

XML Technologies: Exercise Solutions 5

XSL-FO, XML Parsing

Questions and instructions are displayed in a font with normal weight, answers are set in **bold**.

Bear in mind that even though the sample solution might not consider your approach to solve a particular problem, this does by no means invalidate your solution.

1 An XML-Wing Fighter (3 Points)

Using the `scaffold.xml` provided, fill in the blanks to create a FO document, which:

- First shows the SVG image
- Then shows the text for every `<p>`-element in the `xwing-wiki.xml`

The XSLT stylesheet `xwing.xml` can be found attached. It transforms the XML document `xwing-wiki.xml`, but there is a named template which also copies all the content in `xwing.svg` to the output.

The result of the transformation is `xwing.fo`, which is an XSL-FO document. It has a very simple page master that only defines the margins of the page and the dimensions of the body region. The page sequence has blocks that are inside a single flow element.

Finally, the rendered PDF is `xwing.pdf`. This version was rendered with the open-source FO processor *Apache FOP 2.0*.

2 Parsing XML (2 points)

Answer the following questions:

- DOM means “document object model”. What does that mean? Why is it sometimes called a *tree*?

It means that parsing with the DOM method creates an abstract *model*

of the actual, physical file. An applications never has a direct access to stored XML documents, but only get to see models of documents. Because of the hierarchical organisation of an XML document, the DOM model is an up-side down tree (with the trunk at the top) that has a single root node.

- Which method uses a lot more RAM? Why?

DOM uses a lot more RAM. This is because parsing with the DOM method creates an abstract model of the XML document that is held in memory at all times. Therefore, RAM usage will be roughly proportional to the size of the XML document that is to be parsed. On the other hand, SAX uses very little RAM and its RAM usage does not depend on the size of the XML document.

- Which of them is an event-driven and streaming method? Why?

SAX is event-driven and streaming. It should be thought of in the following way: parsing the document generates a stream of events that are presented to an application. For each type of event that occurs, a programmer can either choose to ignore the event or react to it in a certain way.

- Which of them is more suitable for use with XPath? Why? Please substantiate your claim with an XPath expression, i.e. find an XPath expression which will not be possible to execute using both approaches.

A DOM representation (or, more precisely, its variant XDM) is used to evaluate XPath expressions in XML documents. DOM trees are persistent, while SAX events are very short-lived: if no action is taken for a certain event, the event will be discarded immediately. Or, put another way: with SAX, there is no way of going back in the document.

However, there are XPath expressions that go back, for instance:

```
..
parent::*
preceding-sibling::cichlid[1]
```

Which is possible if the document was parsed with the DOM approach, but not with with SAX.