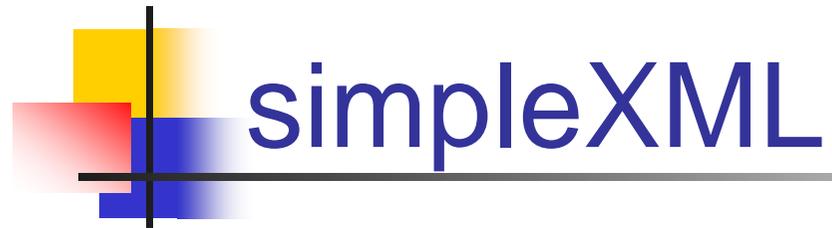


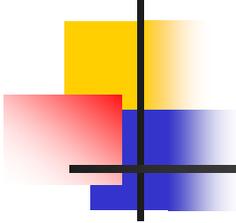


simpleXML



## ■ Introduction

- Dans les versions précédentes de PHP, il n'y avait aucun outil permettant de manipuler simplement des fichiers XML
- Dans PHP 5, on trouve un mode, appelé, **SimpleXML** qui propose une interface pour ce type d'applications
- Son utilisation est faite pour relire ou modifier facilement des fichiers XML simples
- SimpleXML va vous permettre de lire des valeurs dans un fichier XML le plus simplement du monde



# simpleXML

---

## ■ Import et export d'un document

### – Import du code XML

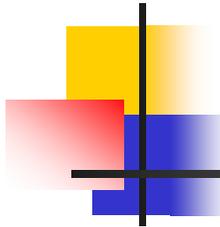
- L'ouverture d'un fichier XML se fait à l'aide de la méthode `simple_load_file()`
- Elle prend en paramètre un chemin de fichier et renvoie un objet de la classe `simpleXMLElement` (racine)
  - ❖ `$racine = simplexml_load_file('monfichier.xml');`
- On peut aussi initialiser l'objet à partir d'une chaîne de caractères représentant le fichier XML : on utilise dans ce cas : `simplexml_load_string()`

```
<?php
```

```
$xml = "<document><titre>PHP5 avance</titre>  
</document>";
```

```
$racine = simplexml_load_string($xml);
```

```
?>
```

The logo for simpleXML, featuring a stylized 's' composed of overlapping yellow, red, and blue squares, with a black crosshair.

# simpleXML

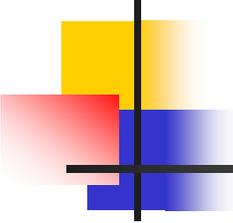
---

- Export et sauvegarde du code XML
  - À tout moment, on peut afficher ou exporter une partie du document XML grâce à `asXML()`
  - Elle renvoie le contenu XML dans une chaîne de caractères (ici, toute l'arborescence car utilisé sur `$racine`)

```
<?php
    $racine = simplexml_load_file('fichier.xml');
    echo $racine->asXml();
?>
```

- Sauvegarde des données XML : `copier_fichier.php`

```
<?php
    $racine = simplexml_load_file('fichier.xml');
    echo $racine->asXml('copie.xml');
?>
```



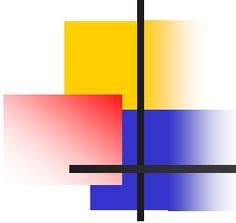
# simpleXML

---

- Manipulation des éléments

- Exemple : copie.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<html lang="fr">
  <head><title>PHP 5</title></head>
  <body>
    <h1>PHP 5</h1>
    <p>La version 5 de
    <acronym>PHP</acronym>vient de sortir</p>
    <p>La dernière version était la 4.3.x</p>
  </body>
</html>
```



# simpleXML

---

- Manipulation des éléments

- Accéder à un nœud par son nom : un seul nœud

- On utilise son nom : `acces_noeud_nom.php`

```
<?php
```

```
    $racine = simplexml_load_file('copie.xml');
```

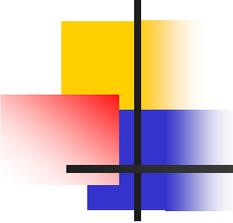
```
    $body = $racine->body;
```

```
    $element_h1 = $body->h1
```

```
    echo $element_h1;
```

```
?>
```

- Affiche : PHP 5



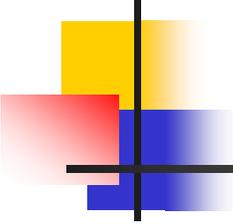
# simpleXML

---

## ■ Manipulation des éléments

- Accéder à plusieurs nœuds : `acces_plus_noeud.php`
  - Si plusieurs nœuds ont le même nom, SimpleXML renvoie un tableau indexé de ces éléments, on peut au choix y accéder avec leur index ou les lire un à un avec `foreach()`

```
<?php
    $racine = simplexml_load_file('fichier.xml');
    // On récupère le premier paragraphe
    $premier_p = $body->p[0];
    // On lit tous les paragraphes
    foreach($body->p as $p){
        echo utf8_decode($p), '<br />';
    }
?>
```



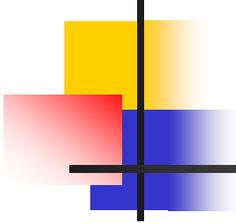
# simpleXML

---

- Manipulation des éléments

- Lire les nœuds fils : children() renvoie une liste : lire\_fils.php

```
<?php
    // Accès au premier fils de l'élément <body>
    $racine = simplexml_load_file('fichier.xml');
    $body = $racine->body;
    $liste = $body->children();
    echo $liste[0];
    // Affiche le titre en <h1>
    // qui était le premier fils de <body>
?>
```



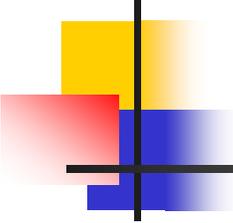
# simpleXML

---

## ■ Manipulation des éléments

- Lire les nœuds fils (suite) : lire\_plus\_fils.php
  - On peut également utiliser la syntaxe foreach() pour itérer à travers tous les éléments fils

```
<?php
// Réaffiche le contenu de <body>
$racine = simplexml_load_file('fichier.xml');
$body = $racine->body;
foreach($body->children() as $nom=>$element){
    echo "La balise $nom contient l'élément \ » »,
    utf8_decode($element), "<br>";
}
?>
```



# simpleXML

---

## ■ Manipulation des éléments

- Ajouter un nœud fils : addChild() : ajouter\_noeud\_fils.php

```
<?php
```

```
// Ajoute un paragraphe à la fin d'une page XHTML
```

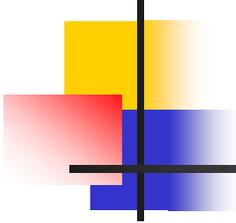
```
$racine = simplexml_load_file('fichier.xml');
```

```
$body = $racine->body;
```

```
$body->addChild("p", "PHP 5 avancé");
```

```
?>
```

- Seul le premier paramètre « p » est obligatoire



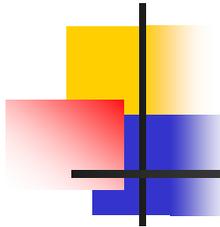
# simpleXML

---

- Manipulation des éléments

- Afficher le contenu textuel d'un nœud : echo et print :  
affiche\_contenu\_noeud.php

```
<?php
// Affichage du titre
$racine = simplexml_load_file('fichier.xml');
$body = $racine->body;
echo $body->h1;
// On peut également forcer avec un cast de ce type
echo htmlentities((string) $body->h1);
?>
```



# simpleXML

---

## ■ Manipulation des attributs

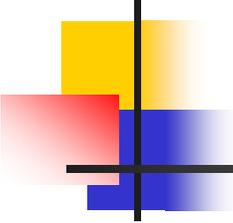
- Accéder à un attribut : `acces_attribut.php`

```
<?php
// langue du document XHTML
$html = simplexml_load_file('fichier.xml');
echo $html['lang'];
?>
```

- Lister tous les attributs : `attributes()` : `lister_attributs.php`

- renvoie un tableau avec, pour chaque ligne, le nom de chaque attribut comme clé, et son contenu comme valeur

```
<?php
// Réécriture de la balise d'ouverture HTML
$html = simplexml_load_file('fichier.xml');
$balise = '<html' ;
foreach($html->attributes() as $nom => $valeur) {
    $balise .= " $nom='$valeur'";
}
$balise .= ">";
echo $balise;
?>
```



# simpleXML

---

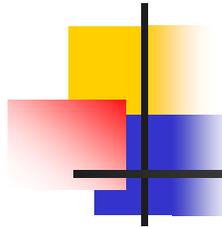
## ■ Manipulation des attributs

- Modifier un attribut : `modifier_attributs.php`
  - On peut créer ou modifier un attribut en changeant directement. Il est possible de supprimer un attribut avec `unset()`

```
<?php
// Passe la déclaration de la langue en anglais
$html = simplexml_load_file('fichier.xml');
$html['lang'] = 'en';
$html->asXml('coipe2.xml');
//effacement de l'attribut
Unset($html['attribut']);
?>
```

- Ajouter un attribut : `addAttribute()` sur l'élément parent

```
<?php
// crée une déclaration de langue si elle n'existe pas
$html = simplexml_load_file('fichier.xml');
$html->body[0]->addAttribute("lang", "e");
?>
```



# simpleXML

---

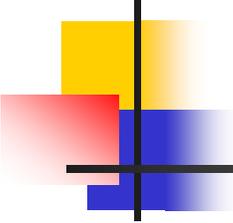
- Fichier RSS (Really Simple Syndication)
  - C'est un format conçu par Netscape afin de présenter les titres des articles en publication de manière standardisée
  - Chacun peut relire ce fichier XML pour connaître les dernières nouvelles et les intégrer à une interface personnelle
  - On peut ainsi être prévenu d'une mise à jour d'un site ou d'un contenu sans avoir à le visiter manuellement
  - On parle de **syndication** de données

## ■ Fichier RSS (Really Simple Syndication)

- On trouve plusieurs versions du format RSS, voici un échantillon de RSS 0.91

```
?xml version="1.0" encoding="iso-8859-1" ?>
<?xml-stylesheet href="/rss/stylerss.xml"
  type="text/xsl" ?>
<!DOCTYPE rss PUBLIC "-//Netscape
  Communications//DTD RSS 0.91//EN"
"http://my.netscape.com/publish/formats/rss-
  0.91.dtd">
<rss version="0.91">
  <channel>
    <title>Titre du canal RSS</title>
    <description>Description globale du fil
      RSS</description>
    <link>URL d'une page décrivant le canal
      RSS</link>
    <image>
      <title>Titre de l'image associée au
        canal</title>
      <url>http://url_de_l_image</url>
      <link>http://lien_associe_à_l_image</link>
    </image>
    <copyright>Auteur du fil RSS</copyright>
    <webMaster>Adresse électronique du
      webmaster</webMaster>
```

```
<item>
  <title>Titre du premier item</title>
  <link>Lien vers la page décrivant
    l'item plus en détail</link>
  <description>Description
    synthétique de l'item>
</item>
<item>
  <title>Titre du second item</title>
  <link>Lien vers la page décrivant
    l'item plus en détail</link>
  <description>Description
    synthétique de l'item>
</item>
...
</channel>
</rss>
```



# simpleXML

---

- Afficher un fichier RSS : lire\_fichier\_rss.php

```
<?php
    $fichier_rss='http://www.generation-
    nt.com/export/rss_soft.xml';
    $racine=simplexml_load_file($fichier_rss);
    foreach($racine->channel->item as $news){
        echo 'Actu: ', utf8_decode((string) $news->title), '<br />';
    }
?>
```

- Exemple réel : lire\_fichier\_rss2.php

```

<?php
function lit_rss($fichier,$objets) {
    // on lit tout le fichier
    if($chaine = @implode("",@file($fichier))) {
        // on découpe la chaîne obtenue en items
        $tmp =
        preg_split("/<V?\".item.\">/",$chaine);
        // pour chaque item
        for($i=1;$i<sizeof($tmp)-1;$i+=2)
            // on lit chaque objet de l'item
            foreach($objets as $objet) {
                // on découpe la chaîne pour obtenir le
                contenu de l'objet
                $tmp2 =
                preg_split("/<V?\".$objet.\">/",$tmp[$i]);
                // on ajoute le contenu de l'objet au
                tableau resultat
                $resultat[$i-1][] = @$tmp2[1];
            }
        // on retourne le tableau resultat
        return $resultat;
    }
}

```

```

$rss =
    lit_rss("http://www.20minutes.fr/rs
s/une.xml",array("title","link","desc
ription","pubDate"));

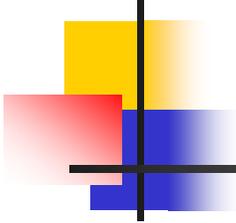
echo '<center><table cellpadding="0"
cellspacing="0"
class="portal_table"
width="100%">';

foreach($rss as $stab) {
    echo '<tr><td
class="header">'.$stab[0].</td></tr
>';
    echo '<tr><td>'.$stab[2].< a
href="\".$stab[1].\">Lire tout
l'article</a></td></tr>';
    echo '<tr><td><div
align=right>posté le
'.date("d/m/Y",strtotime($stab[3])).'
</div></td></tr>';
}

echo '</table></center>';

?>

```



# simpleXML

---

## ■ Recherche Xpath

- Possible avec la méthode xpath()

- Syntaxe

```
$objet_simple_xml->xpath(expression_xpath);
```

- Exemple : chercher tous les paragraphes <p> : `cherche_p.php`

```
<?php
```

```
    $xml = simplexml_load_file('fichier.xml');
```

```
    $xpath = '/html/body/p'; //Recherche des paragraphes
```

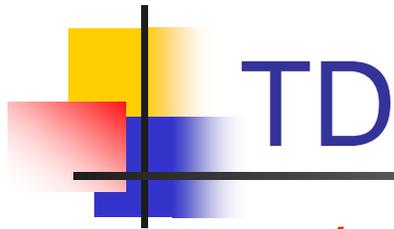
```
    $paragraphes = $xml->xpath($xpath);
```

```
    foreach($paragraphes as $p){
```

```
        echo '<p>', utf8_decode($p), "</p>\n";
```

```
    }
```

```
?>
```



## ■ Énoncé

- Écrire une interface pour interroger par XPath
- Cette interface doit comprendre :
  - Un `inputText` pour écrire l'expression
  - Un `textArea` avec parcours pour aller chercher un document xml
  - Cette zone peut être initialisée avec un document xml
  - La réponse sera donnée dans une autre zone `textArea`

## ■ Pour faire cet exercice

- Regarder le cours sur XPath

## ■ Pour afficher le résultat de manière indentée avec les noms de balises :

- `echo "<pre>".htmlentities($resultat -> asXml()). "</pre>"`