Assignment - 9
Aman Batra

Implemented in R and FreeFormatter.com

Query 1

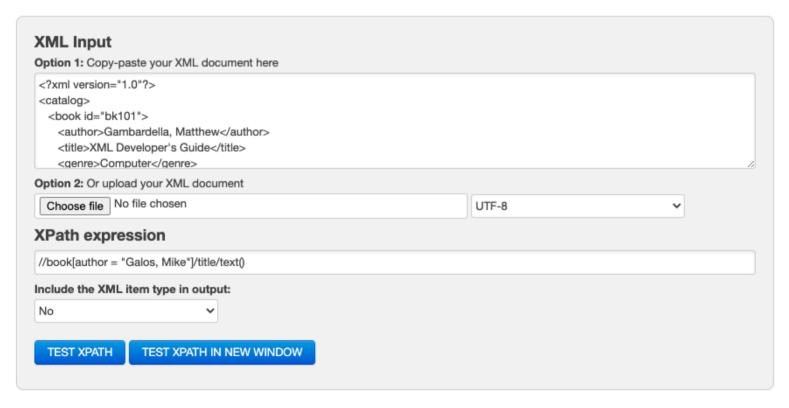## What is the title of the book written by "Galos, Mike"?

```
# QUERY 1 - What is the title of the book written by "Galos, Mike"?

bookname = xpathSApply(xmlDocument,'//book[author="Galos, Mike"]/title/text()', xmlValue)
bookname
```

```
## [1] "Visual Studio 7: A Comprehensive Guide"
```

XPath Query(relative) : //book[author="Galos, Mike"]/title/text()

XPath Query(absolute) : /catalog/book[author="Galos, Mike"]/title/text()

Output from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Screenshot from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Query 2

What are the publication dates of all books written by "O'Brien, Tim"?.

```
#QUERY 2 - What are the publication dates of all books written by "O'Brien, Tim"?.

dfdate = xpathSApply(xmlDocument,'//book[author="O\'Brien, Tim"]/publish_date/text()', xmlValue)
dfdate
```
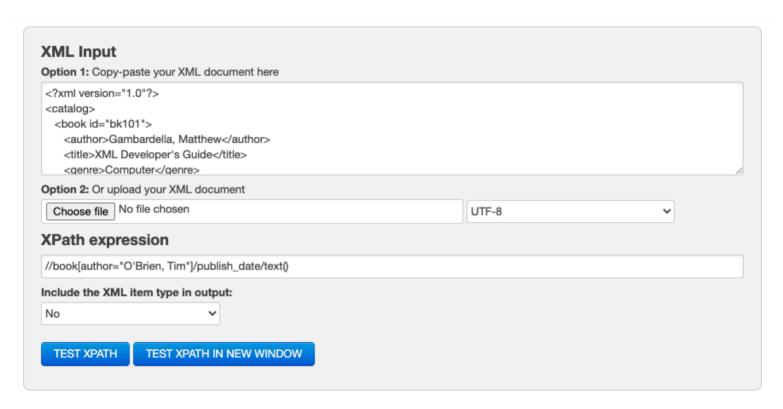
```
## [1] "2000-12-09" "2000-12-01"
```

XPath Query(relative) : //book[author="O'Brien, Tim"]/publish_date/text()

XPath Query(absolute) : /catalog/book[author="O'Brien, Tim"]/publish_date/text()

Output from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

XML Input

Option 1: Copy-paste your XML document here

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
```

Option 2: Or upload your XML document

| Choose file | No file chosen | UTF-8 |

XPath expression

//book[author="O'Brien, Tim"]/publish_date/text()

Include the XML item type in output:

No

TEST XPATH   TEST XPATH IN NEW WINDOW

XPath result:

```
2000-12-09
2000-12-01
```

Screenshot from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Query 3

List the title and prices of all books in the "Fantasy" genre.

```
#QUERY 3 - List the title and prices of all books in the "Fantasy" genre.

titleprice = xpathSApply(xmlDocument,'//book[genre = "Fantasy"]/title/text() | //book[genre = "Fantas
y"]/price/text()', xmlValue)
titleprice
```

```
## [1] "Midnight Rain"      "5.95"                "Maeve Ascendant"
## [4] "5.95"                "Oberon's Legacy"     "5.95"
## [7] "The Sundered Grail" "5.95"
```

XPath Query(relative) : //book[genre = "Fantasy"]/title/text() | //book[genre = "Fantasy"]/price/text()

XPath Query(absolute) : /catalog/book[genre = "Fantasy"]/title/text() | /catalog/book[genre = "Fantasy"]/price/text()

Output from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)
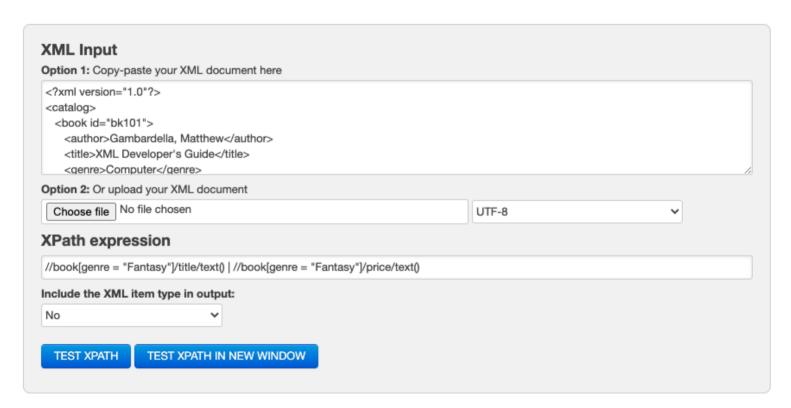

Screenshot from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Query 4

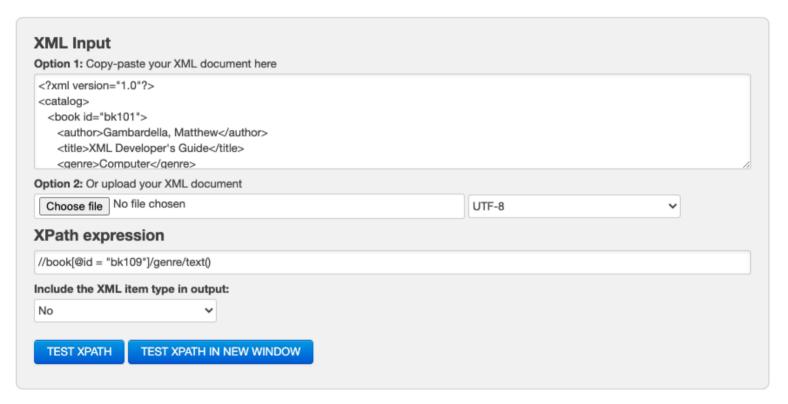In what genre is the book with id "bk109".

```
#QUERY 4 - In what genre is the book with id "bk109".

bookgenre = xpathSApply(xmlDocument,'//book[@id = "bk109"]/genre/text()', xmlValue)
bookgenre
```

```
## [1] "Science Fiction"
```

XPath Query(relative) : //book[@id = "bk109"]/genre/text()

XPath Query(absolute) : /catalog/book[@id = "bk109"]/genre/text()

Output from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Screenshot from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

Query 5

List the title and author of all books that cost less than $5.

```
#QUERY 5 - List the title and author of all books that cost less than $5.

lessthanfive = xpathSApply(xmlDocument,'//book[price < 5]/title/text() | //book[price < 5]/author/text
()', xmlValue)
lessthanfive
```
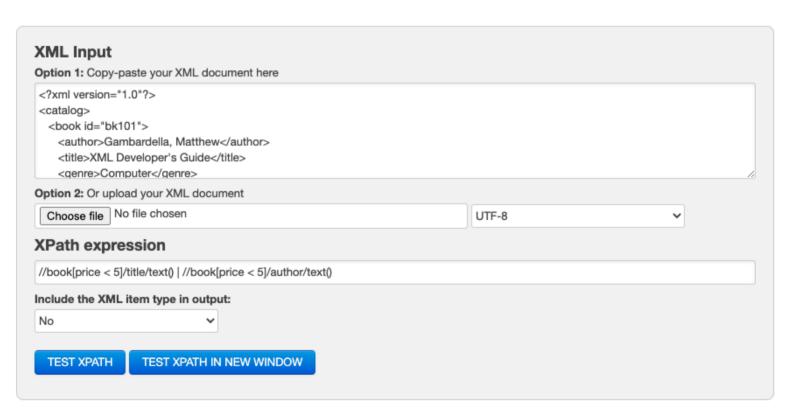
```
## [1] "Randall, Cynthia" "Lover Birds"       "Thurman, Paula"    "Splish Splash"
## [5] "Knorr, Stefan"     "Creepy Crawlies"
```

NOTE: we cannot list (title, author) in that sequence because union operator preserves the order of XML ELEMENTS and XPath has no sorting functionality, so our output instead shows like (Author, Title)

XPath Query(relative) : //book[price < 5]/title/text() | //book[price < 5]/author/text()

XPath Query(absolute) : /catalog/book[price < 5]/title/text() | //book[price < 5]/author/text()

Output from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)

XML Input

Option 1: Copy-paste your XML document here

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
```

Option 2: Or upload your XML document

| Choose file | No file chosen | | UTF-8 ⌄ |

XPath expression

```
//book[price < 5]/title/text() | //book[price < 5]/author/text()
```

Include the XML item type in output:

```
No ⌄
```

[TEST XPATH] [TEST XPATH IN NEW WINDOW]

XPath result:

```
Randall, Cynthia
Lover Birds
Thurman, Paula
Splish Splash
Knorr, Stefan
Creepy Crawlies
```

Screenshot from https://www.freeformatter.com/xpath-tester.html#ad-output (https://www.freeformatter.com/xpath-tester.html#ad-output)