Use XPath to obtain (in more than one way if possible):

1. All titles of the document

//title

1. All titles of a book of the document

//book/title

bookShop/book//title

1. All elements of a chapter

//chapter/\*

1. All elements of a chapter in a section

bookShop/book/section/chapter/\*

1. All elements

//bookShop

bookShop

/bookShop

1. Titles that have a chapter 1

//chapter[@num=1]/title

1. All child attributes of the last book

//book[last()]//@\*

1. The third book with all its information

//book[3]

1. All elements with attributes

bookshop

1. All elements with a category attribute

/bookShop/\*[@category]

1. Books with no cover attribute

Book[not(@cover)]

1. Elements with two chapter children

//book[count(chapter)<2]

1. Elements with at least 3 children

//book[count(\*)<2]

1. Nodes beginning with letter b

//\*[starts-with(name(),'b')]

1. Book’s children nodes beginning with letter a

//book/\*[starts-with(name(),'a')]

1. Book’s children nodes containing io

//book/\*[contains(name(),'io')]

1. Children elements with at least one attribute that contains cub in its name

//\*/\*[contains(@\*, 'cub')]

1. Title of those elements with a category attribute that contains in in its value

//book[contains(@category, 'in')]/title

1. Title of those books with at least one author that contains James in its value

//book/author[contains(text(), 'James')]/preceding-sibling::title

1. Titles whose author’s name value has more than 15 characters

//author[string-length(text())>15]/preceding-sibling::title

1. Years older than 2003

//\*/book[year > 2003]/year

1. Titles written in English and newer than 2003

//\*/book/title[contains(@\*, 'en')]/ancestor::\*/book[year > 2003]/title