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

a. All titles of the document

Expression: `//title`

```
<title language="en">Everyday Italian</title>
<title language="en">Harry Potter</title>
<title language="en">XQuery Kick Start</title>
<title>XML The foundation of XQuery</title>
<title>Understanding the XPath specification</title>
<title language="en">Learning XML</title>
<title>Introduction</title>
<title>Markup and Core Concepts</title>
```

b. All titles of a book of the document

Expression: `//book/title`

```
<title language="en">Everyday Italian</title>
<title language="en">Harry Potter</title>
<title language="en">XQuery Kick Start</title>
<title language="en">Learning XML</title>
```

c. All elements of a chapter

Expression: `//chapter`

```
<chapter num="1">

    <title>XML The foundation of XQuery</title>

    <npages>14</npages>

</chapter>
<chapter num="2">

    <title>Understanding the XPath specification</title>

    <npages>35</npages>

</chapter>
<chapter num="1">

    <title>Introduction</title>

    <npages>24</npages>

</chapter>
<chapter num="2">

    <title>Markup and Core Concepts</title>

    <npages>34</npages>

</chapter>
```

d. All elements of a chapter in a section

Expression: `//section/chapter/*`

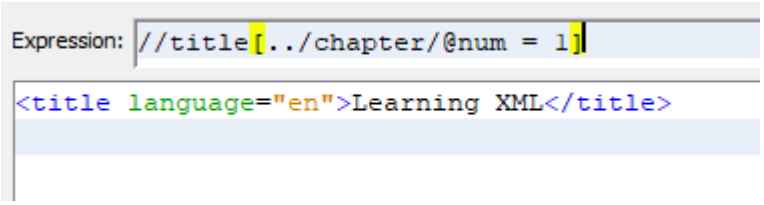
```
<title>XML The foundation of XQuery</title>
<npages>14</npages>
<title>Understanding the XPath specification</title>
<npages>35</npages>
```

e. All elements

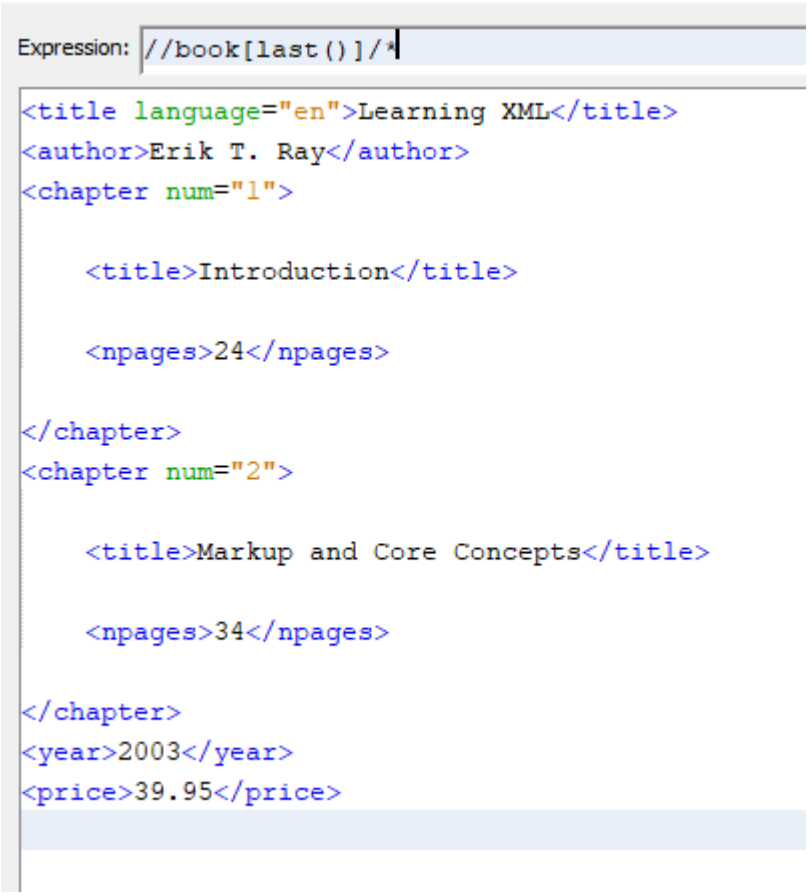
Expression: `/*`

```
<bookShop>

    <book category="cooking">

        <title language="en">Everyday Italian</title>

        <author>Giada De Laurentiis</author>

        <year>2005</year>

        <price>30.00</price>

    </book>

    <book category="children">

        <title language="en">Harry Potter</title>

        <author>J K. Rowling</author>

        <year>2005</year>

        <price>29.99</price>

    </book>

    <book category="web">

        <title language="en">XQuery Kick Start</title>
```

f. Titles that have a chapter 1



g. All child attributes of the last book



h. The third book with all its information

Expression: `//book[3]`

```
<book category="web">

    <title language="en">XQuery Kick Start</title>

    <author>James McGovern</author>

    <author>Per Bothner</author>

    <author>Kurt Cagle</author>

    <author>James Linn</author>

    <author>Vaidyanathan Nagarajan</author>

    <section par="1">

        <chapter num="1">

            <title>XML The foundation of XQuery</title>

            <npages>14</npages>

        </chapter>

        <chapter num="2">

            <title>Understanding the XPath specification</title>

            <npages>35</npages>

        </chapter>

    </section>

    <year>2003</year>

    <price>49.99</price>

</book>
```

i. All elements with attributes

XPath

Expression: `//*[@*]`

```
<book category="cooking">

    <title language="en">Everyday Italian</title>

    <author>Giada De Laurentiis</author>

    <year>2005</year>

    <price>30.00</price>

</book>
<title language="en">Everyday Italian</title>
<book category="children">

    <title language="en">Harry Potter</title>

    <author>J K. Rowling</author>

    <year>2005</year>

    <price>29.99</price>

</book>
<title language="en">Harry Potter</title>
<book category="web">

    <title language="en">XQuery Kick Start</title>

    <author>James McGovern</author>

    <author>Per Bothner</author>

    <author>Kurt Cagle</author>

    <author>James Linn</author>

    <author>Vaidyanathan Nagarajan</author>

    <section par="1">

        <chapter num="1">
```

j. All elements with a category attribute

Expression: `//*[@category]`

```
<book category="cooking">
        <title language="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
<book category="children">
        <title language="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
<book category="web">
        <title language="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <section par="1">
            <chapter num="1">
                <title>XML The foundation of XQuery</title>
                <npages>14</npages>
            </chapter>
            <chapter num="2">
                <title>Understanding the XPath specification</title>
                <npages>35</npages>
            </chapter>
        </section>
        <year>2003</year>
        <price>49.99</price>
    </book>
<book category="web" cover="paperback">
```

k. Books with no cover attribute

Expression: `//book[not(@cover)]`

```
<book category="cooking">

    <title language="en">Everyday Italian</title>

    <author>Giada De Laurentiis</author>

    <year>2005</year>

    <price>30.00</price>

</book>
<book category="children">

    <title language="en">Harry Potter</title>

    <author>J K. Rowling</author>

    <year>2005</year>

    <price>29.99</price>

</book>
<book category="web">

    <title language="en">XQuery Kick Start</title>

    <author>James McGovern</author>

    <author>Per Bothner</author>

    <author>Kurt Cagle</author>

    <author>James Linn</author>

    <author>Vaidyanathan Nagarajan</author>

    <section par="1">

        <chapter num="1">

            <title>XML The foundation of XQuery</title>

            <npages>14</npages>

        </chapter>
```

I. Elements with two chapter children

Expression: `//*[count(chapter) = 2]`

```
<section par="1">

    <chapter num="1">

        <title>XML The foundation of XQuery</title>

        <npages>14</npages>

    </chapter>

    <chapter num="2">

        <title>Understanding the XPath specification</title>

        <npages>35</npages>

    </chapter>

</section>
<book category="web" cover="paperback">

    <title language="en">Learning XML</title>

    <author>Erik T. Ray</author>

    <chapter num="1">

        <title>Introduction</title>

        <npages>24</npages>

    </chapter>

    <chapter num="2">

        <title>Markup and Core Concepts</title>

        <npages>34</npages>

    </chapter>

    <year>2003</year>

    <price>39.95</price>

</book>
```

m. Elements with at least 3 children

Expression: `//*[count(*) >= 3]`

```
<bookShop>

    <book category="cooking">

        <title language="en">Everyday Italian</title>

        <author>Giada De Laurentiis</author>

        <year>2005</year>

        <price>30.00</price>

    </book>

    <book category="children">

        <title language="en">Harry Potter</title>

        <author>J K. Rowling</author>

        <year>2005</year>

        <price>29.99</price>

    </book>

    <book category="web">

        <title language="en">XQuery Kick Start</title>

        <author>James McGovern</author>

        <author>Per Bothner</author>

        <author>Kurt Cagle</author>

        <author>James Linn</author>

        <author>Vaidyanathan Nagarajan</author>

        <section par="1">

            <chapter num="1">

                <title>XML The foundation of XQuery</title>
```

n. Nodes beginning with letter b

Expression: `//*[starts-with(name(), 'b')]`

```
<bookShop>
    <book category="cooking">
        <title language="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title language="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web">
        <title language="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <section par="1">
```

o. Book’s children nodes beginning with letter a

Expression: `//book/*[starts-with(name(), 'a')]`

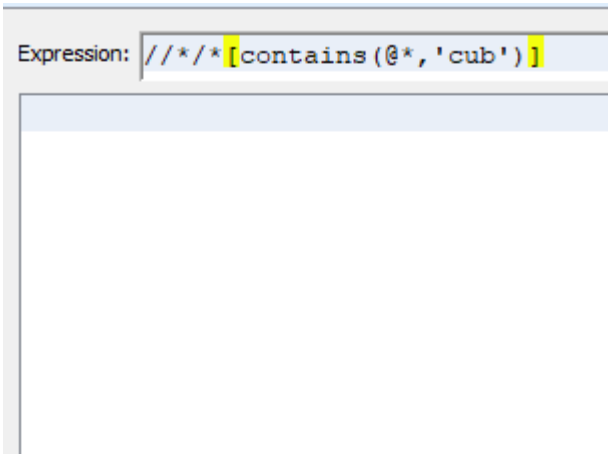
```
<author>Giada De Laurentiis</author>
<author>J K. Rowling</author>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<author>Erik T. Ray</author>
```

p. Book’s children nodes containing io

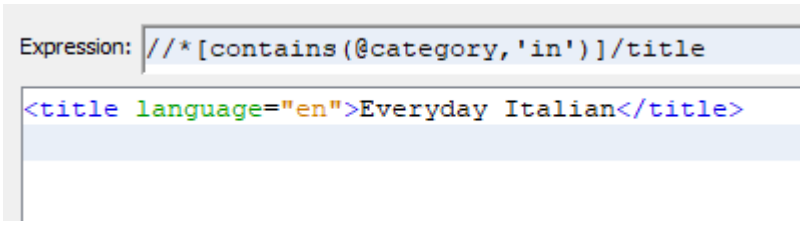
Expression: `//book/*[contains(name(),'io')]`

```
<section par="1">

    <chapter num="1">

        <title>XML The foundation of XQuery</title>

        <npages>14</npages>

    </chapter>

    <chapter num="2">

        <title>Understanding the XPath specification</title>

        <npages>35</npages>

    </chapter>

</section>
```

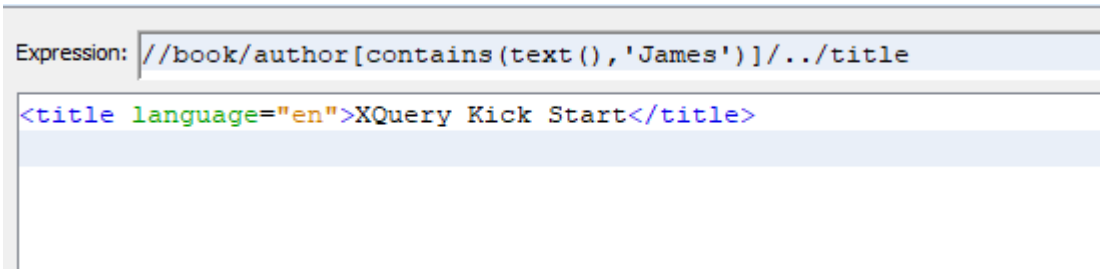
q. Children elements with at least one attribute that contains cub in its name



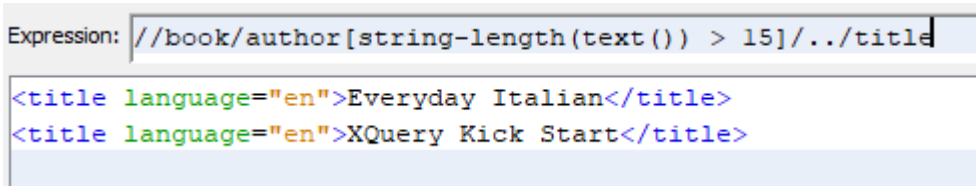
r. Title of those elements with a category attribute that contains in in its value



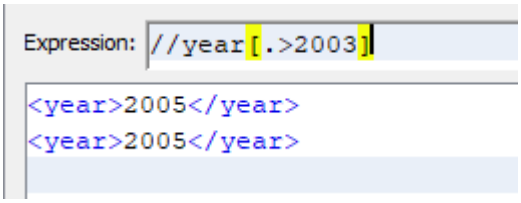
s. Title of those books with at least one author that contains James in its value



t. Titles whose author's name value has more than 15 characters



u. Years older than 2003



v. Titles written in English and newer than 2003