Processing Web Data With XML And XSLT - Part 2

Bogdan Dumitru, Syncro Soft

bogdan_dumitru@sync.ro

Open4Tech Summer School, 2020

Note:

- GitHub repository: https://github.com/dumitrubogdanmihai/processing-web-data-with-xml-and-xslt
- Questions

Let's recap

In the first part we:

- saw how browsers render web pages
- talked about Selenium
- created a web crawler

Today's Agenda

- XML
- XPath
- XSLT
- Live Coding

About XML

eXtensible Markup Language

- It is a markup language
  - text is surrounded by tags (that provide semantics)
- doesn't define a set of elements

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <first-tag id="A">The Title</first-tag>
  <second-tag>
    <child/>
  </second-tag>
</root>
```

About XML - Syntax

XML Syntax Rules

- Prolog must be on the first line (if is present)
- Must have only one root element
- All start tags must have a closing tag
  - or to be self-closing tags
- Entities
  - `< > & &apos; &quot;`
- Comments
  - `<!-- TODO: fix it! -->`

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <first-tag id="A">The Title</first-tag>
  <second-tag>
    <child/>
  </second-tag>
</root>
```

About XML – Verification

Well-formed XML vs Valid XML

- Well-formed = conform the syntax rules
  - e.g: no missing end tags, no overlapping tags
- Valid = conform the schema rules
  - e.g: no more elements that the schema declares

XML Strong Points

- Semantics
  - data is wrapped in semantics
  - XML vocabularies
- Validation
  - controlled structure
- Reuse
  - data isn't duplicated
  - XInclude

Where it is used?

Wherever any of the following needs arise:

- semantic content
- content reuse
- well-structured content
- content validation

XML vs HTML vs XHTML

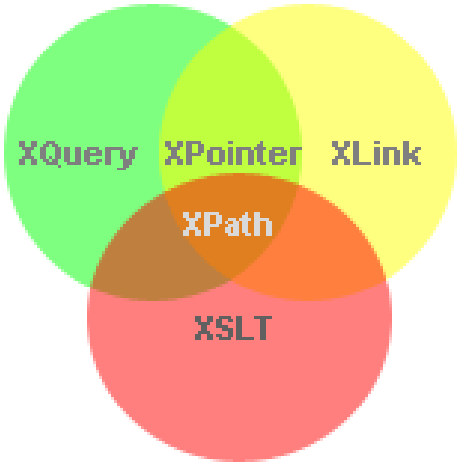
XML
- standard (specification) for describing structure and content
- extensible
- can explain what data means

- HTML (hypertext markup language)
  - non extensible (fixed tags set)
  - can't explain what data means
- XHTML
  - HTML that conform to XML standards (well formed HTML)

XML-related Technologies

The XML world is really big!

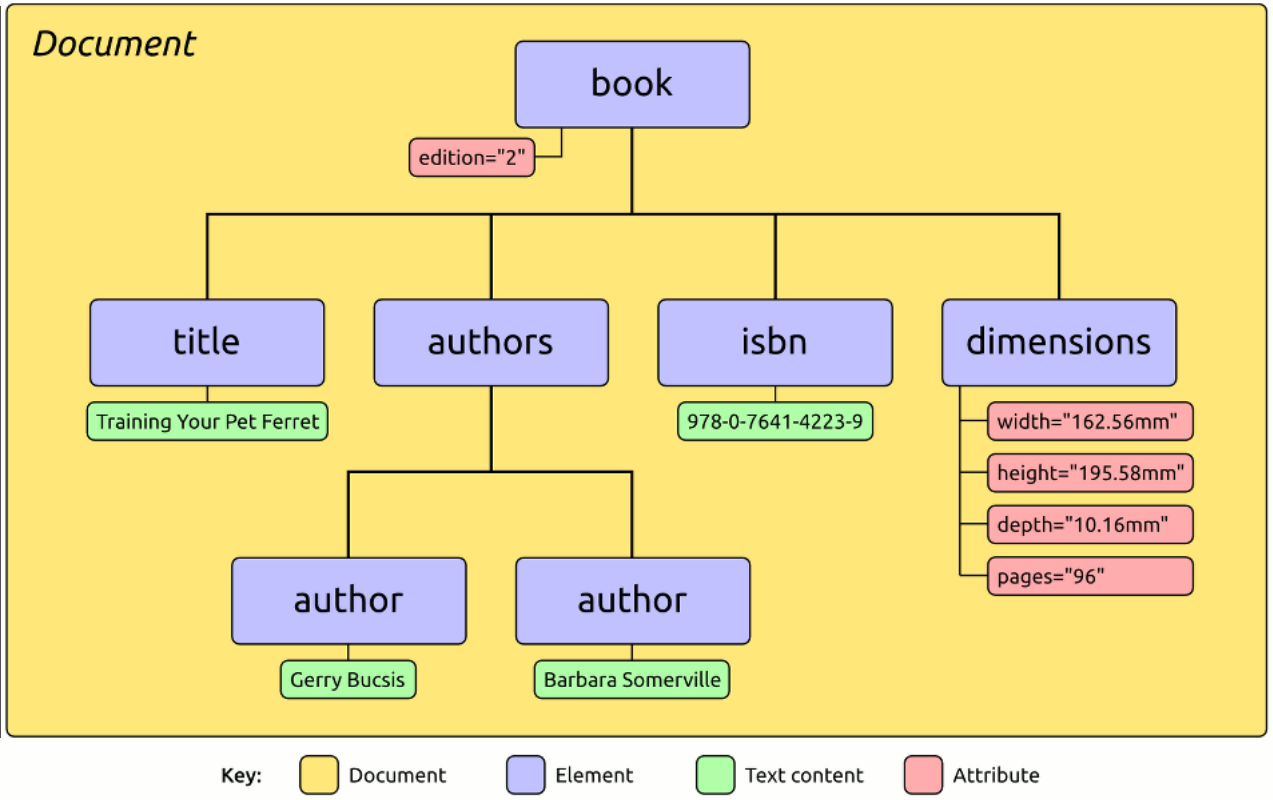
- XML
- XPath
- XSLT
- XQuery
- XSD, DTD
- SVG
- etc.



XML DOM

XML Document Object Model

```
<?xml version="1.0" encoding="UTF-8"?>
<book edition="2">
  <title>Training Your Pet Ferret</title>
  <authors>
    <author>Garry Bucsis</author>
    <author>Barbara Somerville</author>
  </authors>
  <isbn>978-0-7641-4233-9</isbn>
  <dimensions
    width="162.56mm"
    height="195.58mm"
    depth="10.16mm"
    pages="95"/>
</book>
```



XPath

XML Path Language

- **Select a set of nodes within an XML document**
- Highly used in XSLT
- Any CSS selector can be written in XPath

XPath – Syntax 1

- Basic syntax
  - ***/*** *- root element*
  - ***//div*** *- all div elements within document*
  - **/html/body/div** *- div elements within body*
  - **/html/body/../** *- body*
  - **/html/body/*** *- body children*
  - ***/@class** – the "class" attributes*
  - **.** *– the current element*

XPath – Syntax 2

- Axes
  - **//p/following-sibling::** - elements placed after *p*
  - **//p/preceding-sibling::** - elements placed before *p*
  - **//p/descendant::** - descendents of *p*

XPath – Syntax 3

- Operators
  - "|"
    - **//book | //magazine**
  - "="
    - **//book[@price=9.8]**
  - "or"
    - **//book[@price>=9.8 or @price<=10]**

XSLT

Extensible Stylesheet Language Transformations

- **Transform/remodel XML documents**

```
<?xml version="1.0" encoding="UTF-8"?>
<root title="The Main Title">
  <h1>The Header 1</h1>
  <section>
    <p>The first paragraph</p>
    <p>The second paragraph</p>
  </section>
</root>
```


```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <title>The Main Title</title>
   <h2>The Header 1</h2>
   <section>
      <p>The first paragraph</p>
      <p>The second paragraph</p>
   </section>
</root>
```

XSLT

Basic concepts

- <template>
  - <value-of>
  - context (.)
  - <copy>
- <apply-templates>

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">

  <xsl:template match="//h1">
    <h2>
      <xsl:apply-templates select="@*|node()"/>
    </h2>
  </xsl:template>

  <xsl:template match="//root/@title">
    <title>
      <xsl:value-of select="."/>
    </title>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:template>

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

XSLT

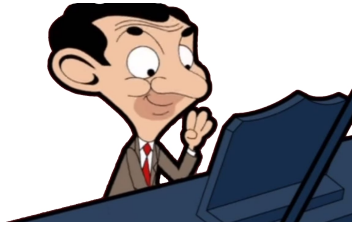
```xml
<?xml version="1.0" encoding="UTF-8"?>
<root title="The Main Title">
  <h1>The Header 1</h1>
  <section>
    <p>The first paragraph</p>
    <p>The second paragraph</p>
  </section>
</root>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">

  <xsl:template match="//h1">
    <h2>
      <xsl:apply-templates select="@*|node()"/>
    </h2>
  </xsl:template>

  <xsl:template match="//root/@title">
    <title>
      <xsl:value-of select="."/>
    </title>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:template>

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>
</xsl:stylesheet>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <title>The Main Title</title>
   <h2>The Header 1</h2>
   <section>
      <p>The first paragraph</p>
      <p>The second paragraph</p>
   </section>
</root>
```

Let's Code

- We'll extract extract useful data from the files generated from previous course.



THANK YOU!

Any questions?

Bogdan Dumitru

bogdan_dumitru@sync.ro