

Activity1:

1)

The screenshot shows the 'Free Online XPath Tester / Validator' website. The left sidebar contains a menu with categories: Validators, XPath Tester, Encoders & Decoders, Code Minifiers / Beautifier, Converters, and Cryptography & Security. The main content area has a warning: 'your browser from being unresponsive.' Below this is the 'XML Input' section with two options: 'Option 1: Copy-paste your XML document here' and 'Option 2: Or upload your XML document'. The XML input field contains the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<authors>
  <author>
    <firstName>Danny</firstName>
    <lastName>Ayers</lastName>
  </author>

```

 The 'XPath expression' field contains '/authors/author[2]'. The 'Include the XML item type in output:' dropdown is set to 'Yes'. There are two buttons: 'TEST XPATH' and 'TEST XPATH IN NEW WINDOW'. The 'XPath result:' section shows the output:

```
Element='<author>
  <firstName>Joe</firstName>
  <lastName>Fawcett</lastName>
</author>'
```

This screenshot shows the same website after a second test. The 'XML Input' section remains the same. The 'XPath expression' field now contains '//author[2]'. The 'TEST XPATH' button has been clicked, and the 'XPath result:' section shows the output:

```
Element='<author>
  <firstName>Joe</firstName>
  <lastName>Fawcett</lastName>
</author>'
```

Free Online XPath Tester / Evaluate

freeformatter.com/xpath-tester.html#ad-output

FREEFORMATTER.COM Contact Like 1.9K

Validators

- JSON Validator
- HTML Validator
- XML Validator - XSD
- XPath Tester**
- Credit Card Number Generator & Validator
- Regular Expression Tester (RegEx)
- Java Regular Expression Tester (RegEx)
- Cron Expression Generator - Quartz

Encoders & Decoders

- Url Encoder & Decoder
- Base 64 Encoder & Decoder
- Convert File Encoding
- QR Code Generator

Code Minifiers / Beautifier

- JavaScript Beautifier
- CSS Beautifier
- JavaScript Minifier
- CSS Minifier

Converters

- XSD Generator
- XSLT (XSL Transformer)
- XML to JSON Converter
- JSON to XML Converter
- CSV to XML Converter
- CSV to JSON Converter
- Epoch Timestamp To Date

Cryptography & Security

- Message Digester (MD5, SHA-256, SHA-512)
- HMAC Generator
- MD5 Generator
- SHA-256 Generator

The maximum size limit for file upload is 2 megabytes. Results bigger than 500k will be written to a new window for performance reason and to prevent your browser from being unresponsive.

XML Input

Option 1: Copy-paste your XML document here

```
<?xml version="1.0" encoding="UTF-8"?>
<authors>
  <author>
    <firstName>William</firstName>
    <lastName>Shakespeare</lastName>
  </author>
</authors>
```

Option 2: Or upload your XML document

Choose File No file chosen UTF-8

XPath expression

//author[last()]

Include the XML item type in output:

Yes

TEST XPATH TEST XPATH IN NEW WINDOW

XPath result:

```
Element='<author>
  <firstName>William</firstName>
  <lastName>Shakespeare</lastName>
</author>'
```

2)

Free Online XPath Tester / Evaluate

freeformatter.com/xpath-tester.html#ad-output

FREEFORMATTER.COM Contact Like 1.9K

Validators

- SQL Formatter
- JSON Validator
- HTML Validator
- XML Validator - XSD
- XPath Tester**
- Credit Card Number Generator & Validator
- Regular Expression Tester (RegEx)
- Java Regular Expression Tester (RegEx)
- Cron Expression Generator - Quartz

Encoders & Decoders

- Url Encoder & Decoder
- Base 64 Encoder & Decoder
- Convert File Encoding
- QR Code Generator

Code Minifiers / Beautifier

- JavaScript Beautifier
- CSS Beautifier
- JavaScript Minifier
- CSS Minifier

Converters

- XSD Generator
- XSLT (XSL Transformer)
- XML to JSON Converter
- JSON to XML Converter
- CSV to XML Converter
- CSV to JSON Converter
- Epoch Timestamp To Date

Cryptography & Security

- Message Digester (MD5, SHA-256, SHA-512)
- HMAC Generator

XPath functions.

*The maximum size limit for file upload is 2 megabytes. Results bigger than 500k will be written to a new window for performance reason and to prevent your browser from being unresponsive.

XML Input

Option 1: Copy-paste your XML document here

```
<?xml version="1.0" encoding="UTF-8"?>
<items>
  <item id="07815">
    <description>2 Person Tent</description>
    <qty>1</qty>
  </item>
</items>
```

Option 2: Or upload your XML document

Choose File No file chosen UTF-8

XPath expression

//customer[1]/@cid

Include the XML item type in output:

Yes

TEST XPATH TEST XPATH IN NEW WINDOW

XPath result:

```
Attribute='cid=c5781'
```

Free Online XPath Tester / Evaluate XPath Expression

Freeformatter.com

XML Validator - XSD
XPath Tester
 Credit Card Number Generator & Validator
 Regular Expression Tester (RegEx)
 Java Regular Expression Tester (RegEx)
 Cron Expression Generator - Quartz

Encoders & Decoders
 Uri Encoder & Decoder
 Base 64 Encoder & Decoder
 Convert File Encoding
 QR Code Generator

Code Minifiers / Beautifier
 JavaScript Beautifier
 CSS Beautifier
 JavaScript Minifier
 CSS Minifier

Converters
 XSD Generator
 XSLT (XSL Transformer)
 XML to JSON Converter
 JSON to XML Converter
 CSV to XML Converter
 CSV to JSON Converter
 Epoch Timestamp To Date

Cryptography & Security
 Message Digests (MD5, SHA-256, SHA-512)
 HMAC Generator
 MD5 Generator
 SHA-256 Generator
 SHA-512 Generator

String Escaper & Utilities
 String Utilities
 HTML Escape
 XML Escape
 Java and .NET Escape
 JavaScript Escape
 JSON Escape
 CSV Escape
 SQL Escape

XML Input
 Option 1: Copy-paste your XML document here
 <?xml version="1.0" encoding="UTF-8" ?>
 <customers>
 <customer id="C5781">
 <name>Hardy, Joan</name>
 <street>105 Terrace Rd </street>
 </customer>
 </customers>
 Option 2: Or upload your XML document
 Choose File No file chosen UTF-8

XPath expression
 //orders/orderItem[qty=2]
 Include the XML item type in output:
 Yes

TEST XPATH TEST XPATH IN NEW WINDOW

XPath result:

```

Element="item id="p5148">
  <description>Self-Inflating Pad</description>
  <qty>2</qty>
</item>
Element="item id="ab2818">
  <description>Down Sleeping Bag</description>
  <qty>2</qty>
</item>
Element="item id="wb7133">
  <description>Insulated Water Bottle</description>
  <qty>2</qty>
</item>
Element="item id="br9002">
  <description>Bug Repellent (Deep Woods)</description>
  <qty>2</qty>

```

Activity:2

1)

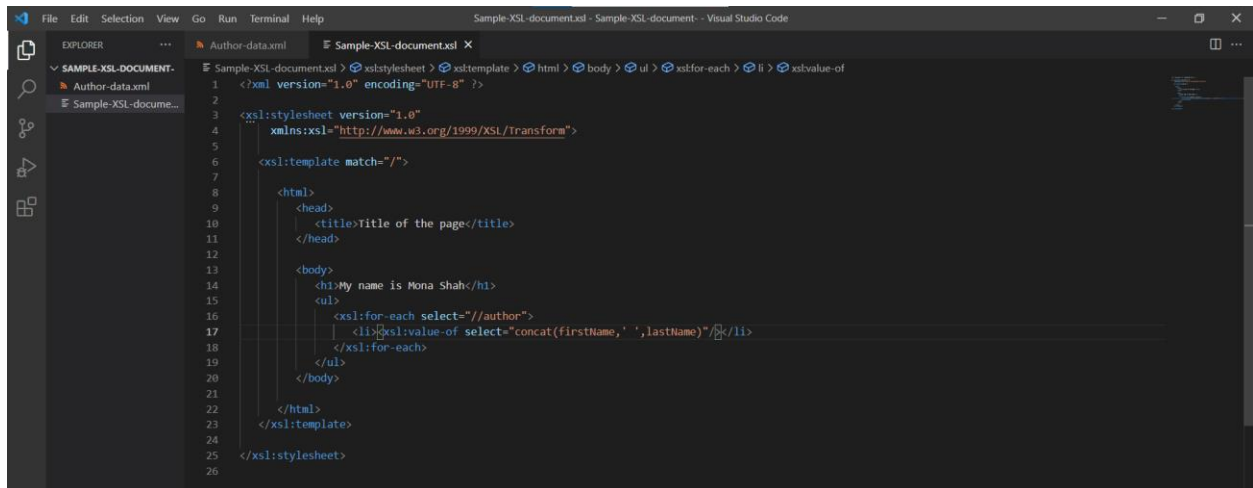


2)



My name is Mona Shah

- Danny Ayers
- Joe Fawcett
- William Shakespeare

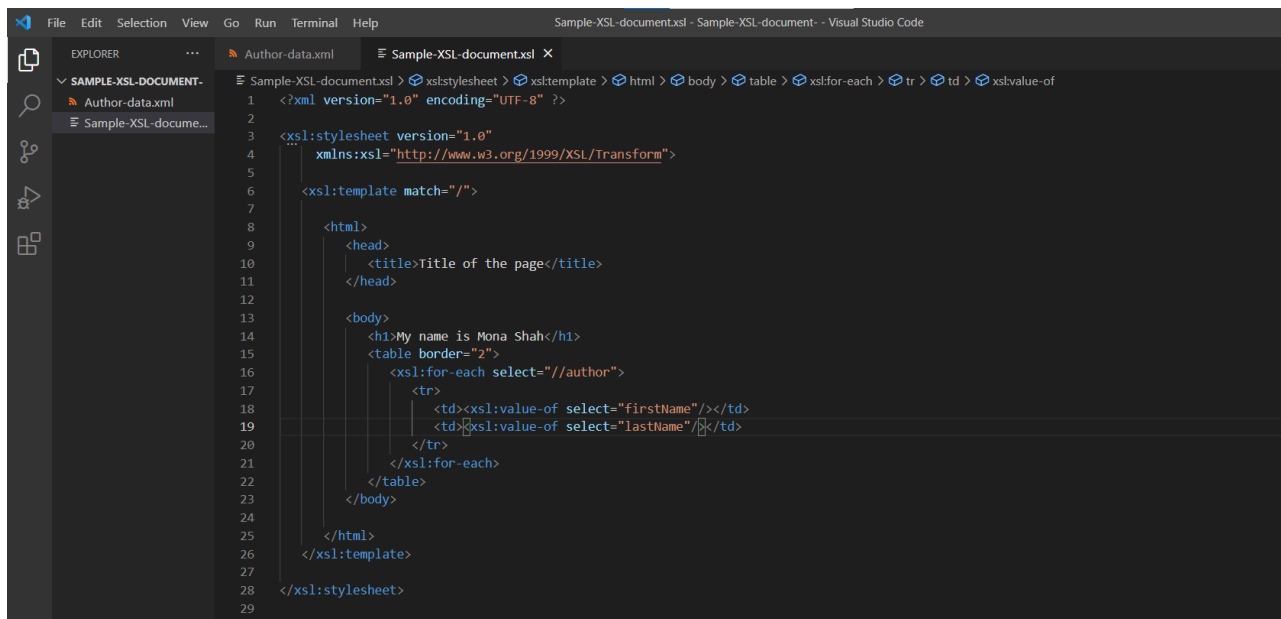


3)



My name is Mona Shah

Danny	Ayers
Joe	Fawcett
William	Shakespeare



Activity 3:



My name is Mona Shah

Name	City	State
Hardy, Joa	Duacna	NE
Evans, Terry	Mount Hope	OH

A screenshot of the Visual Studio Code editor. The Explorer pane on the left shows a project structure with files: Author-data.xml, Sample-XSL-document.xml, and Sample-Order-data.xml. The main editor displays the content of Sample-XSL-document.xml, which is an XSLT stylesheet. The code includes an XML declaration, an xsl:stylesheet declaration, an xsl:template match="/" and its body. The body contains an HTML document with a title 'Title of the page', a heading 'My name is Mona Shah', and a table. The table has a header row with columns 'Name', 'City', and 'State', and a data row generated by an xsl:for-each loop that iterates over the 'customers/customer' elements in the source XML. The xsl:for-each loop uses xsl:value-of elements to populate the table cells with the 'name', 'city', and 'state' attributes of each customer.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2
3 <xsl:stylesheet version="1.0"
4   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
5
6   <xsl:template match="/">
7
8     <html>
9       <head>
10        <title>Title of the page</title>
11      </head>
12
13      <body>
14        <h1>My name is Mona Shah</h1>
15        <table border="2">
16          <tr>
17            <td>Name</td>
18            <td>City</td>
19            <td>State</td>
20          </tr>
21          <xsl:for-each select="//customers/customer">
22            <tr>
23              <td><xsl:value-of select="name"/></td>
24              <td><xsl:value-of select="city"/></td>
25              <td><xsl:value-of select="state"/></td>
26            </tr>
27          </xsl:for-each>
28        </table>
29      </body>
30    </html>
31  </xsl:template>
32 </xsl:stylesheet>
33
34
35
```