

# XML Semestr

Jakub Vondráček  
Daniel Šimánek  
Jiří Heller

# Content

---

- XML Structure
- Validation
  - DTD
  - RelaxNG
- HTML output
- PDF output
  - Advanced formatting



# XML Structure



# XML Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<areas>
  <area name="Greenland">
    <section title="Images">
      <content type="image-link"
        description="Country Map"
        href="https://www.cia.gov/the-world-factbook/static/3c3a8/GL-map.jpg"/>
    </section>
    <section title="Introduction">
      <subsection title="Background">
        <content>Greenland, the world's largest island, is about 80% ice-capped...</content>
      </subsection>
    </section>
  </area>
</areas>
```

# Tags

---

- **<areas>** - root element, holds information about all areas
- **<area>** - element that represents single area
  - **name** - area name
- **<section>** - element that holds information about certain topic
  - **title** - section title
- **<subsection>** - the same as section, used to make hierarchy
- **<content>** - surrounds text, images, URIs...
  - **type** - formatting information
    - **image-link** - uri to an image
    - **strong** - bold text formatting
    - **comparison-link** - uri to an external site
    - **audio-link** - uri to audio file
  - **description** - in *image-link* only, provides description for image
  - **href** - in *image-link*, *external-link*, contains uri path to an external source

# Validation



# DTD Validation



```
<!ELEMENT area {section+}>
<!ELEMENT section ((subsection+ | (content+))>
<!ELEMENT subsection (content+)>
<!ELEMENT content (#PCDATA)>
```

```
<!ATTLIST area name CDATA #REQUIRED>
<!ATTLIST section title CDATA #REQUIRED>
<!ATTLIST subsection title CDATA #REQUIRED>
<!ATTLIST content type ( strong | image-link | external-link | comparison-link | audio-link ) #IMPLIED
|
|
|
|
description CDATA #IMPLIED
href CDATA #IMPLIED>
```

# DTD Validation



- areas are validated area by area
- validation for xml file with all areas

```
<!ENTITY % simplifiedtd SYSTEM "area.dtd">
```

```
<!ELEMENT areas (area+)>
```

```
%simplifiedtd;
```



# RelaxNG Validation

- simplified as much as possible
- validation in external library
- validation for file with all areas, every single area

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">  
    
  <include href="area-elements.rng"/>  
    
  <start>  
    <ref name="area" />  
  </start>  
    
</grammar>
```

# RelaxNG Library

```
<define name="section">
  <element name="section">
    <attribute name="title">
      <text/>
    </attribute>
    <choice>
      <oneOrMore>
        <ref name="subsection"/>
      </oneOrMore>
      <oneOrMore>
        <ref name="content"/>
      </oneOrMore>
    </choice>
  </element>
</define>
```

```
<define name="content">
  <choice>
    <element name="content">
      <optional>
        <attribute name="type">
          <choice>
            <value>strong</value>
            <value>comparison-link</value>
          </choice>
        </attribute>
      </optional>
      <text/>
    </element>
    <element name="content">
      <attribute name="type">
        <value>image-link</value>
      </attribute>
      <attribute name="description">
        <text/>
      </attribute>
      <ref name="href"/>
      <text/>
    </element>
  </choice>
</define>
```

# RelaxNG Advanced Validation



```
<define name="href">
  <attribute name="href">
    <data type="anyURI" datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
      <param name="pattern">https?://.*</param>
    </data>
  </attribute>
</define>
```

# HTML output



# HTML Generation: Index

```
<!-- HTML file generation of index.html -->
<xsl:template match="areas">
  <xsl:result-document href="output/html/index.html">
    <xsl:element name="html">
      <xsl:element name="head">
        <xsl:element name="link">
          <xsl:attribute name="rel">stylesheet</xsl:attribute>
          <xsl:attribute name="href">./style-index.css</xsl:attribute>
        </xsl:element>
      </xsl:element>
      <xsl:element name="body">
        <xsl:element name="h1">Facts about six countries</xsl:element>
        <xsl:element name="h2">from the CIA's World Factbook</xsl:element>
        <xsl:element name="ul">
          <xsl:apply-templates select="area" mode="link"/>
        </xsl:element>
      </xsl:element>
    </xsl:element>
  </xsl:result-document>
  <xsl:apply-templates select="area" mode="detail"/>
</xsl:template>
```

`<xsl:stylesheet version='2.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>`

`<xsl:output indent="yes" method="html" />`

`<xsl:template match="/">`  
`<xsl:apply-templates select="areas"/>`  
`</xsl:template>`

# HTML: Index: Link to country

<!-- Link to the HTML page for a specific country -->

```
<xsl:template match="area" mode="link">
  <xsl:element name="li">
    <xsl:element name="a">
      <xsl:attribute name="href">
        <xsl:value-of select="concat(translate(@name, '()', ''), '.html' )"/>
      </xsl:attribute>
      <xsl:value-of select="@name"/>
      <xsl:text>&#8239;&#8239;&#8239;&#10095;</xsl:text>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

# Facts about six countries

*from the CIA's World Factbook*

[Greenland ›](#)

[Iceland ›](#)

[Saint Kitts and Nevis ›](#)

[Saint Vincent and the Grenadines ›](#)

[Italy ›](#)

[Holy See \(Vatican City\) ›](#)

“index.html”

# HTML: Country: Head

<!-- HTML file generation for every single country -->

<xsl:template match="area" mode="detail">

  <xsl:result-document href="{concat('output/html/', translate(@name, '()', ''), '.html')}">

    <xsl:element name="html">

      <xsl:element name="head">

        <xsl:element name="link">

          <xsl:attribute name="rel">stylesheet</xsl:attribute>

          <xsl:attribute name="href">./style-area.css</xsl:attribute>

        </xsl:element>

      </xsl:element>



# HTML: Country: Body

```
<xsl:element name="body">
  <xsl:element name="h1">
    <xsl:value-of select="@name"/>
  </xsl:element>
  <xsl:element name="a">
    <xsl:attribute name="class">back-button</xsl:attribute>
    <xsl:attribute name="href">index.html</xsl:attribute>
    <xsl:text>&#10094;&#8239;&#8239;&#8239;&#8239;Back to main page</xsl:text>
  </xsl:element>
  <xsl:element name="div">
    <xsl:element name="table">
      <xsl:attribute name="class">images</xsl:attribute>
      <xsl:element name="tr">
        <xsl:apply-templates select="./section[@title='Images']/content[@type='image-link']"/>
      </xsl:element>
    </xsl:element>
  </xsl:element>
  <xsl:element name="hr"/>
  <xsl:element name="h2">
    <xsl:text>Table of contents</xsl:text>
  </xsl:element>
  <xsl:element name="ul">
    <xsl:apply-templates select="./section[not(@title='Images')]" mode="link"/>
  </xsl:element>
  <xsl:apply-templates select="./section[not(@title='Images')]" mode="detail"/>
</xsl:element>
</xsl:element>
</xsl:result-document>
</xsl:template>
```

# HTML: Images

```
<!-- Image with description -->
<xsl:template match="content[@type='image-link']">
  <xsl:element name="td">
    <xsl:element name="p">
      <xsl:element name="i">
        <xsl:value-of select="@description"/>
      </xsl:element>
    </xsl:element>
    <xsl:element name="img">
      <xsl:attribute name="src">
        <xsl:value-of select="@href"/>
      </xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

# HTML: Links to sections

<!-- Link to section -->

```
<xsl:template match="section[not(@title='Images')]" mode="link">
```

```
  <xsl:element name="li">
```

```
    <xsl:element name="a">
```

```
      <xsl:attribute name="class">button-section</xsl:attribute>
```

```
      <xsl:attribute name="href">
```

```
        <xsl:text>#</xsl:text>
```

```
        <xsl:value-of select="generate-id(.)"/>
```

```
      </xsl:attribute>
```

```
      <xsl:value-of select="translate(@title, ' ', '&#160;')"/>
```

```
      <xsl:text>&#8239;&#8239;&#8239;&#10095;</xsl:text>
```

```
    </xsl:element>
```

```
  </xsl:element>
```

```
</xsl:template>
```

# HTML: Sections & Subsections

```
<!-- Section title and contents -->
<xsl:template match="section[not(@title='Images')]" mode="detail">
  <xsl:element name="hr">
    <xsl:attribute name="id">
      <xsl:value-of select="generate-id(.)"/>
    </xsl:attribute>
  </xsl:element>
  <xsl:element name="h2">
    <xsl:value-of select="@title"/>
  </xsl:element>
  <xsl:apply-templates select="subsection"/>
</xsl:template>

<!-- Subsection title and contents -->
<xsl:template match="subsection">
  <xsl:element name="h3">
    <xsl:value-of select="@title"/>
  </xsl:element>
  <xsl:element name="div">
    <xsl:apply-templates select="./content"/>
  </xsl:element>
</xsl:template>
```

# HTML: Content without type



```
<!-- Basic content in a paragraph -->  
<xsl:template match="content[not(@type)]">  
  <xsl:element name="p">  
    <xsl:value-of select="."/>  
  </xsl:element>  
</xsl:template>
```

# HTML: Strong&Comparison Content

<!-- Strong content -->

```
<xsl:template match="content[@type='strong']">
```

```
  <xsl:element name="h4">
```

```
    <xsl:value-of select="concat(upper-case(substring(.,1,1)), substring(., 2))"/>
```

```
  </xsl:element>
```

```
</xsl:template>
```

<!-- Comparison content -->

```
<xsl:template match="content[@type='comparison-link']">
```

```
  <xsl:element name="p">
```

```
    <xsl:attribute name="class">comparison</xsl:attribute>
```

```
    <xsl:element name="i">
```

```
      <xsl:value-of select="."/><xsl:text>.</xsl:text>
```

```
    </xsl:element>
```

```
  </xsl:element>
```

```
</xsl:template>
```

# HTML: Ext. links, Audio

<!-- External link content -->

```
<xsl:template match="content[@type='external-link']">
```

```
  <xsl:element name="p">
```

```
    <xsl:element name="a">
```

```
      <xsl:attribute name="href">
```

```
        <xsl:value-of select="@href"/>
```

```
      </xsl:attribute>
```

```
      <xsl:value-of select="."/>
```

```
    </xsl:element>
```

```
  </xsl:element>
```

```
</xsl:template>
```

<!-- Audio content -->

```
<xsl:template match="content[@type='audio-link']">
```

```
  <xsl:element name="audio">
```

```
    <xsl:attribute name="controls"/>
```

```
    <xsl:element name="source">
```

```
      <xsl:attribute name="src">
```

```
        <xsl:value-of select="@href"/>
```

```
      </xsl:attribute>
```

```
      <xsl:attribute name="type">audio/mpeg</xsl:attribute>
```

```
    </xsl:element>
```

```
  </xsl:element>
```

```
</xsl:template>
```

```
<xsl:template match="*" />
```

# PDF Output





# XSL-FO: Two page-masters



```
<fo:simple-page-master master-name="Cover" page-height="29.7cm" page-width="21.0cm" margin="1cm 2cm 1cm 2cm">
  <fo:region-body margin-top="2cm" margin-bottom="1.5cm"/>
</fo:simple-page-master>
```

```
<fo:simple-page-master master-name="Pages" page-height="29.7cm" page-width="21.0cm" margin="1cm 2cm 1cm 2cm">
  <fo:region-body margin-top="2cm" margin-bottom="1.5cm"/>
  <fo:region-before extent="2cm" display-align="before"/>
  <fo:region-after extent="1.5cm" display-align="after"/>
</fo:simple-page-master>
```

```
<fo:page-sequence-master master-name="PageSequence">
  <fo:repeating-page-master-alternatives>
    <fo:conditional-page-master-reference page-position="first" master-reference="Cover"/>
    <fo:conditional-page-master-reference page-position="rest" master-reference="Pages"/>
  </fo:repeating-page-master-alternatives>
</fo:page-sequence-master>
```

# XSL-FO: Before and after regions

```
<fo:page-sequence master-reference="PageSequence">
```

```
  <fo:static-content flow-name="xsl-region-before">
```

```
    <fo:block text-align="center" color="gray">
```

```
      <fo:retrieve-marker retrieve-class-name="area"/>
```

```
    </fo:block>
```

```
  </fo:static-content>
```

```
  <fo:static-content flow-name="xsl-region-after">
```

```
    <fo:block text-align="center" color="gray">
```

```
      <fo:page-number/> / <fo:page-number-citation ref-id="terminator"/>
```

```
    </fo:block>
```

```
  </fo:static-content>
```

```
  <fo:flow flow-name="xsl-region-body" font-family="Times">
```

```
    <xsl:apply-templates select="areas"/>
```

```
  </fo:flow>
```

```
</fo:page-sequence>
```

```
<fo:retrieve-marker retrieve-class-name="section"/>
```

```
<!-- Section title and contents -->
```

```
<xsl:template match="section[not(@title='Images'>
```

```
  <fo:block id="{generate-id(.)}" space-before="0">
```

```
    <fo:marker marker-class-name="section">
```

```
      <xsl:value-of select="@title"/>
```

```
    </fo:marker>
```

```
<fo:block id="terminator"/>
```

61 / 146

Saint Kitts and Nevis - Environment

# XSL-FO: Advanced text formatting

```
<fo:block id="{generate-id(.)}" font-size="28pt" font-weight="bold" font-family="Helvetica"
| | | text-transform="uppercase" text-align="center" letter-spacing="2px" page-break-before="always">
```

**HOLY SEE (VATICAN CITY)**

Communications

**Telephones - fixed lines**

**Total subscriptions**

12,483 (2020)

**Subscriptions per 100 inhabitants**

11.25 (2020 est.)

*(country comparison to the world: 186.)*

```
<!-- Strong content -->
<xsl:template match="content[@type='strong']">
  <fo:block font-size="12pt" font-weight="bold" space-before="8px" keep-with-next.within-column="always">
    <xsl:value-of select="concat(translate(substring(.,1,1), 'abcdefghijklmnopqrstuvwxyz', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'), substring(., 2))"/>
  </fo:block>
</xsl:template>
```

# XSL-FO: Tables of contents

```
<fo:list-block provisional-distance-between-starts="40px" provisional-label-  
| <xsl:apply-templates select="section" mode="link"/>  
</fo:list-block>
```

```
<!-- Link to a section -->
```

```
<xsl:template match="section[not(@title='Images')]" mode="link">
```

```
  <fo:list-item space-after="5px">
```

```
    <fo:list-item-label end-indent="label-end()">
```

```
      <fo:block>
```

```
        <fo:page-number-citation ref-id="{generate-id()}" />
```

```
      </fo:block>
```

```
    </fo:list-item-label>
```

```
    <fo:list-item-body start-indent="body-start()">
```

```
      <fo:block font-size="12px">
```

```
        <fo:basic-link internal-destination="{generate-id()}" color="blue" text-decoration="underline">
```

```
          <xsl:value-of select="@title"/>
```

```
        </fo:basic-link>
```

```
      </fo:block>
```

```
    </fo:list-item-body>
```

105	<a href="#">Geography</a>
105	<a href="#">People and Society</a>
111	<a href="#">Environment</a>
113	<a href="#">Government</a>
119	<a href="#">Economy</a>
124	<a href="#">Energy</a>
126	<a href="#">Communications</a>
128	<a href="#">Transportation</a>

```
<!-- Section title and contents -->
```

```
<xsl:template match="section[not(@ti
```

```
  <fo:block id="{generate-id()}"
```

# XSL-FO: Images

<!-- Table cell with an image and its description -->

```
<xsl:template match="content[@type='image-link']">
```

```
  <fo:table-cell>
```

```
    <fo:block font-size="12px" font-style="italic" text-align="center" space-after="5px">
```

```
      <xsl:value-of select="@description"/>
```

```
    </fo:block>
```

```
    <fo:block text-align="center">
```

```
      <fo:external-graphic src="url({@href})" content-height="90px"/>
```

```
    </fo:block>
```

```
  </fo:table-cell>
```

```
</xsl:template>
```

*Country Flag*



*Country Map*



*Locator Map*



**END**