SimpleXML Introduction

Voir http://www.php.net/manual/fr/book.simplexml.php

L'extension SimpleXML fournit des outils très simples et faciles à utiliser pour convertir du XML en un objet qui peut être manipulé avec ses propriétés et les itérateurs de tableaux.

Une petite application

Fichier XML

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

  <librairie>
   <rayon id="fiction">
    <livre>
     <titre>Des souris et des hommes</titre>
     <auteur>John Steinbeck</auteur>
    </livre>
    <livre>
     <titre>Harry Potter et la pierre philosophale</titre>
     <auteur>J.K. Rowling</auteur>
    </livre>
   </rayon>
  </librairie>
```

La simplicité de SimpleXML apparaît plus clairement lorsqu'on essaye d'extraire une chaîne ou un nombre d'un document XML basique.

Exemple lecture

```php
<?php
      // Le fichier test.xml contient un document XML avec un élément racine
// et au moins un élément /[racine]/ librairie

if (file_exists(librairie.xml')) {
  $xml = simplexml_load_file('librairie.xml');

   print_r($xml);
} else {
   exit('Echec lors de l\'ouverture du fichier test.xml.');
}
?>
```

Lecture 2

```php
<?php
 $librairie = simplexml_load_file('librairie.xml');
 // par défaut, on est déja sur le premier
  //Ce qui explique l'usage de ->livre alors qu'il y en a 2
  $premier="Titre :".$librairie->rayon->livre->titre;
 // compter les livres phpversion() >= '5.3.0'
  // on peut utiliser :$nombre=$librairie->rayon->livre->count();
  // Il y a une variante ;il faut choisir !
  //Et toujours phpversion() >= '5.3.0'
  $nombre=count($librairie->rayon->livre);
  echo "nombre de livres: $nombre";
```

```
  echo "<br> premier livre -$premier <br>";
  echo "Attribut du rayon: ".$librairie->rayon['id']."<br>";
  //acceder à un livre
  echo "Auteur du 2me livre :".$librairie->rayon->livre[1]->auteur;
  echo "<hr> <h1> contenu de notre object simplexml</h1><pre>";
  print_r($librairie);
  echo "</pre>" ;
?>
```

*simplexml_load_file(*XML *filename)* charge un fichier XML ;Convertit le document XML *filename* en un objet de type SimpleXMLElement.

print_r() imprime une structure des données

- **Accès au premier élément** : utiliser les propriétés `$librairie->rayon->book`
- **Compte tous les éléments** d'un tableau ou d'un objet `count()`
  `count($librairie->rayon->livre)`
- **Accès au n^ème^ élément** : utiliser les propriétés
  `$librairie->rayon->livre[0]->title;`
- Lorsque **plusieurs instances d'un élément** existent en tant que fils d'un élément père unique, les techniques normales d'itération peuvent être appliquées.
  `Foreach (){}`
  `foreach ($rayon->livre as $livre) { ... }`
- `$librairie->rayon['id']` renvoie la valeur de l'attribut **id** de l'élément

autre exemple

```
<?php
 $librairie = simplexml_load_file('librairie.xml');
  foreach ($librairie->rayon as $rayon) {
      echo "<h1>Categorie ". $rayon['id']."</h1>";
      foreach ($rayon->livre as $livre) {
          echo "Titre: ". $livre->titre;
          echo " Auteur:".$livre->auteur."<br>";
      }
  }
?>
```

SimpleXMLElement->addChild()

SimpleXMLElement **addChild** (string `$name` [, string `$value` [, string `$namespace`]])

Liste de paramètres

name

Le nom de l'élément enfant à ajouter.

value

Si spécifié, la valeur de l'élément enfant.

namespace

Si spécifié, l'espace de nom auquel l'élément enfant appartient.

SimpleXMLElement->addAttribute()

Liste de paramètres

name

Le nom de l'attribut à ajouter.

value

La valeur de l'attribut.

namespace

Si spécifié, l'espace de nom auquel l'attribut appartient.

SimpleXMLElement->children()

http://www.php.net/manual/fr/simplexmlelement.children.php

```
<!DOCTYPE HTML >
<html>
<head> <meta  charset="UTF-8">
      <title>genealogie</title>
</head>

<body>

<?php
$xml = new SimpleXMLElement(
      '<person>
            <child role="fils">
                  <child role="fille"/>
            </child>
            <child role="fille">
                  <child role="fils">
                        <child role="fils"/>
                  </child>
            </child>
      </person>');

      foreach ($xml->children() as $second_gen) {
            echo ' La personne a engendré un(e) ' . $second_gen['role'];

            foreach ($second_gen->children() as $third_gen) {
                  echo ' qui a engendré un(e)' . $third_gen['role'] . ';';

                  foreach ($third_gen->children() as $fourth_gen) {
                        echo ' et ce (cette) ' . $third_gen['role'] .
                              ' a engendré ' . $fourth_gen['role'];
                  }
            }
      }
?>
</body>
</html>
```

Ajout s'un noeud et de deux attributs

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
	<title>controls</title>
</head>
<body>

<?php
$xml_string = '
<?xml version=\'1.0\' standalone=\'yes\'?>
<design-patterns>
	<front-controller>x</front-controller>
	<application-controller>x</application-controller>
	<page-controller>x</page-controller>
</design-patterns>';

echo nl2br(htmlentities($xml_string)); echo '<br /><hr>';  //echo above xml

$xml_element = new SimpleXMLElement($xml_string);
$xml_element->addAttribute('type','generic');
$new_pattern = $xml_element->addChild('identity-map','y');
$new_pattern->addAttribute('type','generic');

echo nl2br(htmlentities($xml_element->asXML()));
echo "<hr>";

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

Ce qui donne

<?xml version='1.0' standalone='yes'?>
<design-patterns>
<front-controller>x</front-controller>
<application-controller>x</application-controller>
<page-controller>x</page-controller>
</design-patterns>

<?xml version="1.0" standalone="yes"?>
<design-patterns type="generic">
<front-controller>x</front-controller>
<application-controller>x</application-controller>
<page-controller>x</page-controller>
<identity-map type="generic">y</identity-map></design-patterns>

GetName

```
$Noeud->getName()
```

Récupère les noms des éléments XML

```
$librairie = simplexml_load_file('librairie.xml');
$noeud=$librairie->rayon;
foreach ($noeud->children() as $enfant)
{
    echo $enfant->getName() . "<br>";
}
```

sortie :

```
livre
livre
```

Exemple de synthèse *trouvé sur wikipédia*

http://fr.wikipedia.org/wiki/SimpleXML

Un document XML simple

```xml
<?xml version='1.0' standalone='yes'?>
<films>
 <film>
  <titre>Le nom de la rose</titre>
  <duree>127 min</duree>
 </film>
 <film>
  <titre>Sacré Graal</titre>
  <duree>91 min</duree>
 </film>
 <film>
  <titre>Le livre de la jungle</titre>
  <duree>75 min</duree>
 </film>
</films>
```

En php:

```
<?php

$simpleXml = new SimpleXMLElement($chaineXml);

// écrit "Le nom de la rose"
echo $simpleXml->film[0]->titre;

// supprime le 3eme film (la numérotation des éléments commence à 0, le
troisième élément est donc numéro 2)
unset($simpleXml->film[2]);

// ajoute un film nommé "La liste de Schindler" (197 min)
$nouveauFilm   = $simpleXml->addChild('film');
$nouveauTitre  = $nouveauFilm->addChild('titre', 'La liste de Schindler');
$nouvelleDuree = $nouveauFilm->addChild('duree', '197 min');

// affiche le contenu de l'object simplexml
print_r($simpleXml);

?>
```