DOM via PHP

Plan

PHP

DOM

PHP : Hypertext Preprocessor

Langage de script pour création de pages Web dynamiques

Un ficher PHP est un ficher HTML avec du code PHP

```
<html>
 <body>
<?php
echo "Hello World";
?>
</body>
</html>
```

Serveur Web

```
<html>
<body>
Hello World
</body>
</html>
```

PHP 5.0 orienté objet, vaste documentation sur php.net

La suite du cours fera une introduction via exemples des éléments de base.

PHP, éléments de base

```
$i = 1;
if (1 == $i) {
echo "i est 1 <br>"; } else {
echo "i n'est pas 1<br>";
};

switch ($i)
{ case 1:
  case 2:
  case 3: echo "i est soit  1 soit  2 soit  3 <br>";
          break;
  case 4: echo "i est  4 ";
          break;
  default: echo "0 > i > 4";
};

while ($i < 4)
{ echo "i est $i<br>";
 $i += 1;
};
```

```
do
{ echo "i is $i";
$i--;
}
while ($i > 0);
echo (0==$i ? "i est a nouveu 0" : "i n'est pas 0");
```

Sortie

```
i est 1<br>
i est soit  1 soit  2 soit  3<br>
i est 1<br>
i est 2<br>
i est 3<br>
i est 4<br>
i est 3<br>
i est 2<br>
i est 1<br>
i est a nouveu 0
```

PHP, formulaires

```
<form action="action.php" method="post">
 <p>Votre nom : <input type="text" name="nom" /></p>
 <p>Votre âge : <input type="text" name="age" /></p>
 <p><input type="submit" value="OK"></p>
</form>
```

Code dans action.php

```
Bonjour,
<?
php echo htmlspecialchars($_POST['nom']);
 ?>.
Tu as
<?php echo (int)$_POST['age']; ?>
 ans.
```

Les parameters peuvent etre passés en modalité GET aussi (attention, les valeurs sont concatenées à l'URL de l'action et sont transmises en claire)

DOM

Document Object Model

Un ensemble de librairies permettant le chargement d'un document XML en mémoire primaire sous forme d'arbre

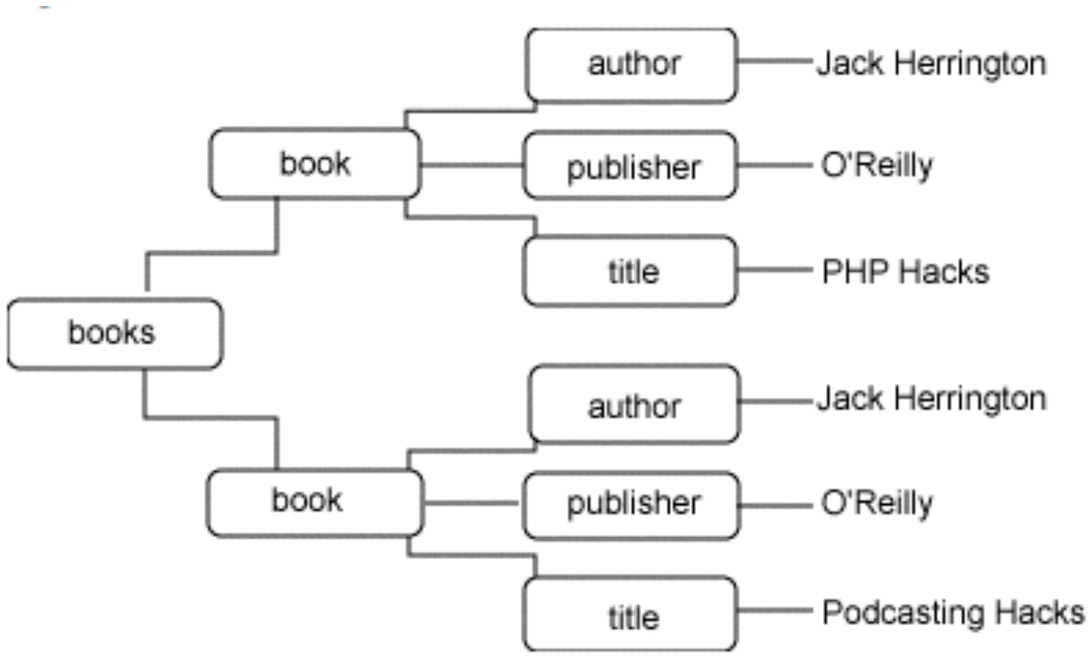
DOM permet de naviguer dans l'arbre, et d'effectuer des mises à jour

DOM permet d'écrire sur disque le contenu de l'arbre XML dans la mémoire primaire.

DOM permet d'évaluer une requête XPath ou une transformation XSLT sur un document XML

Exemple, en image

```
<books>
<book>
<author>Jack Herrington</author>
<title>PHP Hacks</title>
<publisher>O'Reilly</publisher>
</book>
<book>
<author>Jack Herrington</author>
<title>Podcasting Hacks</title>
<publisher>O'Reilly</publisher>
</book>
</books>
```



Lecture de document, et sélection d'éléments

```
<?php

 $doc = new DOMDocument();
 $doc->load( 'books.xml' );

 $books = $doc->getElementsByTagName( "book" );

 foreach( $books as $book )
        {
         $authors = $book->getElementsByTagName( "author" );
         $author = $authors->item(0)->nodeValue;

         $publishers = $book->getElementsByTagName( "publisher" );
         $publisher = $publishers->item(0)->nodeValue;

         $titles = $book->getElementsByTagName( "title" );
         $title = $titles->item(0)->nodeValue;

          echo "$title - $author - $publisher\n";
        }
?>
```

Création de document et écriture

```
<?php
$books = array();
$books [] = array(
'title' => 'PHP Hacks',
'author' => 'Jack Herrington',
'publisher' => "O'Reilly"
);

$books [] = array(
  'title' => 'Podcasting Hacks',
  'author' => 'Jack Herrington',
  'publisher' => "O'Reilly"
  );

  $doc = new DOMDocument();
  $doc->formatOutput = true;

  $r = $doc->createElement( "books" );
  $doc->appendChild( $r );
...
```

Création de document et écriture

```
...
foreach( $books as $book )
  {
  $b = $doc->createElement( "book" );

  $author = $doc->createElement( "author" );
  $author->appendChild(
  $doc->createTextNode( $book['author'] )
  );
$b->appendChild( $author );

$title = $doc->createElement( "title" );
$title->appendChild(
$doc->createTextNode( $book['title'] )
);
$b->appendChild( $title );

....
```

Création de document et écriture

```
....

$publisher = $doc->createElement( "publisher" );
$publisher->appendChild(
$doc->createTextNode( $book['publisher'] )
);
$b->appendChild( $publisher );

$r->appendChild( $b );
}

$doc->save(bib.xml);
?>
```

Evaluation de requêtes XPath

Exemple : compter le nombre de livres écrits par un auteur

Version sans XPath

```
<?php

$doc = new DOMDocument();

$doc->load('book.xml');

$author =‘Vianu’;

$total = 0;

$elements = $doc->domDocument->getElementsByTagName("author");

foreach ($elements as $element) {

        if ($element->nodeValue == $author) {

            $total++;  }}

.....
```

Evaluation de requetes XPath

Exemple : compter le nombre de livres écrits par un auteur

Version 1 avec XPath

```
<?php

$author =‘Vianu’;

$doc = new DOMDocument();

$doc->load('book.xml');

$xpath = new DOMXPath($doc);

$query = "//book[author=’$author’]";

$result = $xpath->query($query);

....
```

Evaluation de requêtes XPath

Exemple : compter le nombre de livres écrits par un auteur

Version 2 avec XPath

```
<?

$doc = new DOMDocument();

$doc->load('book.xml');

$xpath = new DOMXPath($doc);

$query = "count(//book[author='$author'])";

return $xpath->evaluate($query);

.....

?>
```

XSLT via PHP

```
$xp = new XSLTProcessor();

$xsl = new DOMDocument();

$xsl->load('unetransformationXMLtoHTML.xsl');

$xp->importStylesheet($xsl);

$xmldoc = new DOMDocument;

$xmldoc->load('document.xml');

$html = $xp->transformToXML($xmldoc);

echo $html;
```

Exercice

Ecrire une page Web pour la recherche des livres par auteur, et pour l'insertion de livres, dans le fichier books.xml

SAX

Parsing XML with SAX

```
<?php
  // ...
  $sax = xml_parser_create();
  xml_parser_set_option($sax, XML_OPTION_CASE_
    FOLDING, false);
  xml_parser_set_option($sax, XML_OPTION_SKIP_WHITE,
    true);
  xml_set_element_handler($sax, 'sax_start',
    'sax_end');
  xml_set_character_data_handler($sax, 'sax_cdata');
  xml_parse($sax, file_get_contents('quotes.xml'),
    true);
  xml_parser_free($sax);
?>
```

Credits : http://www.brainbell.com/tutorials/php/Parsing_XML_With_SAX.htm

SAX

You create a SAX (Simple API for XML) parser using `xml_parser_create()`. This parser can look at an XML file and react upon various events. The following three events are the most important ones:

- Beginning of an element
- End of an element
- CDATA blocks

SAX

You can then define handler functions for these elements and use them to transform the XML into something else, for instance Hypertext Markup Language (HTML). Listing on previous page shows this and outputs the contents of the XML file as a bulleted HTML list, as shown in figure. The function `xml_set_element_handler()` sets the handlers for the beginning and end of an element, whereas `xml_set_character_data_handler()` sets the handler for CDATA blocks. With `xml_parser_set_option()`, you can configure the handler, for instance to ignore whitespace and to handle tag names as case sensitive (then tag names are not converted into uppercase letters automatically). The following code contains the code for the handler functions:

SAX

```
function sax_start($sax, $tag, $attr) {
  if ($tag == 'quotes') {
    echo '<ul>';
  } elseif ($tag == 'quote') {
    echo '<li>' . htmlspecialchars($attr['year'])
      ': ';
  } elseif ($tag == 'coding') {
    echo '"';
  } elseif ($tag == 'author') {
    echo ' (';
  }
}
```

SAX

```
function sax_end($sax, $tag) {
  if ($tag == 'quotes') {
    echo '</ul>';
  } elseif ($tag == 'quote') {
    echo '</li>';
  } elseif ($tag == 'coding') {
    echo '"';
  } elseif ($tag == 'author') {
    echo ') ';
  }
}
```

SAX

```
function sax_cdata($sax, $data) {

  echo htmlspecialchars($data);

}
```

SAX


http://localhost/sax.php
File Edit View Web Go Bookmarks Tabs Help
Back Forward Stop Refresh Home Fullscreen
100
http://localhost/sax.php
Go
1991: "Hasta la vista, baby!" (Arnold Schwarzenegger)
2000: "This race is as tight as the rusted lug nuts on a '55 Ford." (Dan Rather)
2004: "This race is hotter than a Times Square Rolex." (Dan Rather)