Slide 13.3: PHP XML DOM (Document Object Model)
Slide 13.5: PHP SimpleXML
Home

PRINT VERSION

# PHP XML DOM (Cont.)

## Load and Output XML

This example performs the following tasks:

1. Create a `DOMDocument` object.
2. Load the XML from `note.xml` into it by using the `load(filename)` method.
3. Then the `saveXML` method puts the internal XML document into a string, so we can output it.

```
<?php
 $xmlDoc = new DOMDocument( );
 $xmlDoc->load( "note.xml " );
 print $xmlDoc->saveXML( );
?>
```

Check result   View PHP   View XML

## Looping through XML

This example performs the following tasks:

1. Initialize the XML parser.
2. Load the XML.
3. Loop through all elements of the `note.xml` element using the properties `documentElement`,

```
<?php
 $xmlDoc = new DOMDocument( );
 $xmlDoc->load( "note.xml " );
 $x = $xmlDoc->documentElement;
 foreach ( $x->childNodes AS $item ) {
  print $item->nodeName . " = ";
  print $item->nodeValue . "<br />";
 }
?>
```

Check result   View PHP   View XML

`childNodes`, `nodeName`, and `nodeValue`.

In the example above, there are empty text nodes between each element. When XML generates, it often contains white-spaces between the nodes. The XML DOM parser treats these as ordinary elements, and if you are not aware of them, they sometimes cause problems.

Slide 13.3: PHP XML DOM (Document Object Model)
Slide 13.5: PHP SimpleXML
Home

PRINT VERSION