

Slide 13.4: PHP XML DOM (cont.)

Slide 13.6: HTML DOM (Document Object Model)

[Home](#)

# PHP SimpleXML

PHP includes some XML functions. Among them, [SimpleXML](#) handles the most common XML tasks and leaves the rest for other extensions.

SimpleXML is able to perform basic tasks like:

- reading XML files,
- extracting data from XML strings, and
- editing text nodes or attributes.

An XML file [catalog.xml](#) is on the right. The program below performs the following tasks:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <game>
    <title>Super Mario All Stars</title>
    <ASIN>B000050FBJ</ASIN>
    <price>34.99</price>
    <producer>Nintendo</producer>
  </game>
  <game>
    <title>New Super Mario Bros</title>
    <ASIN>B002BRZ9G0</ASIN>
    <price>43.99</price>
    <producer>Nintendo</producer>
  </game>
  <game>
    <title>Super Mario Galaxy</title>
    <ASIN>B002BSA388</ASIN>
    <price>46.37</price>
    <producer>Nintendo</producer>
  </game>
  ...
</catalog>
```

1. Load the XML file by using the function [simplexml\\_load\\_file](#).
2. Get the name of the first element by using the function [getName](#).
3. Create a loop that will trigger on each child node, using the [children](#) function.
4. Output the element name and data for each child/descendant node.

```
<html><body>
  <?php
    $xml = simplexml_load_file( "catalog.xml" );
    echo strtoupper( $xml->getName( ) ) . "<br />";
```

```
$no = 0;
foreach( $xml->children( ) as $child1 ) {
    echo "<br />Game #" . ++$no . "<br />";
    foreach( $child1->children( ) as $child2 )
        if ( $child2->getName( ) != "producer" )
            echo $child2->getName( ) . ": <em>" . $child2 . "</em><br />";
    }
?>
</body></html>
```

[Check result](#)[View PHP](#)[View XML](#)[Reset](#)[Slide 13.4: PHP XML DOM \(cont.\)](#)[Slide 13.6: HTML DOM \(Document Object Model\)](#)[Home](#)