

1 Curso POO PHP SimpleXML

1.1 Sumario

- 1 SimpleXML
 - ◆ 1.1 Cargar un documento XML
 - ◆ 1.2 Acceder ao contido dun documento XML
 - ◆ 1.3 Xogos de caracteres
 - ◆ 1.4 Modificar o contido dun documento XML

1.2 SimpleXML

A **extensión SimpleXML** seguramente a forma máis sinxela de traballar con documentos XML dende PHP. Carga un ficheiro XML en memoria, e representa o seu contido como unha xerarquía de obxectos. Nas versións actuais da linguaxe ven habilitada por defecto.

1.2.1 Cargar un documento XML

Para cargar en memoria un documento XML empregando SimpleXML, podemos:

- Empregar unha das funcións **simplexml_load_file**, **simplexml_load_string** ou **simplexml_import_dom** dependendo da orixe do documento XML (un ficheiro, unha cadea de texto ou un nodo DOM respectivamente). Todas elas devolven un obxecto da clase **SimpleXMLElement** que representa o elemento raíz do documento XML.

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

- Empregar o constructor da clase SimpleXMLElement, indicando como primeiro parámetro a orixe do documento XML (unha cadea de texto, unha ruta a un ficheiro ou unha URL), e como terceiro *true* se a orixe é un ficheiro ou unha URL. O obxecto instanciado representa, como no caso anterior, elemento raíz do documento XML.

```
$xml = new SimpleXMLElement('exemplo.xml',0,true);
```

1.2.2 Acceder ao contido dun documento XML

Unha vez cargado en memoria o documento XML, podemos empregar as seguintes técnicas para acceder ao seu contido a partires do obxecto xerado:

- Aos **fillos dun elemento** accédese co operador frecha `->`. Se existe máis de un fillo co mesmo nome, empréganse arrays. Por exemplo, para acceder ao segundo actor do **documento de exemplo** faríamos:

```
$actor2 = $xml->actor[1];
```

- Aos **atributos dun elemento** accédese empregando arrays asociativos nos que o nome do atributo pasa a ser o nome do elemento no array.

```
$actor2 = $xml->actor[1];  
echo "O ID do actor é " . $actor2['id'];
```

- Ao **contido dun elemento** accédese directamente co nome do elemento.

```
$actor2 = $xml->actor[1];  
echo "O nome do actor é " . $actor2->nome;
```

1.2.3 Xogos de caracteres

Cando traballamos con documentos XML debemos ser conscientes dos xogos de caracteres que estamos a empregar en todos os ficheiros:

- O **documento XML** ten un xogo de caracteres que figura na súa cabeceira.

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

- A **páxina web HTML** que estamos a xerar tamén depende do xogo de caracteres que esteamos a empregar, e defínese nunha etiqueta `<meta>`.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

- Os **scripts PHP** da nosa aplicación web tamén se almacenan no xogo de caracteres que define o editor que estemos a empregar.

É unha boa práctica empregar sempre o mesmo xogo de caracteres, e moi recomendable que sexa **UTF-8**. Cando os xogos de caracteres non coinciden, deberemos facer transformacións para adaptalos.

Por exemplo, se o documento XML atópase en ISO-8859-1 e queremos amosar o seu contido nunha páxina codificada en UTF-8, deberemos empregar a función **utf8-encode**.

```
echo "O nome do actor é " . utf8-encode($actor2->nome);
```

Se estamos a xerar unha páxina UTF-8 dende un script PHP almacenado noutro xogo de caracteres, deberemos empregar a función *utf8-encode* incluso para as cadeas de texto que amosemos.

```
echo utf8-encode("O nome do actor é ") . utf8-encode($actor2->nome);
```

E de forma semellante, se traballamos cun documento XML en UTF-8 dende un script PHP xerado noutro xogo de caracteres, deberemos adaptar incluso os nomes dos elementos e atributos XML cos que traballemos.

```
echo $xml[utf8_encode('data_creación')];  
echo $xml->{utf8_encode('película')}[0]->{utf8_encode('xénero')};
```

1.2.4 Modificar o contido dun documento XML

Empregando SimpleXML tamén é posible modificar o contido existente nun documento XML, ou mesmo engadir novo contido.

Para **modificar un contido** (elemento ou atributo) existente, simplemente facemos os cambios sobre o obxecto SimpleXMLElement que o representa.

```
$xml->película[0]->título = "O Santo";
```

Se o que queremos é **engadir novos elementos**, empregamos o método **addChild** sobre o obxecto que representa ao elemento pai.

```
$xml->actor[1]->addChild('url', 'http://www.imdb.com/name/nm0000174/');
```

E de forma semellante, para **engadir novos atributos** empregamos o método **addAttribute**.

```
$xml->addAttribute('versión', '1.0');
```

Estes cambios almacénanse en memoria. Se queremos **gardalos nun ficheiro** (no orixinal ou noutro calquera), teremos que empregar o método **asXML**, que xera unha cadea de texto a partires dun nodo.

```
// Amosa o documento XML  
echo $xml->asXML();  
// Garda o documento XML nun ficheiro  
$xml->asXML('exemplo.xml');
```

--Víctor Lourido 02:19 24 jul 2013 (CEST)