Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek
Sveučilište u Zagrebu
Fakultet organizacije i informatike
Pavlinska 2, Varaždin 42000
dkermek@foi.hr

22. dio

Programiranje na strani poslužitelja

Smarty predlošci

XML, SAX, DOM, XSL, XSLT, RSS

PDF, Grafika

XML

- dobro (ispravno) formiran dokument - prema XML spacifikaciji (zatvaranje elemenata, atributi u navodnicima, dokument s najmanje jednim elementom)
- validan dokument – ispravno formiran i zadovoljava zahtjeve i specifikacije iz DTD ili XML scheme

```
<?xml version="1.0"?>
<bookmarks category="News">
      <link id="15696">
            <title>CNN</title>
            <url>http://www.cnn.com/</url>
            <last_access>2000-09-08</last_access>
      </link>
      <link id="84574">
            <title>Slashdot</title>
            <url>http://www.slashdot.com/</url>
            <last_access>2001-12-30</last_access>
      </link>
</bookmarks>
```

XML dokument

- prolog dokumenta
- elementi
- atributi
- entiteti
- notacije
- CDATA blokovi
- instrukcije procesiranja
- komentari

Primjer XML dokumenta

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE invoice
[
<!ENTITY logo SYSTEM "images/logo.gif" NDATA gif>
<!NOTATION gif SYSTEM "gifviewer.exe">
]>
<invoice id="AS2354R">
     <logo src="logo" />
     <date>03-12-2000</date>
     <!-- consignor address -->
     <vendor>
          <name>Bobby's Biotech Bazaar</name>
          <address>
               <street>25, Main Street</street>
          </address>
     </vendor>
     <material>
        <item id="23">
               <name>Vampire bat (DNA sequence)</name>
               <quantity>1</quantity>
               <price currency="USD">34.99</price>
               <dna_sequence>
                    <![CDATA[
                    1627 3494 #190 2230 8549 0732 #232 8238 2833 01&6 9929
                    ]]>
               </dna_sequence>
          </item>
 <?sales_agent include_promotional_message="1"?>
</invoice>
```

Korištenje XML dokumenta

- **Obrada lokalne datoteke-dokumenta ili primljenih podataka-dokumenta od poslužitelja:**
  - **SAX – Simple API for XML**
  - **DOM – Document Object Model**

SAX – Simple API for XML

- Obrada dokumenta temelji se na događajima
- Postoje funkcije za rad s XML dokumentom:

```
xml_parser_create();
xml_set_element_handler(...);
xml_set_character_data_handler(...);
xml_parse(...);
xml_parser_free(...);
...
```

- Vlastite funkcije slobodno se pridružuju za obradu potrebnih tipova dijelova dokumenta

fox.xml

```
<?xml version="1.0"?>
<sentence>The <animal color="blue">fox</animal> leaped over the
<vegetable color="green">cabbage</vegetable> patch and vanished into
the darkness.</sentence>
```

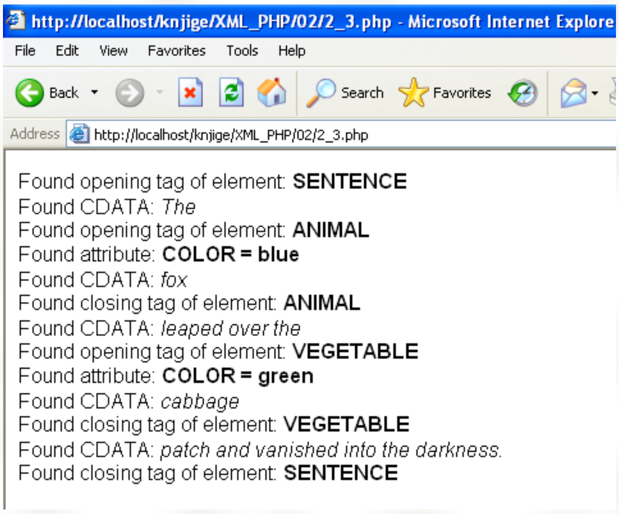
xml_1.php

```
<html>
<head>
<basefont face="Arial">
</head>
<body>
<?php
// run when start tag is found
function startElementHandler($parser, $name, $attributes)
{
      echo "Found opening tag of element: <b>$name</b> <br>";

      // process attributes
      while (list ($key, $value) = each ($attributes))
      {
            echo "Found attribute: <b>$key = $value</b> <br>";
      }
}
// run when end tag is found
function endElementHandler($parser, $name)
{
      echo "Found closing tag of element: <b>$name</b> <br>";
}
// run when cdata is found
function characterDataHandler($parser, $cdata)
{
      echo "Found CDATA: <i>$cdata</i> <br>";
}
```

xml_1.php / 2.

```
// XML data file
$xml_file = "fox.xml";
// initialize parser
$xml_parser = xml_parser_create();
// set callback functions
xml_set_element_handler($xml_parser, "startElementHandler", "endElementHandler");
xml_set_character_data_handler($xml_parser, "characterDataHandler");

// read XML file
if (!($fp = fopen($xml_file, "r")))
{
      die("File I/O error: $xml_file");
}
// parse XML
while ($data = fread($fp, filesize($xml_file)))
{
    // error handler
      if (!xml_parse($xml_parser, $data, feof($fp)))
      {
            die("XML parser error: " . xml_error_string(xml_get_error_code($xml_parser)));
      }
}

// all done, clean up!
xml_parser_free($xml_parser);
?>
</body>
</html>
```

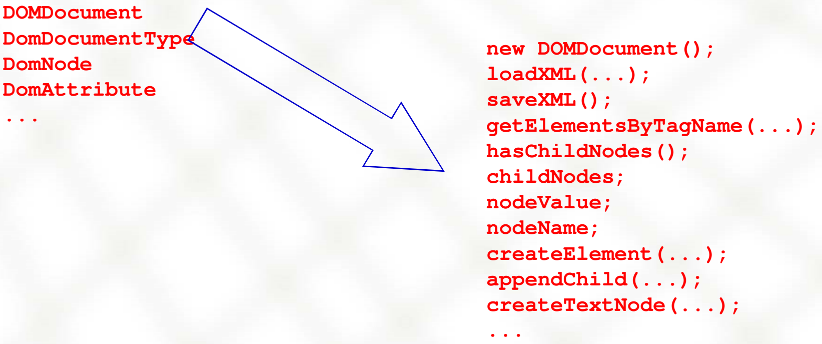
Rezultat


http://localhost/knjige/XML_PHP/02/2_3.php - Microsoft Internet Explore
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/knjige/XML_PHP/02/2_3.php

```
Found opening tag of element: SENTENCE
Found CDATA: The
Found opening tag of element: ANIMAL
Found attribute: COLOR = blue
Found CDATA: fox
Found closing tag of element: ANIMAL
Found CDATA: leaped over the
Found opening tag of element: VEGETABLE
Found attribute: COLOR = green
Found CDATA: cabbage
Found closing tag of element: VEGETABLE
Found CDATA: patch and vanished into the darkness.
Found closing tag of element: SENTENCE
```

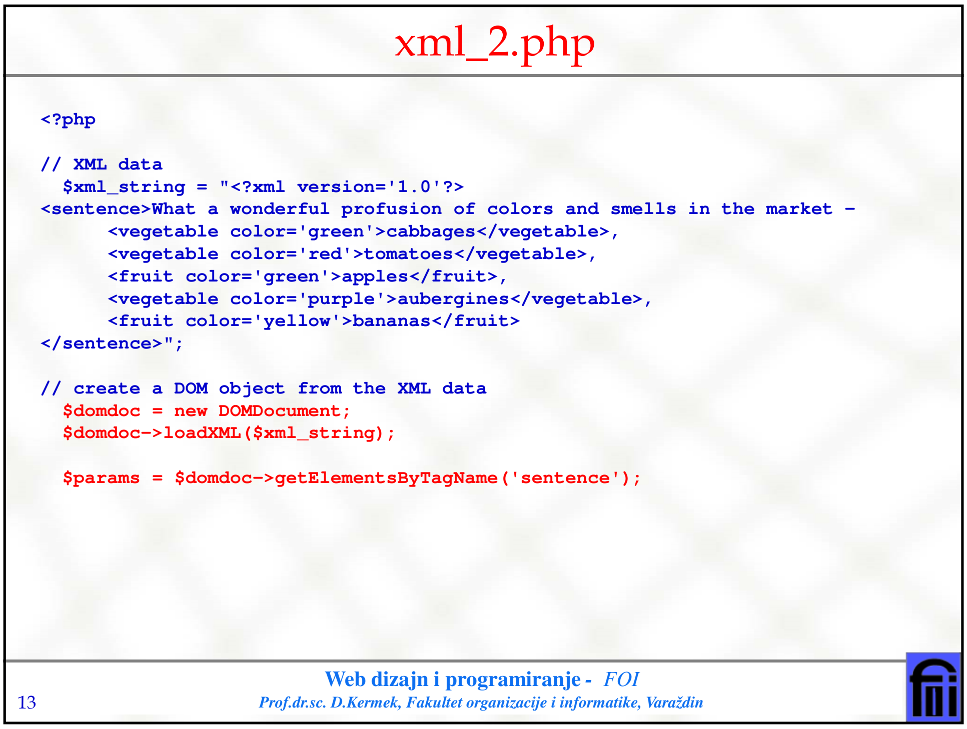
DOM – Document Object Model

- **Obrada dokumenta temelji se na kreiranju hijerarhijskog stabla koje predstavlja XML dokument**
- **Značajna promjena dolazi s verzijom PHP 5**
- **Primijenjuje se objektna orijentacija tako da postoje klase sa svojim atributima i metodama za rad s XML dokumentom:**



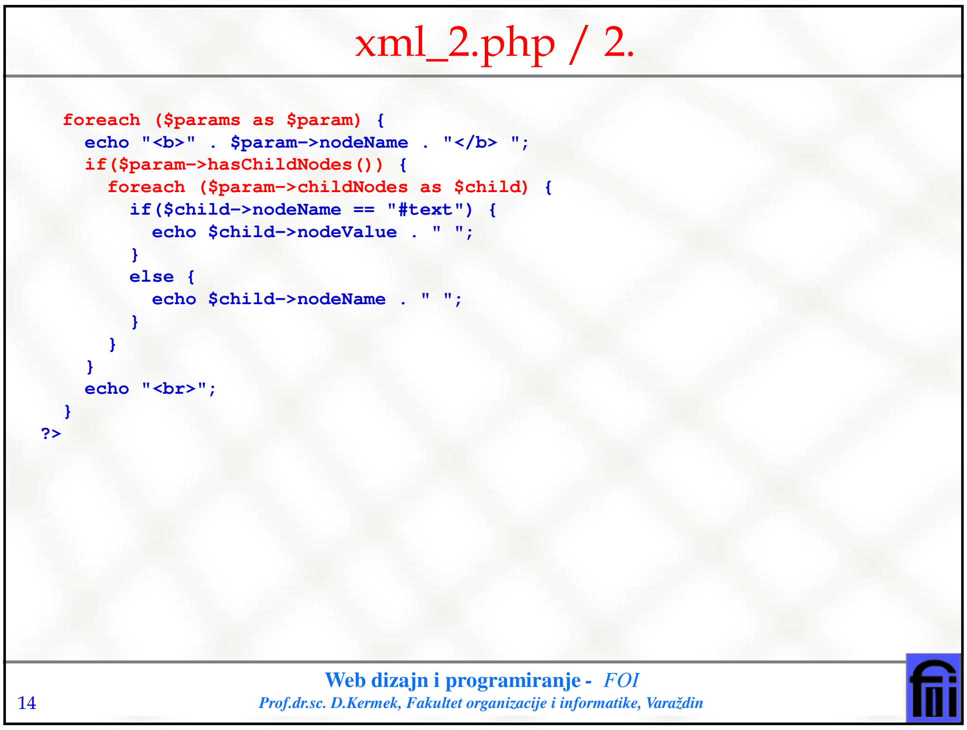
```
DOMDocument
DomDocumentType
DomNode
DomAttribute
...
```

```
new DOMDocument();
loadXML(...);
saveXML();
getElementsByTagName(...);
hasChildNodes();
childNodes;
nodeValue;
nodeName;
createElement(...);
appendChild(...);
createTextNode(...);
...
```



xml_2.php

```
<?php

// XML data
  $xml_string = "<?xml version='1.0'?>
<sentence>What a wonderful profusion of colors and smells in the market -
      <vegetable color='green'>cabbages</vegetable>,
      <vegetable color='red'>tomatoes</vegetable>,
      <fruit color='green'>apples</fruit>,
      <vegetable color='purple'>aubergines</vegetable>,
      <fruit color='yellow'>bananas</fruit>
</sentence>";

// create a DOM object from the XML data
  $domdoc = new DOMDocument;
  $domdoc->loadXML($xml_string);

  $params = $domdoc->getElementsByTagName('sentence');
```



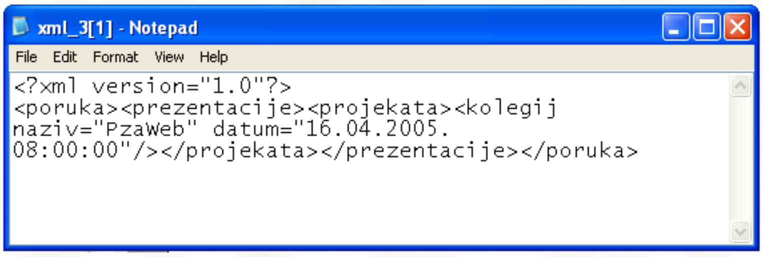
xml_2.php / 2.

```
  foreach ($params as $param) {
    echo "<b>" . $param->nodeName . "</b> ";
    if($param->hasChildNodes()) {
      foreach ($param->childNodes as $child) {
        if($child->nodeName == "#text") {
          echo $child->nodeValue . " ";
        }
        else {
          echo $child->nodeName . " ";
        }
      }
    }
    echo "<br>";
  }
?>
```

xml_3.php

```php
<?php
  $doc = new DOMDocument("1.0");
  $root = $doc->createElement("poruka");
  $root = $doc->appendChild($root);
  $head = $doc->createElement("prezentacije");
  $head = $root->appendChild($head);
  $titl = $doc->createElement("projekata");
  $titl = $head->appendChild($titl);
  $body = $doc->createElement("kolegij");
  $body = $titl->appendChild($body);
  $name = $doc->createAttribute("naziv");
  $name = $body->appendChild($name);
  $text = $doc->createTextNode("WebDiP");
  $attr = $name->appendChild($text);
  $date = $doc->createAttribute("datum");
  $date = $body->appendChild($date);
  $text = $doc->createTextNode(date("y.m.d h:i:s"));
  $text = $date->appendChild($text);
  header("Content-type: text/xml");
  echo $doc->saveXML();
?>
```

Rezultat



```
<?xml version="1.0"?>
<poruka><prezentacije><projekata><kolegij
naziv="PzaWeb" datum="16.04.2005.
08:00:00"/></projekata></prezentacije></poruka>
```

xml_4.php

```
<?php
  $url = "http://localhost/PzaWeb/PHP_Primjeri/xml/xml_3.php";

  if(! ($fp = fopen($url,'r'))) {
    echo "Problem: nije moguće otvoriti url: " . $url;
    exit;
  }

// XML data
  $xml_string = fread($fp, 10000);
  fclose($fp);

// create a DOM object from the XML data
  $domdoc = new DOMDocument;
  $domdoc->loadXML($xml_string);
  header("Content-type: text/xml");
  echo $domdoc->saveXML();
?>
```

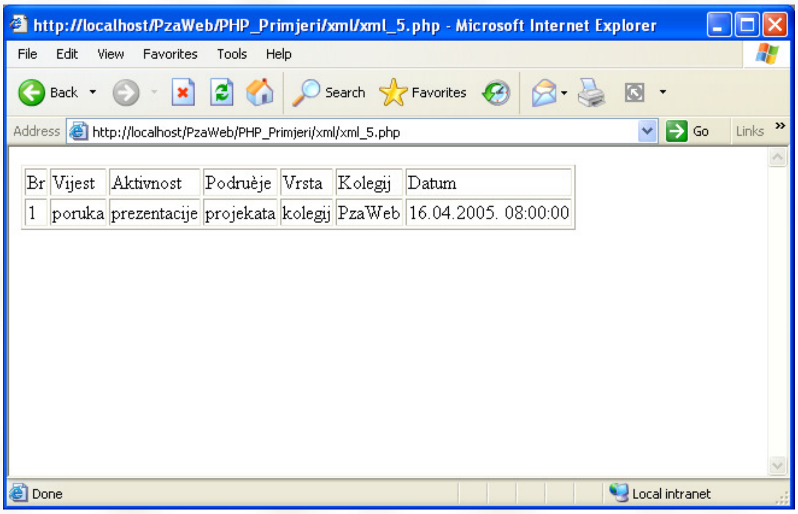
xml_5.php

```
<?php
  $url = "http://localhost/PzaWeb/PHP_Primjeri/xml/xml_4.php";

  if(! ($fp = fopen($url,'r'))) {
    echo "Problem: nije moguće otvoriti url: " . $url;
    exit;
  }

// XML data
  $xml_string = fread($fp, 10000);
  fclose($fp);

// create a DOM object from the XML data
  $domdoc = new DOMDocument;
  $domdoc->loadXML($xml_string);

  $params = $domdoc->getElementsByTagName('poruka');
?>
<table border="1">
<tr><td>Br</td><td>Vijest</td><td>Aktivnost</td><td>Područje</td>
<td>Vrsta</td><td>Kolegij</td><td>Datum</td></tr>
```

xml_5.php / 2.

```
<?
  $i = 0;
  foreach ($params as $param) {
    $i++;
    echo "<tr><td>$i</td><td>$param->nodeName</td>";
    foreach ($param->childNodes as $child) {
      echo "<td>$child->nodeName</td>";
      foreach ($child->childNodes as $gchild) {
        echo "<td>$gchild->nodeName</td>";
        foreach ($gchild->childNodes as $ggchild) {
          echo "<td>$ggchild->nodeName</td>";
          $attributes = $ggchild->attributes;
          foreach ($attributes as $index => $domobj) {
            echo "<td>$domobj->value</td>";
          }
        }
      }
    }
    echo "</td></tr>";
  }
?>
</table>
```

Rezultat


http://localhost/PzaWeb/PHP_Primjeri/xml/xml_5.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://localhost/PzaWeb/PHP_Primjeri/xml/xml_5.php
Go
Links
Done
Local intranet

Br	Vijest	Aktivnost	Podruèje	Vrsta	Kolegij	Datum
1	poruka	prezentacije	projekata	kolegij	PzaWeb	16.04.2005. 08:00:00

XSL, XSLT

- XSL – eXtensive Stylesheet Language - vrlo snažan jezik stilova koji je posebno dizajniran za XML dokumente
- XSL stilske upute su pravilno formirani XML dokumenti
- Postoje 2 različite vrste XML aplikacija:
    - XSL Transformacije (XSLT) – predstavlja opis transformacije ulaznog XML dokumenta iz jednog formata u izlazni XML dokument drugog formata
    - XSL Formatting Objects (XSL-FO) – predstavlja opis kako će stranice izgledati kada budu prezentirane čitatelju
- postoji cijeli XSL jezik (varijable, petlje, brojači, uvjeti,...)

elementi.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="application/xml" href="15-2.xsl"?>
<PERIODIC_TABLE>

  <ATOM STATE="GAS">
    <NAME>Hydrogen</NAME>
    <SYMBOL>H</SYMBOL>
    <ATOMIC_NUMBER>1</ATOMIC_NUMBER>
    <ATOMIC_WEIGHT>1.00794</ATOMIC_WEIGHT>
    <BOILING_POINT UNITS="Kelvin">20.28</BOILING_POINT>
    <MELTING_POINT UNITS="Kelvin">13.81</MELTING_POINT>
    <DENSITY UNITS="grams/cubic centimeter">
      <!-- At 300K, 1 atm -->
      0.0000899
    </DENSITY>
  </ATOM>
  ...
</PERIODIC_TABLE>
```

elementi_1.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="PERIODIC_TABLE">
      <HTML>
        <HEAD><TITLE>Atomic Number vs. Atomic Weight</TITLE></HEAD>
        <BODY>
          <TABLE border="1">
            <xsl:apply-templates>
              <xsl:sort select="ATOMIC_NUMBER"/>
             </xsl:apply-templates>
           </TABLE>
        </BODY>
      </HTML>
    </xsl:template>

    <xsl:template match="ATOM">
     <tr>
        <td><xsl:apply-templates select="NAME"/></td>
        <td><xsl:apply-templates select="ATOMIC_NUMBER"/></td>
        <td><xsl:apply-templates select="ATOMIC_WEIGHT"/></td>
     </tr>
    </xsl:template>
</xsl:stylesheet>
```

elementi_1.php

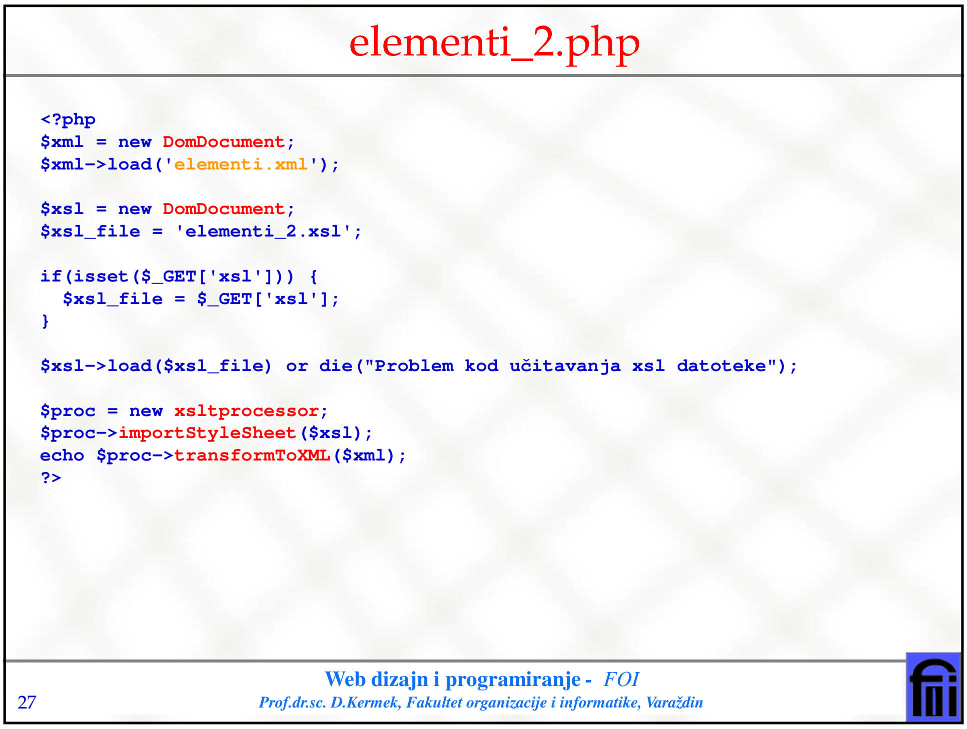
```
<?php
$xml = new DomDocument;
$xml->load('elementi.xml');

$xsl = new DomDocument;
$xsl->load('elementi_1.xsl');

$proc = new xsltprocessor;
$proc->importStyleSheet($xsl);
echo $proc->transformToXML($xml);
?>
```

elementi_2.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="PERIODIC_TABLE">
      <HTML>
        <HEAD><TITLE>Atomic Number vs. Atomic Weight</TITLE></HEAD>
        <BODY>
         <h1>Atomic Number vs. Atomic Weight</h1>
         Stilska datoteka: <a href="elementi_2.php?xsl=elementi_2_1.xsl">
                  elementi_2_1.xsl</a><br/><br/>
           <TABLE border="1">
             <xsl:apply-templates>
               <xsl:sort data-type="number" select="ATOMIC_NUMBER"/>
             </xsl:apply-templates>
            </TABLE>
        </BODY>
      </HTML>
    </xsl:template>
```

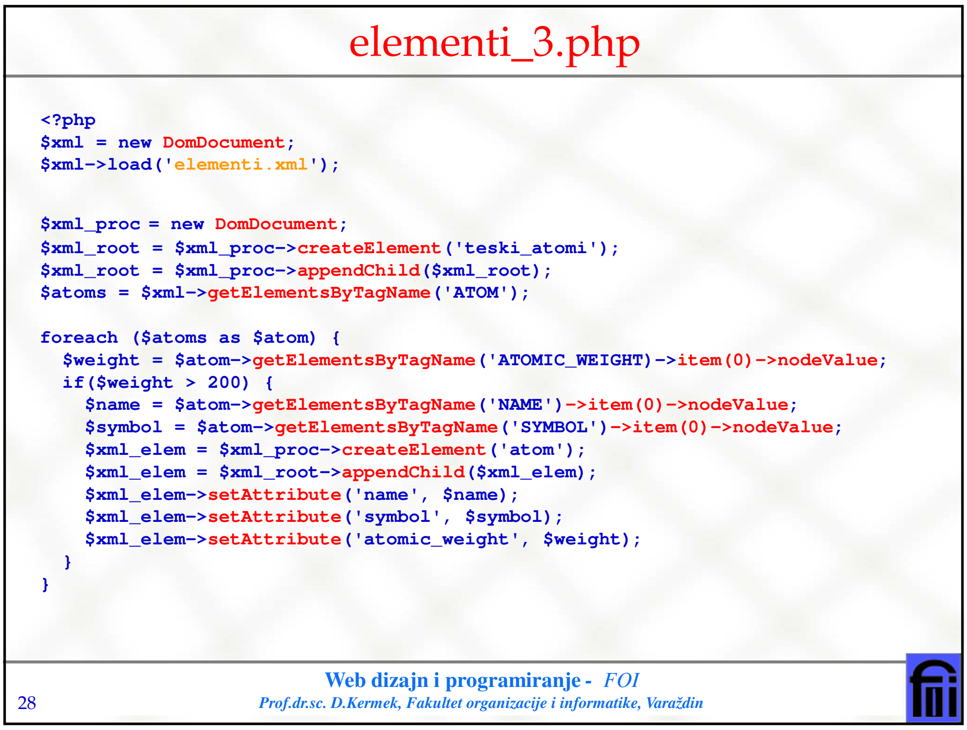
elementi_2.xsl / 2.

```
   <xsl:template match="ATOM">
      <xsl:variable name="position">
         <xsl:value-of select="position()"/>
      </xsl:variable>
      <xsl:variable name="style">
      <xsl:choose>
        <xsl:when test="$position mod 2 = 0">
          #e0e0e0
        </xsl:when>
        <xsl:otherwise>
          #ffffff
        </xsl:otherwise>
      </xsl:choose>
      </xsl:variable>

      <tr style="BACKGROUND-COLOR: {$style};">
         <td><xsl:apply-templates select="ATOMIC_NUMBER"/></td>
         <td><xsl:apply-templates select="NAME"/></td>
         <td><xsl:apply-templates select="ATOMIC_WEIGHT"/></td>
     </tr>
    </xsl:template>
</xsl:stylesheet>
```



elementi_2.php

```php
<?php
$xml = new DomDocument;
$xml->load('elementi.xml');

$xsl = new DomDocument;
$xsl_file = 'elementi_2.xsl';

if(isset($_GET['xsl'])) {
  $xsl_file = $_GET['xsl'];
}

$xsl->load($xsl_file) or die("Problem kod učitavanja xsl datoteke");

$proc = new xsltprocessor;
$proc->importStyleSheet($xsl);
echo $proc->transformToXML($xml);
?>
```



elementi_3.php

```php
<?php
$xml = new DomDocument;
$xml->load('elementi.xml');


$xml_proc = new DomDocument;
$xml_root = $xml_proc->createElement('teski_atomi');
$xml_root = $xml_proc->appendChild($xml_root);
$atoms = $xml->getElementsByTagName('ATOM');

foreach ($atoms as $atom) {
  $weight = $atom->getElementsByTagName('ATOMIC_WEIGHT)->item(0)->nodeValue;
  if($weight > 200) {
    $name = $atom->getElementsByTagName('NAME')->item(0)->nodeValue;
    $symbol = $atom->getElementsByTagName('SYMBOL')->item(0)->nodeValue;
    $xml_elem = $xml_proc->createElement('atom');
    $xml_elem = $xml_root->appendChild($xml_elem);
    $xml_elem->setAttribute('name', $name);
    $xml_elem->setAttribute('symbol', $symbol);
    $xml_elem->setAttribute('atomic_weight', $weight);
  }
}
```

elementi_3.php / 2.

```
$xml_proc->saveXML();
header("Content-type: text/xml");
?>
```

elementi_4.xsl

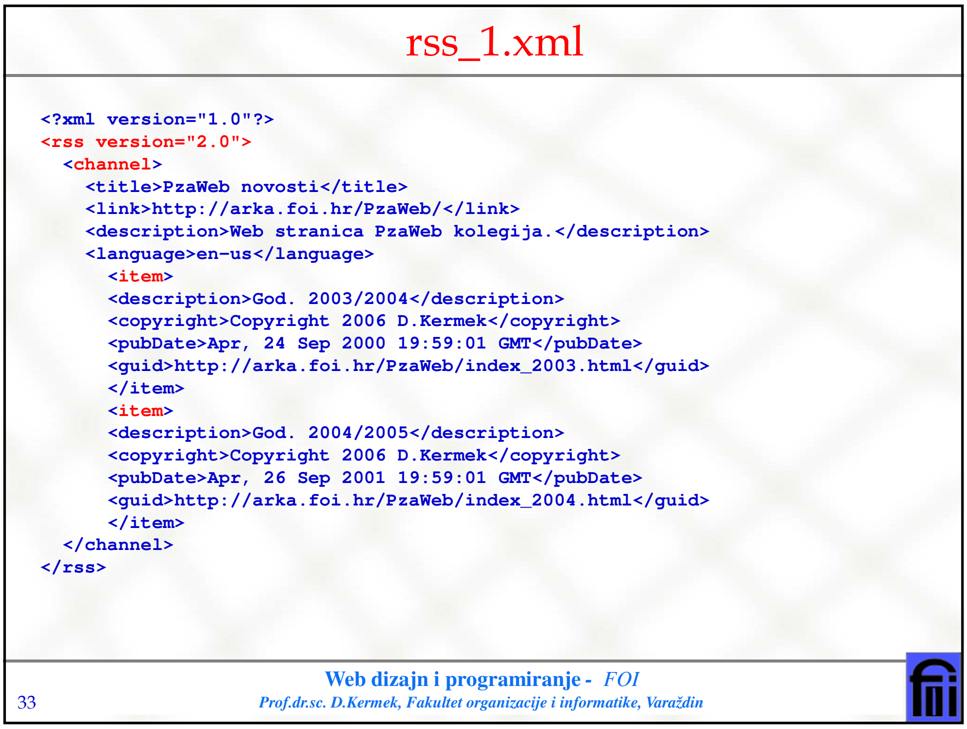
```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="PERIODIC_TABLE">
      <HTML>
        <HEAD><TITLE>Atomic Number vs. Atomic Weight</TITLE></HEAD>
        <BODY>
          <TABLE border="1">
            <xsl:apply-templates>
              <xsl:sort select="NAME"/>
            </xsl:apply-templates>
          </TABLE>
        </BODY>
      </HTML>
    </xsl:template>

    <xsl:template match="ATOM">
     <tr>
        <td><xsl:apply-templates select="NAME"/></td>
        <td><xsl:apply-templates select="ATOMIC_NUMBER"/></td>
        <td><xsl:apply-templates select="ATOMIC_WEIGHT"/></td>
     </tr>
    </xsl:template>
</xsl:stylesheet>
```

elementi_4.php

```
<?php
$url = "http://localhost/PHP_Primjeri/xml/elementi_3.php";

if(! ($fp = fopen($url,'r'))) {
  echo "Problem: nije moguæe otvoriti url: " . $url;
  exit;
}

// XML data
$xml_string = fread($fp, 10000);
fclose($fp);
$xml_proc = new DomDocument;
$xml_proc->loadXML($xml_string);

$xsl = new DomDocument;
$xsl->load('elementi_5.xsl') or die('Problem kod otvaranja xsl datoteke');
$proc = new xsltprocessor;
$proc->importStyleSheet($xsl);
echo $proc->transformToXML($xml_proc);
?>
```

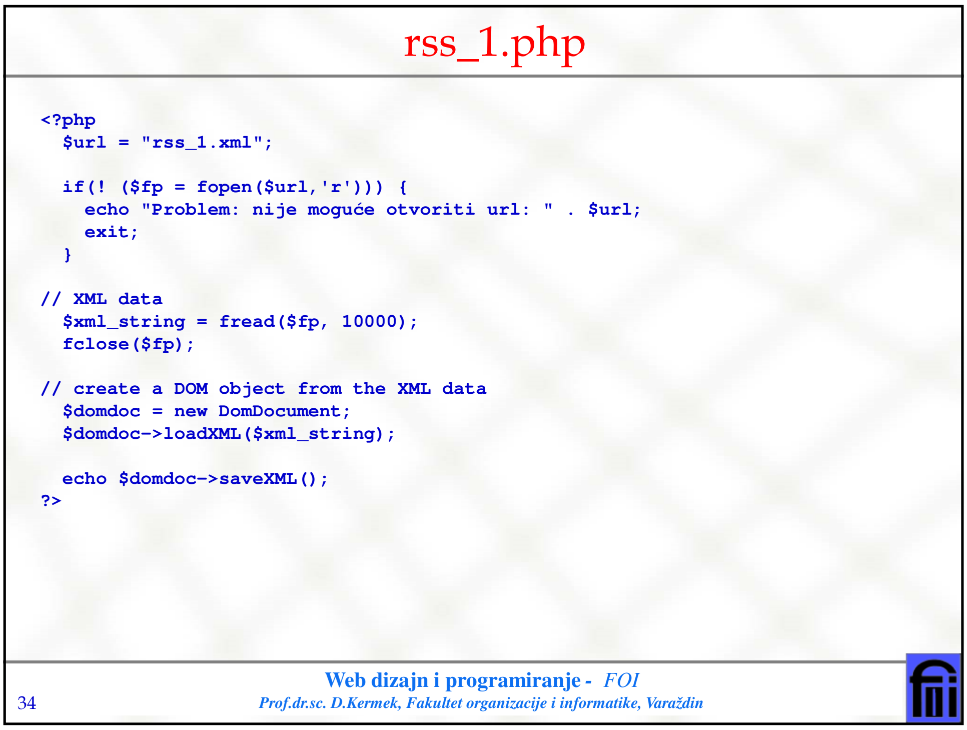
RSS

- **RSS - Really Simple Syndication**
- **Temelji se na XML 1.0**
- **Ideja RSS temelji se na bržem pristupu do podataka koji se često mijenjaju, a oblično se objavljuju putem Web aplikacija (CMS i sl)**
- **Posebno kada se radi o permanentnom objavljivanju ažurnih podataka (vijesti, burzovni podaci i sl).**
- **Koriste se posebni alati za čitanje RSS-a (Sage kod Forefoxa)**



rss_1.xml

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>PzaWeb novosti</title>
    <link>http://arka.foi.hr/PzaWeb/</link>
    <description>Web stranica PzaWeb kolegija.</description>
    <language>en-us</language>
      <item>
      <description>God. 2003/2004</description>
      <copyright>Copyright 2006 D.Kermek</copyright>
      <pubDate>Apr, 24 Sep 2000 19:59:01 GMT</pubDate>
      <guid>http://arka.foi.hr/PzaWeb/index_2003.html</guid>
      </item>
      <item>
      <description>God. 2004/2005</description>
      <copyright>Copyright 2006 D.Kermek</copyright>
      <pubDate>Apr, 26 Sep 2001 19:59:01 GMT</pubDate>
      <guid>http://arka.foi.hr/PzaWeb/index_2004.html</guid>
      </item>
  </channel>
</rss>
```



rss_1.php

```
<?php
  $url = "rss_1.xml";

  if(! ($fp = fopen($url,'r'))) {
    echo "Problem: nije moguće otvoriti url: " . $url;
    exit;
  }

// XML data
  $xml_string = fread($fp, 10000);
  fclose($fp);

// create a DOM object from the XML data
  $domdoc = new DomDocument;
  $domdoc->loadXML($xml_string);

  echo $domdoc->saveXML();
?>
```

Korištena i dodatna literatura

- http://hr2.php.net/domxml
- http://www.php.net/dom
- Vaswani, V. *XML and PHP*, New Riders, 2002
- Harold, E.R. *XML 1.1 Bible 3rd edition*, Wiley Pubishing, Inc. 2004.
- http://www.cafeconleche.org/books/bible3/
- http://www.ibiblio.org/xml/
- http://www.webreference.com/authoring/languages/xml/rss/intro/
- http://www-128.ibm.com/developerworks/xml/library/x-rss20/
- http://pear.php.net/