

XSL-FO Tables

150px x 350px

Contents

Introduction	2
Download Exercise File.....	3
Create a New Oxygen Project.....	5
Create a New DTD file	6
Brief Review of XSL-FO Syntax.....	9
Create Basic XSL File	11
Review of XSL-FO Syntax	13
XSL-FO Tables Syntax.....	15
Create Basic Structure of XSL-FO Table.....	17
Enable XSL-FO Table to Pull Data from XML File	21

Introduction

This tutorial demonstrates how to create XSL-FO tables to format XML data.

Purpose

This tutorial demonstrates how to create and format XML tables using XSL formatting objects (XSL-FO).

Software and Files Needed

Resource	Description
Oxygen XML Editor 23.0	A popular software application used for XML-based editing, authoring, and publishing.
W3schools.com plant catalog XML file	A sample XML file located on the w3schools.com website

Prerequisites

- Oxygen XML Editor 23.0 or equivalent software.
- Basic knowledge of XML file structure
- Basic knowledge of XML DTD files
- Basic knowledge of XML XSL formatting for HTML and PDF output

Download Exercise File

In this section you will need to download the plant catalog XML file from [w3schools.com](https://www.w3schools.com).

Context

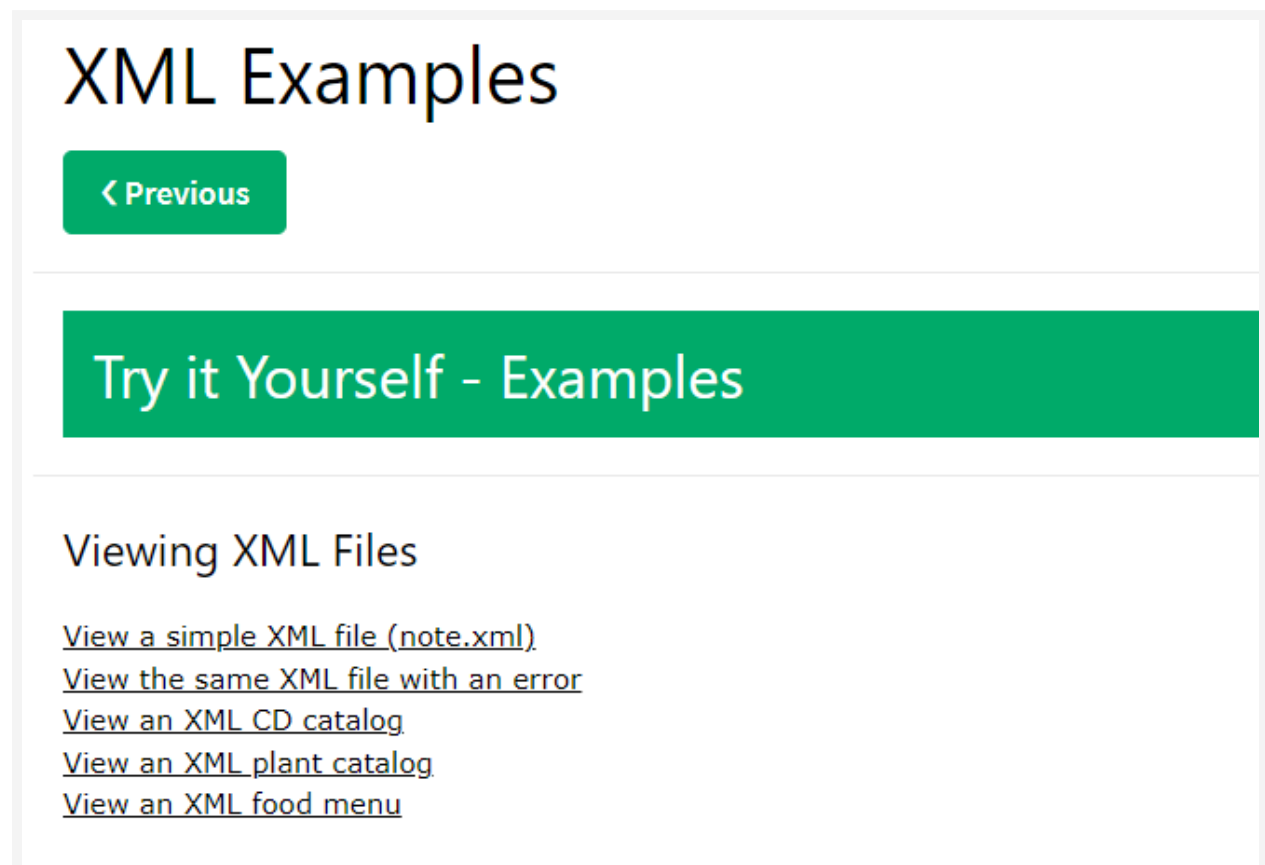
The plant catalog XML file is needed for for you to complete this tutorial. You can download it from the [w3schools.com](https://www.w3schools.com) website.

Procedure

- 1 Go to the XML Examples page on the w3schools.com website:

https://www.w3schools.com/xml/xml_examples.asp

Result:



- 2 Click on the View XML Plant Catalog hyperlink. This will open the XML file in your browser.

Result:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<CATALOG>
  ▼<PLANT>
    <COMMON>Bloodroot</COMMON>
    <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$2.44</PRICE>
    <AVAILABILITY>031599</AVAILABILITY>
  </PLANT>
  ▼<PLANT>
    <COMMON>Columbine</COMMON>
    <BOTANICAL>Aquilegia canadensis</BOTANICAL>
    <ZONE>3</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$9.37</PRICE>
    <AVAILABILITY>030699</AVAILABILITY>
  </PLANT>
  ▼<PLANT>
    <COMMON>Marsh Marigold</COMMON>
    <BOTANICAL>Caltha palustris</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Sunny</LIGHT>
    <PRICE>$6.81</PRICE>
    <AVAILABILITY>051799</AVAILABILITY>
  </PLANT>
  ▼<PLANT>
    <COMMON>Cowslip</COMMON>
    <BOTANICAL>Caltha palustris</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$9.90</PRICE>
    <AVAILABILITY>030699</AVAILABILITY>
  </PLANT>
```

- 3 Right-click on the screen. A dialog window will open.
- 4 Select SAVE AS in the dialog window.
- 5 Save the file to your Oxygen project folder for this tutorial.

Create a New Oxygen Project

Context

In this section, you will create a new Oxygen project.

Procedure

- 1 Open Oxygen XML editor.
- 2 Click on the **Project** link in the menu bar.
- 3 Click **New Project**.
- 4 Provide an appropriate name for the project, for example, plant catalog.
- 5 Browse to your project folder.
- 6 Click the **Create** button.

Result

The program will open to your project window.

Create a New DTD file

In this section you will create a simple DTD file.

Context

In this section you will create a simple DTD file using the **Learn Structure** feature.

Procedure

- 1 Open Oxygen.
- 2 Open the plant catalog XML file.
- 3 Click on the **Document** link in the top menu bar.
- 4 Next, click **XML Document**.
- 5 Then, click **Learn Structure**. Oxygen will begin to generate a DTD file.
- 6 A "**Learn Completed**" message will appear at the very bottom of the screen.
- 7 Next, click on the **Document** menu bar link again.
- 8 Click **XML document**.
- 9 Click **Save Structure** to save the DTD file.
- 10 Rename the file if desired.
- 11 Then click the **Save** button. Make sure the save location is the same as your project folder.
- 12 The top of your XML file contain a link to the DTD file.

Result:

```
<!DOCTYPE CATALOG SYSTEM "file:/D:/FILES/CLASSES/FALL%202021/ETWR%202478%20-%20XML%20DITA%20-%20McMurrey/Assignments/DITA/DITA%20Final%20Project/XSL-F0%20Tables/plant-catalog.dtd">
```

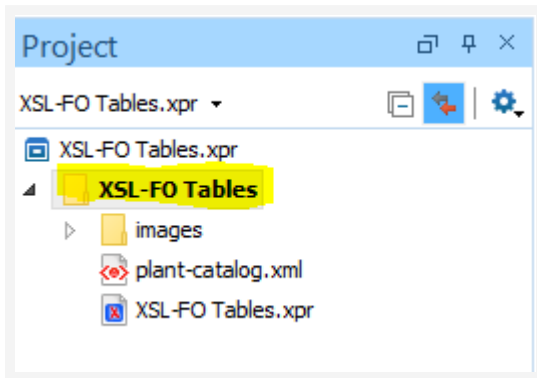
- 13 If the link shown is a fully qualified link, as shown above, convert it into a relative link by deleting all the characters before the "p" in "plant-catalog", as shown below.

Result:

```
<!DOCTYPE CATALOG SYSTEM "plant-catalog.dtd">
```

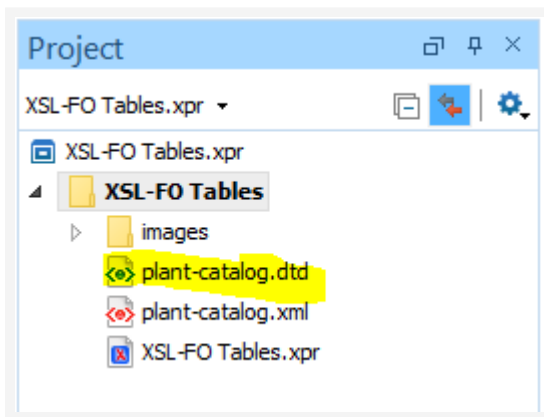
- 14 Your new DTD file may not be visible in your project folder. If not:
 - a Go to the Project viewer and right-click on the project folder. Click "Refresh."

Result:



- b The new DTD file should appear.

Result:



- 15 Your DTD file will look similar to the code below. This is not the best file syntax.

Result:

```
<!ELEMENT PLANT (#PCDATA | COMMON | BOTANICAL | ZONE | LIGHT | PRICE | AVAILABILITY)*>  
<!ELEMENT CATALOG (#PCDATA | PLANT)*>  
<!ELEMENT PRICE (#PCDATA)>  
<!ELEMENT AVAILABILITY (#PCDATA)>  
<!ELEMENT ZONE (#PCDATA)>  
<!ELEMENT LIGHT (#PCDATA)>  
<!ELEMENT COMMON (#PCDATA)>  
<!ELEMENT BOTANICAL (#PCDATA)>
```

- 16 Update the file so that it is similar to the code below.

Result:

```
<!ELEMENT CATALOG (PLANT)*>
<!ELEMENT PLANT (COMMON , BOTANICAL , ZONE , LIGHT , PRICE ,
AVAILABILITY)>
<!ELEMENT COMMON (#PCDATA)>
<!ELEMENT BOTANICAL (#PCDATA)>
<!ELEMENT ZONE (#PCDATA)>
<!ELEMENT LIGHT (#PCDATA)>
<!ELEMENT PRICE (#PCDATA)>
<!ELEMENT AVAILABILITY (#PCDATA)>
```

- 17 Validate both the DTD and XML files by clicking the red check-box icon in the toolbar. This will ensure there are no errors.

Brief Review of XSL-FO Syntax

This section will briefly review the basics of XSL-FO syntax. It assumes that you have basic knowledge of XSL-FO. If you need a more extensive review, visit [Introduction to XSL-FO](#).

Example XSL-FO Stylesheet

Below is an example of a basic XSL-FO stylesheet. We will use it as a starting point for building the tables.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
  exclude-result-prefixes="xs xd"
  version="2.0">
  <xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <!-- overall structure of the document -->
      <fo:layout-master-set>
        <fo:simple-page-master master-name="main">
          <fo:region-body region-name="body"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <!-- content of the page itself -->
      <fo:page-sequence master-reference="main">
        <fo:flow flow-name="body">
          <fo:block>
            <xsl:apply-templates/>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

Code Source: [Introduction to XSL-FO](#)

XSL-FO Primary Elements

The main elements of the XSL-FO file are shown in the table below. (Source: [Introduction to XSL-FO](#))

Element	Description
<code><fo:root></code>	Root element for the page.
<code><fo:layout-master-set></code>	Includes all the master pages for the document.
<code><fo:simple-page-master></code>	The name of a specific master page layout, named "main".
<code><fo:region-body></code>	Defines the region of the page which will contain the page's body content will display.
<code><fo:page-sequence></code>	Includes the page master, content, and formatting of the page
<code><fo:flow></code>	Defines where content should be placed.
<code><fo:block></code>	Defines page content which appears on it's own line (e.g., paragraph, heading).
<code><fo:inline></code>	Defines page content which appears on the same line as other content (e.g., bold, color applied to specific words in a paragraph).

Create Basic XSL File

In this section, you will create a basic XSL file.

Context

In this section you will create a basic XSL file. This file will then be used as the starting point for creating a table.

Procedure

- 1 Open Oxygen
- 2 Create a new XSL file. Ensure that you save it to the same folder as your project.
- 3 Copy the code below and paste it into your new XSL file. This is the code from the topic [Review of XSL-FO Syntax](#).

Result:


```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <!-- overall structure of the document -->
      <fo:layout-master-set>
        <fo:simple-page-master master-name="sample">
          <fo:region-body region-name="body"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <!-- content of the page itself -->
      <fo:page-sequence master-reference="sample">
        <fo:flow flow-name="body">
          <fo:block>
            <xsl:apply-templates/>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

Source: [Introduction to XSL-FO](#)

- 4 Run the PDF output transformation on the file.
- 5 Your output should appear similar to the screen capture below as basically an "unformatted" PDF file. 

Review of XSL-FO Syntax

This section will review the basics of XSL formatting objects (XSL-FO) syntax. It assumes that you have basic knowledge of XSL-FO.

Example XSL-FO Stylesheet

Below is an example of a basic XSL-FO stylesheet. We will use it as a starting point for building the tables.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
  exclude-result-prefixes="xs xd"
  version="2.0">
  <xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <!-- overall structure of the document -->
      <fo:layout-master-set>
        <fo:simple-page-master master-name="main">
          <fo:region-body region-name="body"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <!-- content of the page itself -->
      <fo:page-sequence master-reference="main">
        <fo:flow flow-name="body">
          <fo:block>
            <xsl:apply-templates/>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

Code Source: [Introduction to XSL-FO](#)

XSL-FO Primary Elements

The main elements of the XSL-FO file are shown in the table below. (Source: [Introduction to XSL-FO](#))

Element	Description
<code><fo:root></code>	Root element for the page.
<code><fo:layout-master-set></code>	Includes all the master pages for the document.
<code><fo:simple-page-master></code>	The name of a specific master page layout, named "main".
<code><fo:region-body></code>	Defines the region of the page which will contain the page's body content will display.
<code><fo:page-sequence></code>	Includes the page master, content, and formatting of the page
<code><fo:flow></code>	Defines where content should be placed.
<code><fo:block></code>	Defines page content which appears on it's own line (e.g., paragraph, heading).
<code><fo:inline></code>	Defines page content which appears on the same line as other content (e.g., bold, color applied to specific words in a paragraph).

XSL-FO Tables Syntax

In this section, we will overview the basic syntax of XSL-FO tables.

Introduction

The syntax for XSL-FO tables is similar to that of [HTML tables](#).

They have a parent table tag, headers, rows, and cells.

Syntax Summary

Below is the syntax for a simple table with a header, body, two rows, and two columns.

```
<fo:table>
  <fo:table-header>
    <fo:table-cell><fo:block>Heading A</fo:block></fo:table-cell>
    <fo:table-cell><fo:block>Heading B</fo:block></fo:table-
cell>
  </fo:table-header>
  <fo:table-body>
    <fo:table-row>
      <fo:table-cell><fo:block>CELL VALUE R1 C1</fo:block></fo:table-
cell>
      <fo:table-cell><fo:block>CELL VALUE R1 C2</fo:block></fo:table-
cell>
    </fo:table-row>
    <fo:table-row>
      <fo:table-cell><fo:block>CELL VALUE R2 C1</fo:block></fo:table-
cell>
      <fo:table-cell><fo:block>CELL VALUE R2 C2</fo:block></fo:table-
cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>
```

Syntax Details

Below is a description of the table's syntax elements.

Syntax	Details
<code><fo:table></code>	Parent element of the entire table; encloses all the table elements

Syntax	Details
<code><fo:table-header></code>	Designates the first row of the table as the header.
<code><fo:table-body></code>	Defines the main body of the table; does not include the header
<code><fo:table-row></code>	Defines a table row; parent element of any cells in the row
<code><fo:table-cell></code>	Defines a single table cell
<code><fo:block></code>	Defines a region within a cell to output literal text or data from the XML file

Table Content

Table Header

For this tutorial, you will manually enter the table header text (e.g., "Common Name", "Botanical Name", "Hardiness Zone").

Table Cells

The table cell values will be dynamically pulled from the XML file using `<xsl:value-of select="" />` entries.

Create Basic Structure of XSL-FO Table

In this section, we will build the basic structure for the XSL-FO table.

Context

In this section, we will use the XSL-FO table syntax from the previous section to begin building our table.

Procedure

- 1 Below is our starter code. Copy and paste it into your XSL-FO file if you have not already done so.

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
  exclude-result-prefixes="xs xd"
  version="2.0">
  <xsl:template match="/CATALOG">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <!-- overall structure of the document -->
      <fo:layout-master-set>
        <fo:simple-page-master master-name="main">
          <fo:region-body region-name="body"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <!-- content of the page itself -->
      <fo:page-sequence master-reference="main">
        <fo:flow flow-name="body">
          <fo:block>
            <xsl:apply-templates/>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

- 2 Note that `<xsl:template match="/CATALOG" >` is set to the parent element of the XML file, CATALOG.
- 3 Go to the `<!-- content of the page itself -->` section.

Result:

```
<!-- content of the page itself -->
  <fo:page-sequence master-reference="main">
    <fo:flow flow-name="body">
      <fo:block>
        <xsl:apply-templates/>
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
```

- 4 Replace `<xsl:apply-templates/>` with the `<fo:table>` code below.

Result:

```
<fo:table>

</fo:table>
```

- 5 Add a table header row with three table cells containing `<fo:block>` tags.

Result:

```
<fo:table>

  <fo:table-header>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
  </fo:table-header>

</fo:table>
```

- 6 Add the following values for the table header cells: Common Name, Botanical Name, and Hardiness Zone.

Result:

```
<fo:table-header>
  <fo:table-cell><fo:block>Common Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Botanical Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Hardiness Zone</fo:block></fo:table-
cell>
</fo:table-header>
```

- 7 After the table header, add a table body section.

Result:

```
<fo:table-body>  
  
</fo:table-body>
```

- 8 Within the `<fo:table-body></fo:table-body>` cells, add a table row (`<fo:table-row></fo:table-row>`). Then, add three table cells (`<fo:table-cell></fo:table-cell>`).

Result: Add `<fo:block></fo:block>` tags within the table cells.

```
<fo:table-row>  
  <fo:table-cell><fo:block></fo:block></fo:table-cell>  
  <fo:table-cell><fo:block></fo:block></fo:table-cell>  
  <fo:table-cell><fo:block></fo:block></fo:table-cell>  
</fo:table-row>
```

Result

Your table should look similar to the code below.

```
<fo:table>  
  
  <fo:table-header>  
    <fo:table-cell><fo:block>Common Name</fo:block></fo:table-cell>  
    <fo:table-cell><fo:block>Botanical Name</fo:block></fo:table-cell>  
    <fo:table-cell><fo:block>Hardiness Zone</fo:block></fo:table-cell>  
  </fo:table-header>  
  
  <fo:table-body>  
    <fo:table-row>  
      <fo:table-cell><fo:block></fo:block></fo:table-cell>  
      <fo:table-cell><fo:block></fo:block></fo:table-cell>  
      <fo:table-cell><fo:block></fo:block></fo:table-cell>  
    </fo:table-row>  
  </fo:table-body>  
  
</fo:table>
```

Enable XSL-FO Table to Pull Data from XML File

In this section we will add coding to the table so that it pulls data from the XML file.

Context

In this section we will continue building our table.

We will enable the table to pull data from our XML file using `<xsl:for-each />`.

Procedure

- 1 Below is the latest code from the [previous section](#).

Result:

```
<fo:table>

<fo:table-header>
  <fo:table-cell><fo:block>Common Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Botanical Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Hardiness Zone</fo:block></fo:table-
cell>
</fo:table-header>

<fo:table-body>
  <fo:table-row>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
  </fo:table-row>
</fo:table-body>

</fo:table>
```

- 2 Immediate after the `<fo:table-body>` opening tag, type `<xsl:for-each select="PLANT">`. The closing `</xsl:for-each>` closing tag should be placed immediately before the `</fo:table-body>` closing tag.

Result: This sets our point of reference in the XML file to the PLANT element so that we can loop through its child elements..

```
<fo:table-body>
<xsl:for-each select="PLANT">
  <fo:table-row>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
    <fo:table-cell><fo:block></fo:block></fo:table-cell>
  </fo:table-row>
</xsl:for-each>
</fo:table-body>
```

3 Next, you will add code to populate each `<fo:table-cell></fo:table-cell>` with a value from the XML file using the `<xsl:value-of select="" />` tag.

4 Type `<xsl:value-of select="COMMON" />` into the first table cell.

Result:

```
<fo:table-row>
  <fo:table-cell><fo:block><xsl:value-of
select="COMMON" /></fo:block></fo:table-cell>
  <fo:table-cell><fo:block></fo:block></fo:table-cell>
  <fo:table-cell><fo:block></fo:block></fo:table-cell>
</fo:table-row>
```

5 Type `<xsl:value-of select="BOTANICAL" />` into the second table cell.

Result:

```
<fo:table-row>
  <fo:table-cell><fo:block><xsl:value-of
select="COMMON" /></fo:block></fo:table-cell>
  <fo:table-cell><fo:block><xsl:value-of
select="BOTANICAL" /></fo:block></fo:table-cell>
  <fo:table-cell><fo:block></fo:block></fo:table-cell>
</fo:table-row>
```

6 Type `<xsl:value-of select="ZONE" />` into the third table cell.

Result:

```
<fo:table-row>
  <fo:table-cell><fo:block><xsl:value-of
select="COMMON"/></fo:block></fo:table-cell>
  <fo:table-cell><fo:block><xsl:value-of
select="BOTANICAL"/></fo:block></fo:table-cell>
  <fo:table-cell><fo:block><xsl:value-of
select="ZONE"/></fo:block></fo:table-cell>
</fo:table-row>
```

7 Your table code should appear as shown below.

Result:

```
<fo:table>

<fo:table-header>
  <fo:table-cell><fo:block>Common Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Botanical Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Hardiness Zone</fo:block></fo:table-
cell>
</fo:table-header>

<fo:table-body>
  <xsl:for-each select="PLANT">
    <fo:table-row>
      <fo:table-cell><fo:block><xsl:value-of
select="COMMON"/></fo:block></fo:table-cell>
      <fo:table-cell><fo:block><xsl:value-of
select="BOTANICAL"/></fo:block></fo:table-cell>
      <fo:table-cell><fo:block><xsl:value-of
select="ZONE"/></fo:block></fo:table-cell>
    </fo:table-row>
  </xsl:for-each>
</fo:table-body>

</fo:table>
```

8 Your entire XSLT file should appear as shown below.

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xd="http://www.oxygenxml.com/ns/doc/xsl"
  exclude-result-prefixes="xs xd"
  version="2.0">
<xsl:template match="/CATALOG">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

<!-- overall structure of the document -->
<fo:layout-master-set>
  <fo:simple-page-master master-name="main">
    <fo:region-body region-name="body"/>
  </fo:simple-page-master>
</fo:layout-master-set>

<!-- content of the page itself -->
<fo:page-sequence master-reference="main">
  <fo:flow flow-name="body">
    <fo:block>

<!-- <xsl:apply-templates/>-->
<fo:table>

<fo:table-header>
  <fo:table-cell><fo:block>Common Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Botanical Name</fo:block></fo:table-cell>
  <fo:table-cell><fo:block>Hardiness Zone</fo:block></fo:table-
cell>
</fo:table-header>

<fo:table-body>
  <xsl:for-each select="PLANT">
    <fo:table-row>
      <fo:table-cell><fo:block><xsl:value-of
select="COMMON"/></fo:block></fo:table-cell>
      <fo:table-cell><fo:block><xsl:value-of
select="BOTANICAL"/></fo:block></fo:table-cell>
      <fo:table-cell><fo:block><xsl:value-of
select="ZONE"/></fo:block></fo:table-cell>
    </fo:table-row>
  </xsl:for-each>
```

```
</fo:table-body>  
  
</fo:table>  
  
</fo:block>  
</fo:flow>  
</fo:page-sequence>  
  
</fo:root>  
</xsl:template>  
  
</xsl:stylesheet>
```

9 Below is the PDF output.

Result:

Common Name	Botanical Name	Hardiness Zone
Bloodroot	Sanguinaria canadensis	4
Columbine	Aquilegia canadensis	3
Marsh Marigold	Caltha palustris	4
Cowslip	Caltha palustris	4
Dutchman's-Breeches	Dicentra cucullaria	3
Ginger, Wild	Asarum canadense	3
Hepatica	Hepatica americana	4
Liverleaf	Hepatica americana	4
Jack-In-The-Pulpit	Arisaema triphyllum	4
Mayapple	Podophyllum peltatum	3
Phlox, Woodland	Phlox divaricata	3
Phlox, Blue	Phlox divaricata	3
Spring-Beauty	Claytonia Virginica	7