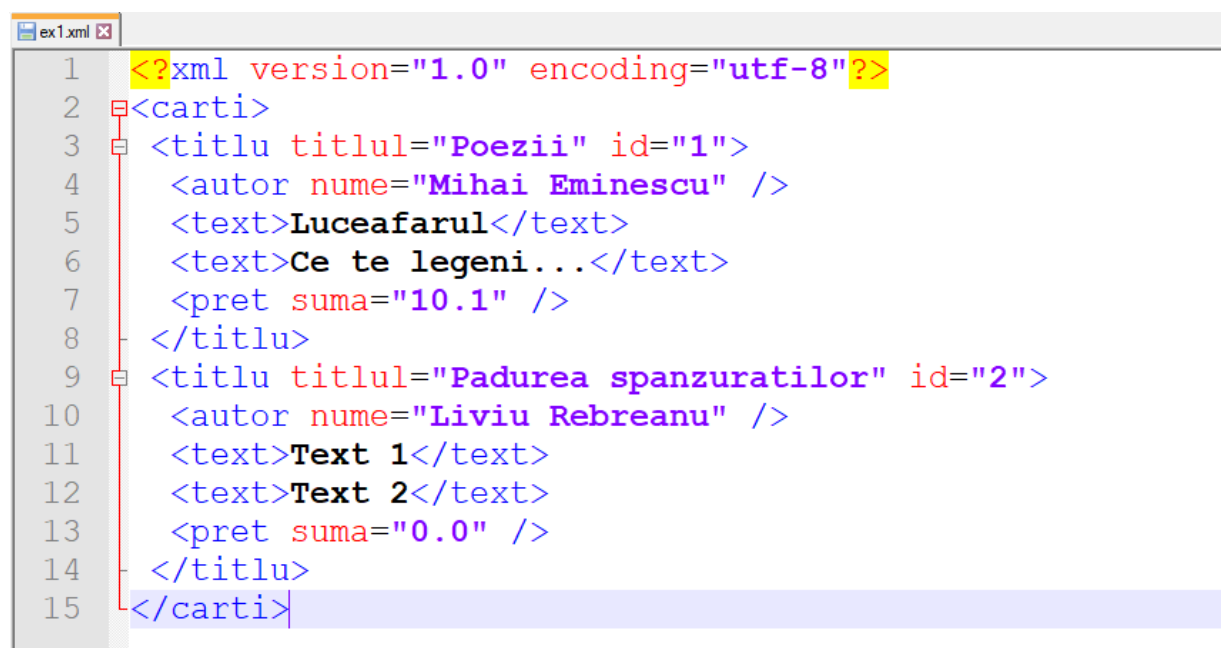
PHP - XML - SimpleXML

SimpleXML a aparut in PHP 5. Lucreaza ca si DOM, cu obiecte, preia tot documentul XML sub forma unui arbore ierarhic in memorie, dar spre deosebire de acesta, e mai flexibil si foloseste mai putina memorie deoarece elementele sunt stocate direct ca variabile PHP (de tip string si array) si astfel pot fi imediat utilizate. Foloseste un minim necesar de cod si are o forma intuitiva a datelor.

*In SimpleXML se lucreaza mult cu functii pentru Array, majoritatea datelor sunt stocate in variabile **de tip array**.*

Este util cand se doreste citirea catorva date dintr-un document XML, a carui structura o cunoasteti, si scrierea altor informatii/date.

Citire document XML cu SimpleXML



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <carti>
3   <titlu titlul="Poezii" id="1">
4     <autor nume="Mihai Eminescu" />
5     <text>Luceafarul</text>
6     <text>Ce te legeni...</text>
7     <pret suma="10.1" />
8   </titlu>
9   <titlu titlul="Padurea spanzuratilor" id="2">
10    <autor nume="Liviu Rebreanu" />
11    <text>Text 1</text>
12    <text>Text 2</text>
13    <pret suma="0.0" />
14  </titlu>
15 </carti>
```

Urmatorul script preia si afisaza date din acest document XML

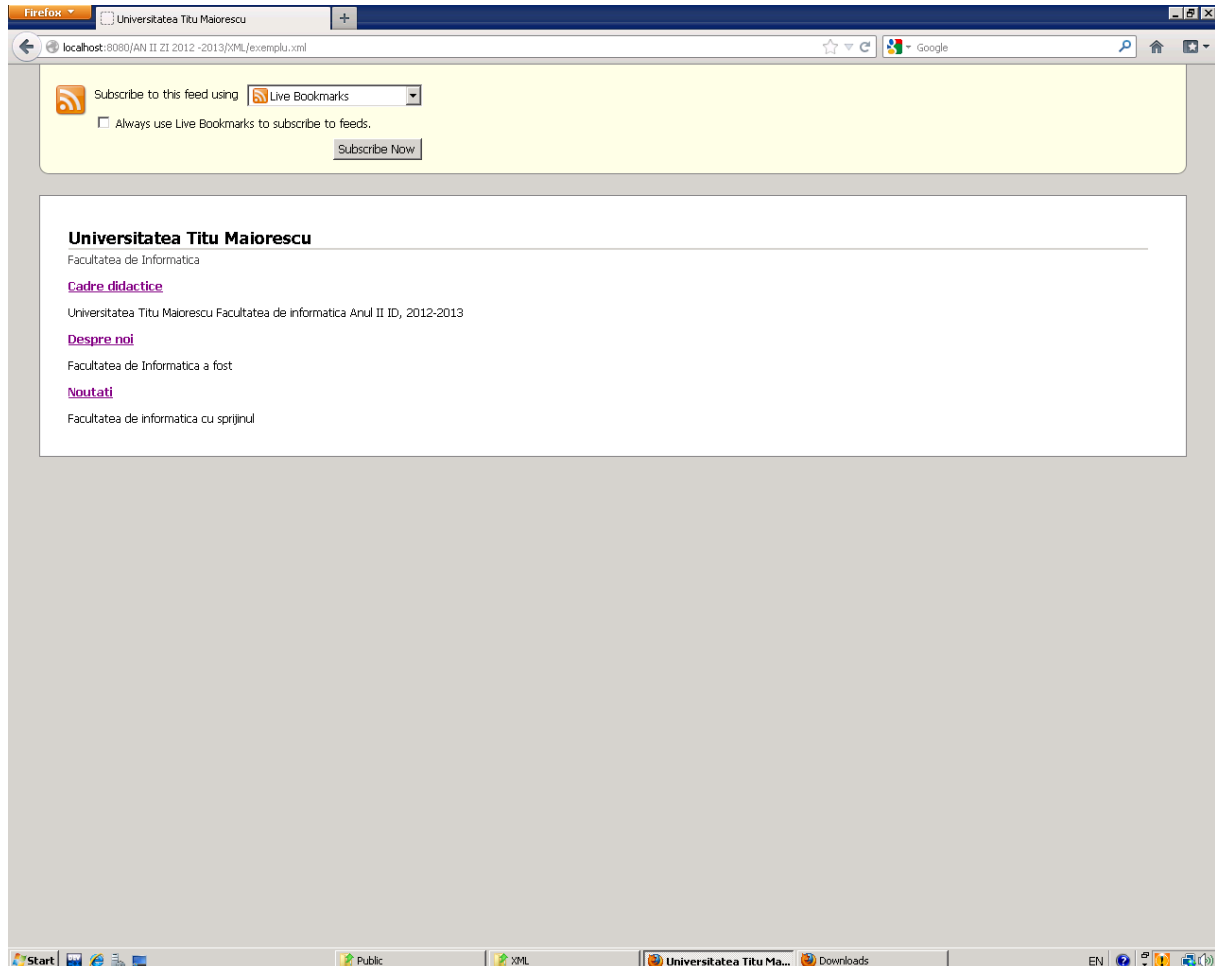
```
afisare1.php
1 <?php
2 $obj = simplexml_load_file("ex1.xml");
3 $titlu = $obj->titlu;
4 $text = $obj->titlu[0]->text;
5 // preia intr-un array elementele "autor" si "pret"
6 for($i=0; $i<count($titlu); $i++) {
7     $autor[] = $titlu[$i]->autor;
8     $pret[] = $titlu[$i]->pret;
9 }
10 echo $autor[0]['nume']. '<br >';
11 echo $pret[1]['suma']. '<br> Afisare text...<br />';
12
13 for($i=0; $i<count($text); $i++) {
14     echo $text[$i]. ' <br /> ';
15 }
16 ?>
17
```

Sa se afiseze intreg continutul fisierului **ex1.xml**

Modificare document XML cu SimpleXML

```
modificare1.php
1 <?php
2 $obj = simplexml_load_file("ex1.xml");
3 $titlu = $obj->titlu;
4 // Preia toate elementele cu nume "titlu" (intr-o variabila tip array)
5 // Parcurge matricea cu $titlu si adauga, cu "addChild()" inca un element stoc cu val 1
6 for($i=0; $i<count($titlu); $i++) {
7     $titlu[$i]->addChild('stoc', 1);
8 }
9 // Modifica valoarea atributului 'nume' din elementul "autor" al primul element "titlu"
10 $titlu[0]->autor['nume'] = 'Nume Modificat';
11
12 // Folosind sistemul de lucru cu array, creaza inca un element "autor" )
13 $titlu[0]->autor[1] = 'Alt autor';
14 $titlu[0]->autor[1]['atribut'] = 'Atribut_adaugat';
15 // Adauga datele XML intr-un sir
16 $xml_doc = $obj->asXML();
17 echo htmlentities($xml_doc);
18 ?>
19
```

Exemplu cu RSS



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <rss version="2.0">
3   <channel>
4     <title>Universitatea Titu Maiorescu</title>
5     <description>Facultatea de Informatica</description>
6     <link>http://www.utm.ro</link>
7     <language>ro</language>
8     <item>
9       <title><![CDATA[ Informatii Secretariat ]]> </title>
10      <description><![CDATA[ Universitatea Titu Maiorescu Facultatea de informatica Anul II ZI, 2020-2021]]>
11    </description>
12    <link><![CDATA[ https://www.utm.ro/facultatea-de-informatica/informatii-de-secretariat/]]>
13  </link>
14  </item>
15  <item>
16    <title><![CDATA[ Noutati ]]> </title>
17    <description><![CDATA[ Facultatea de Informatica a fost ]]> </description>
18    <link><![CDATA[ https://www.utm.ro/facultatea-de-informatica/noutati/]]>
19  </link>
20  </item>
21  <item>
22    <title><![CDATA[ Avizier ]]></title>
23    <description><![CDATA[ Facultatea de informatica cu sprijinul ]]>
24  </description>
25    <link><![CDATA[ https://www.utm.ro/facultatea-de-informatica/avizier-studii-zi/]]>
26  </link>
27  </item>
28  </channel>
29 </rss>
```



```
test.php
1 <?php
2 // incarca fisierul exemplu.xml
3 if(!$xml = simplexml_load_file('exemplu.xml')){
4     trigger_error('Eroare la citirea fisierului', E_USER_ERROR);
5 }
6
7 foreach($xml as $channel){
8     echo $channel->title."<br>";
9     echo $channel->link;
10 }
11 ?>
```

```
test 1.php
1 <?php
2 // incarca fisierul exemplu.xml
3
4 if(!$xml = simplexml_load_file('exemplu.xml')){
5     trigger_error('Eroare ', E_USER_ERROR);
6 }
7
8 // afiseaza titlul primului element
9
10 echo '<div style=color:red>Elementele sunt :<br />';
11 for($i=0;$i<3;$i++){
12 {
13
14     echo $xml->channel->item[$i]->title.'';
15     } echo '</div>';
16 }
17 ?>
```

Functii SimpleXML

- **simplexml_load_file("fisier.xml")** - Incarca in memorie, ca obiect, datele din "fisier.xml"
- **simplexml_load_string("sir_xml")** - Incarca in memorie, ca obiect, datele din sirul "sir_xml"
- **simplexml_import_dom("dom_node")** - Transforma (preia) un obiect DOM (sau Nod dintr-un obiect DOM) in obiect SimpleXML
- **addChild("nume", "continut")** - Aadauga un element copil in cel curent (la sfarsit daca are si altele) cu numele "nume" si continutul text "continut" (acesta e optional).
- **addAttribute("nume", "valoare")** - Aadauga un atribut intr-un element
- **children()** - Preia intr-o matrice toti copii dintr-un nod (element).
- **attributes()** - Preia intr-o matrice toate attributele dintr-un element.
- **count()** - Returneaza numarul de copii dintr-un element.
- **getName()** - Returneaza numele unui element

- **asXML("fisier.xml")** - Transforma datele obiectului SimpleXML intr-un sir, iar daca parametrul "fisier.xml" e specificat (*e optional*) scrie sirul in acel fisier.

Crearea unui fisier XML

```

1 <?php
2 //creare document XML
3 $document = new DomDocument('1.0', 'utf-8');
4 $r = $document->createElement('html');
5 $r = $document->appendChild($r);
6 $body = $document->createElement('body');
7 $body = $r->appendChild($body);
8 $body->setAttribute('bgcolor', '#ffff00');
9 $paragraff = $document->createElement('p');
10 $paragraff = $body->appendChild($paragraff);
11 $text = $document->createTextNode('Facultatea de informatica');
12 $text = $paragraff->appendChild($text);
13 //salveaza fisier si testeaza salvarea pe server
14 if($document->save("exemplu_d.xml"))
15 echo 'Documentul exemplu_d.xml a fost creat';
16 else echo 'Eroare: nu a putut fi creat documentul exemplu_d.xml ';
17 ?>
18

```

Citire fisier XML, si crearea unui nou fisier modificat

```

1 <?php
2 $file = 'exemplu_d.xml';
3 // Calea si numele fisierului .xml
4 $doc = new DOMDocument();
5 // Creaza un nou obiect in memorie
6 $doc->load($file);
7 // Incarca datele din $file in obiectul nou creat
8 $get_elms = $doc->getElementsByTagName("*");
9 // Preia toate elementele ("*") stocate in obiect
10 $nr_elms = $get_elms->length;
11 for($i = 0; $i<$nr_elms; $i++) { | $node = $get_elms->item($i);
12 // Preia fiecare Nod din parcurgere
13 $element = $node->nodeName;
14 if($element=='p') {
15 $node->nodeValue = 'MODIFICARE CONTINUT';
16 // Creaza noul element dupa 'p' (in elementul Parinte unde se afla si 'p'),
17 // cu nume si continut text
18 $element_nou = $doc->createElement('div', 'NOUL ELEMENT ADAUGAT');
19 $node->parentNode->appendChild($element_nou);
20 }
21 }
22 if($doc->save('exemplu2_dom.xml')) {
23 echo htmlentities($doc->saveXML());
24 }
25 ?>

```

Tema:

- Sa se testeze toate aplicatiile din curs.

```
<form name="f1" method="post" enctype="multipart/form-data" action="introducere.php">
Nume: <input type="text" name="t1" class="c2"><br>
Prenume:<input type="text" name="t2" class="c2"><br>
Poze: </div></td><td><input type="file" name="t6" class="c2">
```

```
<?php
$uploadaddir="abc/";
$a=$_POST["t1"];
$b=$_POST["t2"];
$f=$_FILES["t6"]["name"];
$con = mysqli_connect("localhost", "root", "", "utm") or die("conectare nereusita");
mysqli_select_db($con,"utm") or die("eroare la baza de date");
$uploadfile=$uploadaddir.$f;
if(move_uploaded_file($_FILES["t6"]["tmp_name"],$uploadfile))
print "fisierul ".$f." s-a incarcat cu succes<br>";
else print "eroare";
echo"<a href='menu.html'>Click aici pentru a va intoarce la MENU</a>";
//echo "Ati introdus ".$a." ".$b." ".$c." ".$d." ".$e;
$x="insert into prieteni set nume='".$a."',prenume='".$b."',poze='".$f.'";";
$y=mysqli_query($con,$x);
if($y)
{
    echo "datele s-au introdus cu succes<br><br>";
}
else
    echo "eroare la adaugare";
```

Meeting in "Tehnologii web"

03:01:53

People

Chat

Reactions

Rooms

More

Camera

Mic

Share

Leave

IC

Iancu C...

Amaritei AL...

SV

Sterea ...

CS

Creang...

Cenusu ...

RL

Robescu Cl...

VM

Voicu Remy...

IR

Ionescu Ele...

Marin Vlad ...

+17

MP

Participants

Type a name

Share invite

In this meeting (26)

DA

Dragnea Nicolae Alexan...

1

SV

Sterea Victor

2

Cenusu Dorinel

3

CS

Creanga Stefan

4

IC

Iancu Catalin Cristian

5

HP

Handra Florin Petru

6

Dumitru Vlad

7

MP

Mironela Pirnau

Organiser

AB

Al Raevs Ahmed Basir

ENG

INTL

13:03

10/05/2022