



## **Data processing technologies (TTD)**

**Class 3**

# Parsing XML

Up until now, we have seen that XML could be displayed using XSL, and it does a great job. But as it was mentioned, XSL isn't always supported by all browsers, forcing us to find better solutions. Using JavaScript and PHP to extract and display data from XML is definitely a great solution.

## Parsing XML with JavaScript

In order to parse an external file, in jQuery, an Ajax request must first be made. Using the find() method, the external XML file is searched to find the <book> tags within the root (<bookstore>). For every entry (books), 4 variables are used to find and store every <book> children. Finally, an output is made using concatenation.

```
<div id="results"> </div>

<script language="JavaScript">
$(document).ready(function(){
$.ajax({                                     // AJAX request
type: "GET",
url: "my_xml-02.xml",
dataType: "xml",
success: function(xml){
    var i = 0;

    $(xml).find('bookstore').children('book').each(function(){

        var sTitle = $(this).find('title').text();
        var sAuthor = $(this).find('author').text();
        var sYear = $(this).find('year').text();
        var sPrice = $(this).find('price').text();

        $("<p></p>").html("<b>" + sTitle + "</b>, " + sAuthor + ", " + sYear + ", " +
sPrice).appendTo("#results");
        i++;
    });

    var sTotalBooks = i;

    $("<p></p>").html("<b>Total of books:</b> " + sTotalBooks).prependTo("#results");
    },
    error: function() {
        $("<p></p>").html('An error occurred while processing XML file.').prependTo("#results");
    }
});
});
</script>
```

Using a jQuery shorthand for the AJAX request makes it even more simple:

```
<div id="results"> </div>

<script language="JavaScript">
$.get("my_xml-02.xml", function(data) {                                // AJAX request shorthand

    var i = 0;

    $(data).find('bookstore').children('book').each(function(){
        var sTitle = $(this).find('title').text();
        var sAuthor = $(this).find('author').text();
        var sYear = $(this).find('year').text();
        var sPrice = $(this).find('price').text();

        $("<p></p>").html("<b>" + sTitle + "</b>, " + sAuthor + ", " + sYear + ", " +
sPrice).appendTo("#results");
        i++;
    });

    var sTotalBooks = i;
    $("<p></p>").html("<b>Total of books:</b> "+ sTotalBooks).prependTo("#results");

});
</script>
```

## Parsing XML with PHP

Parsing XML using PHP is a lot more simple. It is supported by all browsers and it is executed on server-side.

All there is to do is to store the content of the XML file in a variable using the function *simplexml\_load\_file()*. Then, simply access the children of <book> tags by storing them in a variable and using a *foreach* loop with keys to display them.

```
<h2>XML with PHP</h2>

<?php
$xml=simplexml_load_file("my_xml-02.xml") or die("Error: Cannot create object");

foreach($xml->children() as $books) {
    echo "<b>" . $books->title . "</b>, ";
    echo $books->author . ", ";
    echo $books->year . ", ";
    echo $books->price . "<br>";
}
?>
```