

# XML Documents



# eXtensible Markup Language

- A textual document format used all over the web is ***XML***
  - Used to represent ***hierarchically organized*** data for “ease of use” both by humans and computers
  - Had its origins in SGML (Standard Generalized Markup Language) from the 1980s
  - Became a standard in the late 1990s

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>
<book printISBN="978-1-118-06331-6"
  webISBN="1-118063-31-7"
  pubDate="Dec 20 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<book printISBN="978-1-118-06331-6"
```

```
webISBN="1-118-06331-7"
```

```
pubDate="Dec 20 2011">
```

```
<author>Cay Horstmann</author>
```

```
<title>
```

```
    Java for Everyone: Late Objects
```

```
</title>
```

```
...
```

```
</book>
```

An ***XML declaration*** is the first line of well-formed XML file.

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<book printISBN="978-1-118-0637-7"
```

```
  webISBN="1-118063-31-7"
```

```
  pubDate="Dec 20 2011">
```

```
  <author>Cay Horstmann</author>
```

```
  <title>
```

```
    Java for Everyone: Late Objects
```

```
  </title>
```

```
  ...
```

```
</book>
```

This is not  
part of the  
actual  
document/file!

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<book printISBN="978-1-118-063-31-7"
      webISBN="1-118063-31-7"
      pubDate="Dec 20 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

This is an  
author  
*element.*

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8" ?>
<book printISBN="978-1-118-063-31-2"
      webISBN="1-118063-31-2"
      pubDate="Dec 29 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

This is a ***start tag*** for an author element.

# Example XML Document/File

```
<?xml version="1.0" encoding="
<book printISBN="978-1-118-063
  webISBN="1-118063-31-7"
  pubDate="Dec 20 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

This is an *end tag* for that author element.



# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8" />
<book printISBN="978-1-118-063-31-7"
      webISBN="1-118063-31-7"
      pubDate="Dec 20 2012" />
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

The **content** for this `author` element is everything between its start and end tags.

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8" >
<book printISBN="978-1-118-063-31-5"
      webISBN="1-118063-31-5"
      pubDate="Dec 26 2011" >
  <author></author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

If the `author` element had no content, then you *might* find this, or...

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8" ?>
<book printISBN="978-1-118-063-31-7"
      webISBN="1-118063-31-7"
      pubDate="Dec 20 2011">
  <author />
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```

... or you *might* find  
a single  
***self-closing tag***  
like this.

# Example XML Document/File

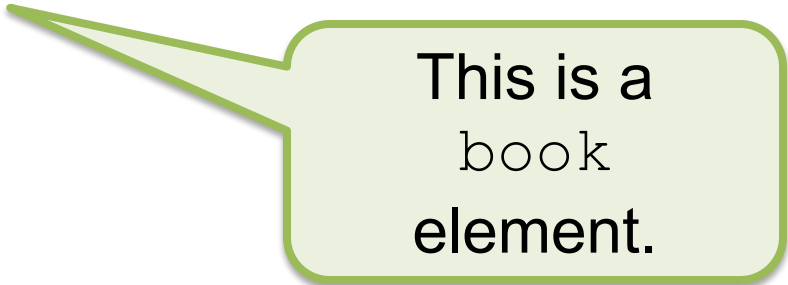
```
<?xml version="1.0" encoding="UTF-8"?>  
<book printISBN="978-1-118-06331-6"  
  webISBN="1-118063-31-7"  
  pubDate="Dec 20 2011">  
  <author>Cay Horstmann</author>  
  <title>  
    Java for Everyone: Late Objects  
  </title>  
  ...  
</book>
```



This is a  
title  
element.

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>  
<book printISBN="978-1-118-06331-6"  
  webISBN="1-118063-31-7"  
  pubDate="Dec 20 2011">  
  <author>Cay Horstmann</author>  
  <title>  
    Java for Everyone: Late Objects  
  </title>  
  ...  
</book>
```



This is a  
book  
element.

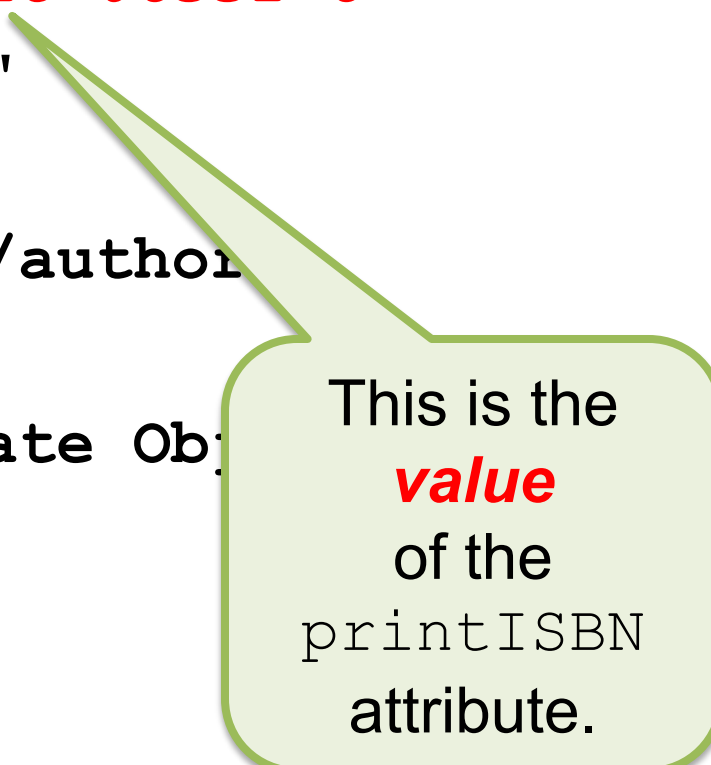
# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>
<book printISBN="978-1-118-06331-6"
  webISBN="1-118063-31-7"
  pubDate="Dec 20 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late
  </title>
  ...
</book>
```

This is the *name* of an *attribute* of the book element.

# Example XML Document/File

```
<?xml version="1.0" encoding="UTF-8"?>
<book printISBN="978-1-118-06331-6"
  webISBN="1-118063-31-7"
  pubDate="Dec 20 2011">
  <author>Cay Horstmann</author>
  <title>
    Java for Everyone: Late Objects
  </title>
  ...
</book>
```



This is the **value** of the printISBN attribute.

# Recursive Structure

- An XML document (without the XML declaration in the first line) is made up of:
  - A ***top-level element***
  - A string of zero or more ***child elements*** of the top-level element, each of which is exactly like the top-level element of an XML document
- Notice the similarity to a tree: the structure of an XML document is also ***recursive***
- Information it represents is ***hierarchical***



# Can You Find All The Errors?

```
<?xml version="1.0" encoding="UTF-8"?>
<pony>
  <unicorn mark="three lozenges">
    <color>cyan</unicorn>
  </color>
  <unicorn>Twilight<color>cyan</color>Sparkle</unicorn>
  <unicorn mark="dolphins" version="G4" />
</pony>
<dragon>
  <youth>Spike
</dragon>
```

# Resources

- *Big Java Late Objects*, Section 23.1 (but not the rest of Chapter 23)
  - <http://proquest.safaribooksonline.com.proxy.lib.ohio-state.edu/book/programming/java/9781118087886/chapter-23-xml/navpoint-172>
- Wikipedia: XML
  - <http://en.wikipedia.org/wiki/XML>

Report "Java for Everyone, Late Objects". Please fill this form, we will try to respond as soon as possible. Your name. Email. Reason. - Select Reason- Copyright complaint Illegal/Unlawful Spam Other Terms Of Service Violation. Description. Submit. Download "Java for Everyone, Late Objects". We are a sharing community. So please help us by uploading 1 new document or like us to download: Upload document file. Or like to download immediately. Close. Copyright © 2017 TUXDOC Inc. Sams Teach Yourself Java,ç in 21 Days Page 7 . programming and web publishing, including Sams Java Object-Oriented Programming (OOP) with Java. 1,770 Pages 2013 8.59 MB 11,923 Downloads. 4.6 Jb0120: Java OOP: A Gentle Introduction to Java Programming . to take that first step than Java, Java, Java - Trinity College. 865 Pages 2012 6.28 MB 6,551 Downloads. It took two or three days for a team of several pro- CHAPTER 0 ç Exercises 21 Learning Java™s jjj-os.dvi The Conscious Parent: Transforming Ourselves, Empowering Our Children. 334 Pages 2017 1.55 MB 93,440 Downloads. The\_Conscious\_ Are you sure you want to report the file Java for Everyone\_Late Object - Cay S. Horstmann\_529.pdf having link [http://97.98.91.79:8080/legacy/get/PDF/529/Book/Java%20for%20Everyone\\_%20Late%20Object%20-%20Cay%20S.%20Horstmann\\_529.pdf](http://97.98.91.79:8080/legacy/get/PDF/529/Book/Java%20for%20Everyone_%20Late%20Object%20-%20Cay%20S.%20Horstmann_529.pdf). Java for Everyone\_Late Object - Cay S. Horstmann\_529.pdf reported successfully, action will be taken soon. Close. Java for Everyone\_Late Object - Cay S. Horstmann\_529.pdf. Name: Java for Everyone\_Late Object - Cay S. Horstmann\_529.pdf. Size: 18.47 MB. Through this website you are able to link to other websites which are not under the control of FilePursuit.

Java for Everyone provides them with a more user-friendly, graphics-intensive design that conveys complex concepts simply and effectively. It differs from other books in the market by focusing on topics such as loops and graphical interfaces, while avoiding a focus on object-orientation. The emphasis is placed on simple programs that use pre-built data structures and algorithms whenever possible. The examples and case studies explore practical applications that programmers, computer scientists, and engineers will find useful, such as processing tabular data, harvesting Web information, creatin , "JAVA: Easy Java Programming For Beginners- Your Step-By-Step Guide to Learning Java Program java interview questions: Top 20 java interview programs and answers. 457 PagesÂ·2017Â·1.35 MBÂ·19,433 DownloadsÂ·New! Java Interview Question is here to help you through the INTERVIEW process, teaching you what you Emotional Healing for Dummies. 379 PagesÂ·2009Â·7.64 MBÂ·67,667 Downloads. and depression to functional syndromes like irritable bowel, fibromyalgia Dummies, is a member of the Association for Load more similar PDF files. PDF Drive investigated dozens of problems and listed the