Modelling humanities data with TEI-XML

SCHOLARLY EDITING AND MANUSCRIPT CATALOGUING IN THE DIGITAL AGE

Dr Katarzyna Anna Kapitan
10 October 2024

Recap: XML

- XML is an international non-proprietary standard, which is widely used to export, share, and store structured data.
- XML is expressed in plain text, so it's hardware and software independent.

Recap: XML

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>This is a paragraph</p>
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<myRoot>
  <myContent>
    <content>
      Here is my content
    </content>
  </myContent>
</myRoot>
```

Structure of a class in XML

```
<?xml version="1.0" encoding="UTF-8"?>
    <workshop name="XML_workshop”>
       <instructors>
          <person>
             Katarzyna Kapitan
          </person >
        </instructors>
       <participants>
          <person>
             John Doe
          </person>
          <person>
                Anna Smith
          </person>
          <person>
             Jan Kowalski
          </person>
       </participants>
     </workshop>
```

Workshop:

Instructors:

Katarzyna Kapitan

Participants:

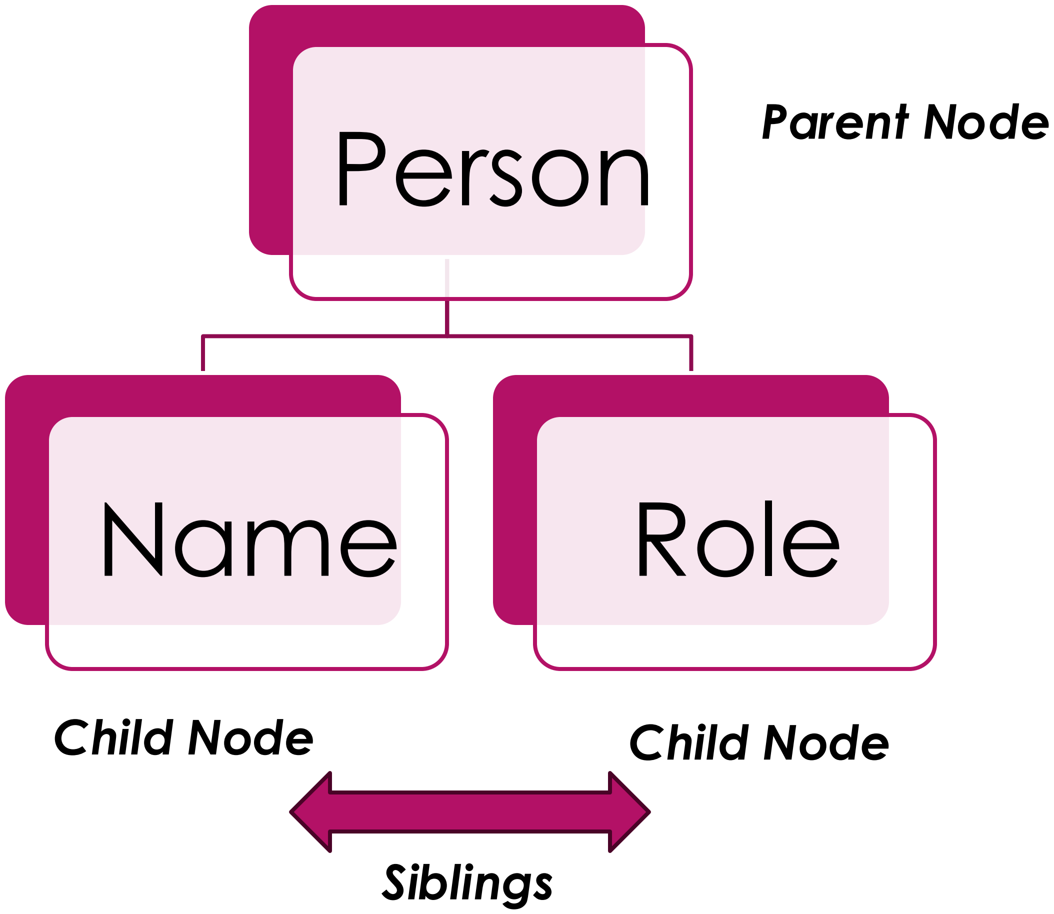
John Doe

Anna Smith

Jan Kowalski

Element Nesting

```
<person>
   <name>
         Katarzyna Kapitan
   </name>
   <role>
         instructor
   </role>
</person>
```



Person

Parent Node

Name

Role

Child Node

Child Node

Siblings

Exercise 1: XML & HTML

https://github.com/KAKDH/TNAH_XML2025/tree/main/Week2/Exercises

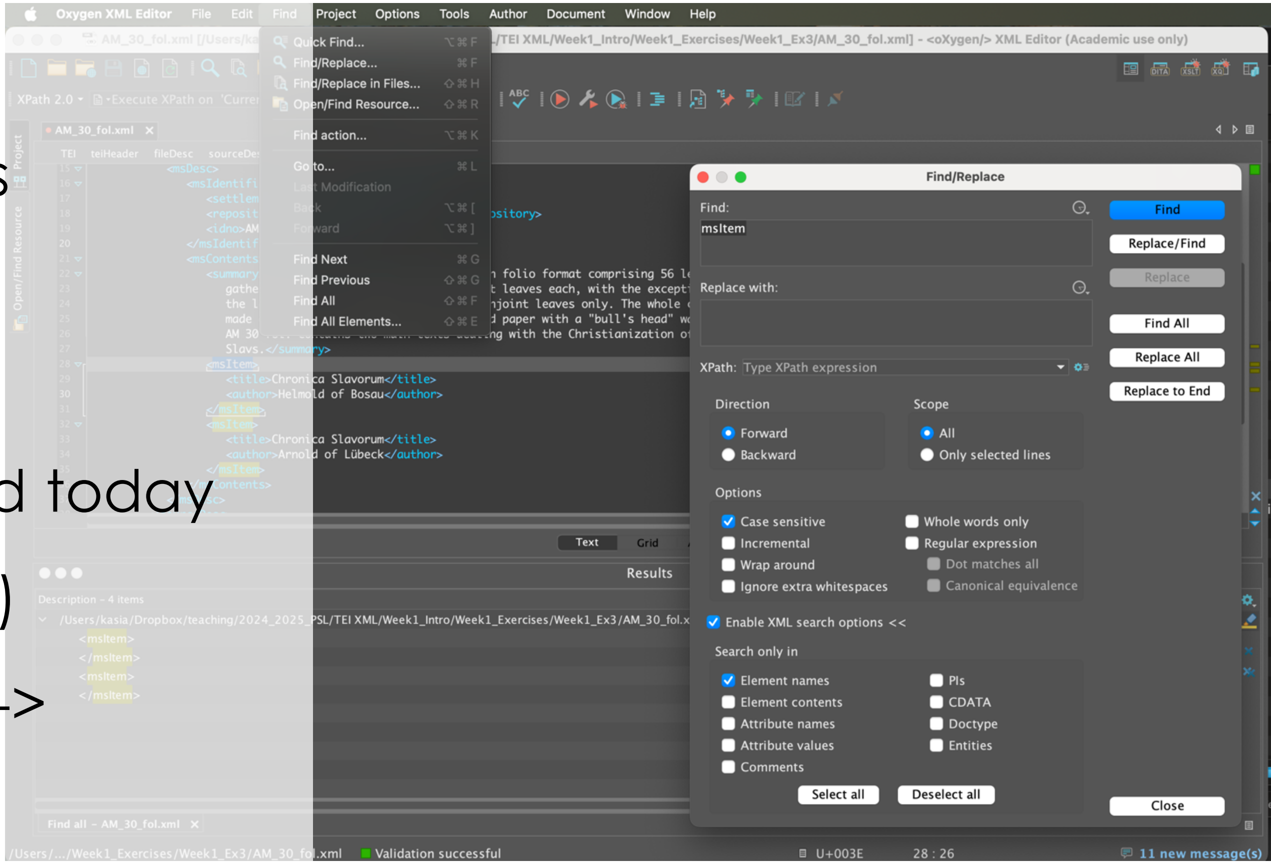
Task 1: Analyse the structure of an XML document using Oxygen XML Editor (Window -> Show View -> Outline)

- Which element is the parent element of msDesc?
- Which elements are siblings of the parent element of msDesc?
- Which elements are children elements of msDesc?

Task 2 : Generate an HTML file from XML with XSLT; follow the guidelines in Guidelines_Transformation_Scenario.pdf

- Compare the XML and HTML files.
- Analyse what information is lost in the HTML version, **find three examples**.

- XML encoding allows you to easily retrieve all the information **you chose to encode**, for example:
    - How many texts are preserved in this manuscript (encoded within **msItem** elements)
    - In which library this manuscript is held today (encoded within **repository** element)
- In Oxygen XML Editor you can use Find -> Find/Replace to find elements



```
<msItem>
   <title>Chronica Slavorum</title>
   <author>Helmold of Bosau</author>
</msItem>
<msItem>
   <title>Chronica Slavorum</title>
   <author>Arnold of Lübeck</author>
</msItem>
```

Find/Replace

Find: msItem

Replace with:

XPath: Type XPath expression

Direction: ● Forward ○ Backward

Scope: ● All ○ Only selected lines

Options: ☑ Case sensitive ☐ Incremental ☐ Wrap around ☐ Ignore extra whitespaces ☐ Whole words only ☐ Regular expression ☐ Dot matches all ☐ Canonical equivalence

☑ Enable XML search options <<

Search only in: ☑ Element names ☐ Element contents ☐ Attribute names ☐ Attribute values ☐ Comments ☐ PIs ☐ CDATA ☐ Doctype ☐ Entities

Results: Description - 4 items

Encoding choices: Attribute or Element

Example 1:

```
<person role=”instructor">

        Katarzyna Kapitan

</person>
```

Example 2:

```
<person>
    <name>
            Katarzyna Kapitan
    </name>
    <role>
             instructor
    </role>
</person>
```

DTD (Document Type Definition)

DTD (Document Type Definition)

- A **document type definition** (DTD) is a specification that defines the valid building blocks of an XML document.
- A DTD defines the document structure with a list of validated elements and attributes.
- A DTD can be
  - declared inline inside an XML document,
  - or as an external reference (DTD specification file)
- The DTD specification file can be used to validate XML documents.

More: https://en.wikipedia.org/wiki/Document_type_definition

DTD (Document Type Definition)

- **DTDs** describe the structure of a class of documents via
  - **Element declarations** (describing elements and their relationship)
  - **Attribute-list declarations** (describing attributes and their values)

More: https://en.wikipedia.org/wiki/Document_type_definition

Element Declarations

- **Element Declarations**
  - list the elements which are allowed within the document
  - specify whether and how declared elements may be nested (contained within each element)

More: https://en.wikipedia.org/wiki/Document_type_definition

Element Declarations

- <!ELEMENT **ElementName** **ElementSpec**>
- **Specification** of the **Element** can have different values, for example
  - **EMPTY**: for specifying that the defined element allows no content.
  - **ANY**: for specifying that the defined element allows any content.
  - an **expression** in brackets (), specifying the only elements allowed as direct children in the content of the defined element, including:
    - **#PCDATA**: parsed character data for specifying that the defined element allows textual content.

Element Declarations: Example

- **DTD**: <!ELEMENT lb EMPTY>
    - Element Name: lb
    - Element Specification: EMPTY
- **XML:** <lb/>

Element Declarations: Example

- **DTD:** <!ELEMENT title (#PCDATA)>
  - Element Name: title
  - Element Specification:
    Contains #PCDATA (i.e. textual content)
- **XML:** <title> My title </title>

Element Declarations

- **Element Specification** within Element Declaration can define how nested elements relate to each other by using sequence list or choice list:
  - **Sequence list** – a list of one or more content particles. It is specified within parentheses and separated by a comma. **All the content particles must** appear successively as direct children in the content of the defined element.
  - **DTD:** <!ELEMENT publication **(title, author, date)**>
  - **Choice list** – a mutually exclusive list of two or more content particles. It is list specified within parentheses and separated by a pipe. **Only one of these content particles may** appear in the content of the defined element at the same position.
  - **DTD:** <!ELEMENT publication **(title | author |date)**>

Element Declarations

- **Element Specification** can contain **Quantifiers:**
  - + for specifying that there must be one or more occurrences of the item; one or more
  - * for specifying that any number of occurrences is allowed (the item is optional); zero or more
  - ? for specifying that there must **not** be more than one occurrence (the item is optional); zero or one
- **DTD:** <!ELEMENT publication (title, **author+**, date, **publicationPlace?**)>

Element Declarations: Example

- **DTD:** <!ELEMENT publication (title, author, date)>
- **XML:**

```
<publication>
        <title></title>
        <author></author>
        <date></date>
</publication>
```

The elements included in the declaration of another element need their own declaration.

Attribute-list declaration

- **Attribute-list declarations**
  - list the attributes which are allowed for each declared element
  - specify the type of each attribute value, and/or an explicit set of valid values

Attribute-list declaration

- **<!ATTLIST ElementName AttributeName DataType Value>**
- An attribute list specifies the list of all possible attributes associated with the element type.
- For each possible attribute, it contains:
  - the declared name of the attribute,
  - its data type (or a list of its possible values),
  - its default value (or usage)

Attribute-list declaration

- **Model: <!ATTLIST ElementName AttributeName DataType Value**>
- The most common values for **DataType** are:
  - **CDATA (characters data)** – value of the attribute can be any textual value.
  - **ID (identifier)** – value of the attribute must be a valid identifier. It is used to define the current element.
  - **IDREF** (reference to an identifier) – value of the attribute must be a valid identifier and must be referencing the unique element with an ID.
  - a defined list of values within parenthesis.

Attribute-list declaration

- **Model: <!ATTLIST ElementName AttributeName DataType Value**>
- The most common values for **Value** are:
  - *value* – the default value of the attribute
  - #REQUIRED – the attribute is required
  - #IMPLIED the attribute is optional;
  - #FIXED the attribute has a fixed value

Attribute-list declaration: Example

- **Model: <!ATTLIST ElementName AttributeName DataType Value>**
- **DTD: <!ATTLIST date when CDATA #REQUIRED>**
  - ElementName: date
  - AttributeName: when
  - DataType: CDATA
  - Value/Usage: Required
- **XML**: <date when="2025-10-10"/>

Attribute-list declaration: Example

DTD:

```
     <!ATTLIST date
    when CDATA #REQUIRED
    calendar (Gregorian | Chinese) #IMPLIED>
```

XML:

```
     <date when="2025-10-10" calendar="Gregorian"/>
     <date when="2025-10-10"/>
```

Exercise 2: Internal DTD

https://github.com/KAKDH/TNAH_XML2025/tree/main/Week2/Exercises

- Open the XML file bibliography_dtd_internal.xml in Oxygen
- Add a new element publisherName as a child of the publication element.
- Make sure it required, i.e. there must be only one publisherName element per publication.
- Create a closed list of attributes for the types of publications, the values of the attribute should be *book, book chapter, journal article*, make the attribute required.
- Adjust the encoding of your bibliography accordingly.

DOCTYPE (Document Type Declaration)

DTD & DOCTYPE

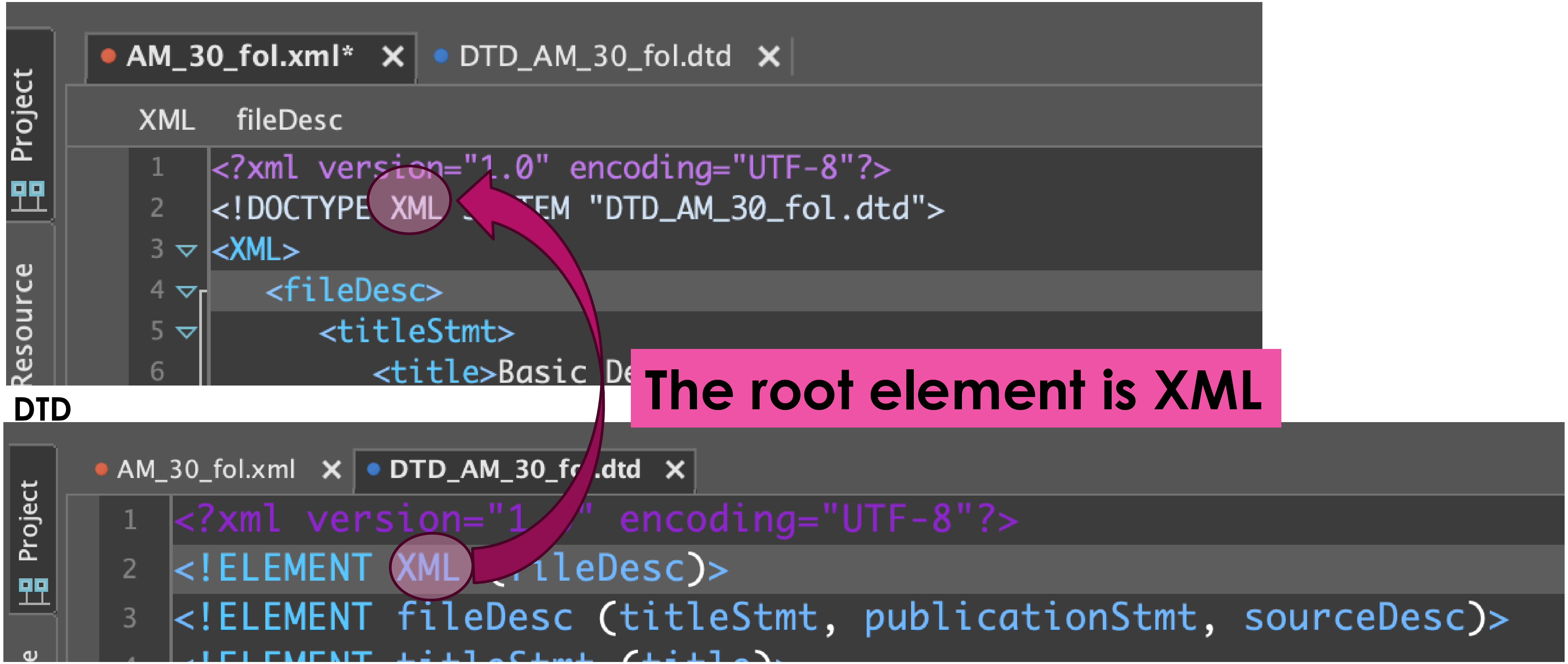
- A DTD is associated with an XML document by means of a **document type declaration (DOCTYPE)**.
- The DOCTYPE appears in near the start of an XML document.
- The declaration establishes that the document is an instance of the type defined by the referenced DTD.

DOCTYPE

- DOCTYPEs make two sorts of declarations:
  - an optional internal subset
    - **<!DOCTYPE RootElement [** *<!-- internal subset declarations -->* **]>**
  - an optional external subset:
    - **<!DOCTYPE RootElement SYSTEM** "myDtdFile.dtd"**>**
    - **<!DOCTYPE RootElement PUBLIC** "/quotedFPI/" "/quotedURI/" **>**

- **<!DOCTYPE** RootElementOfYourDTDFile **SYSTEM** " NameOfYourDTDFile.dtd **">**
- Document Type Declaration in an XML file referring to an external DTD, which is stored locally on your computer (in the same folder as your XML file).

XML



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE XML SYSTEM "DTD_AM_30_fol.dtd">
<XML>
    <fileDesc>
        <titleStmt>
            <title>Basic De
```

The root element is XML

DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT XML (fileDesc)>
<!ELEMENT fileDesc (titleStmt, publicationStmt, sourceDesc)>
<!ELEMENT titleStmt (title)>
```

Exercise 3: External DTD

https://github.com/KAKDH/TNAH_XML2025/tree/main/Week2/Exercises

- Open the XML file bibliography_dtd_external.xml in Oxygen
- Associate the DTD file bibliography_dtd_external.dtd with your XML file to validate, follow the model:

```
<!DOCTYPE RootElementOfYourDTDFile SYSTEM
"NameOfYourDTDFile.dtd">
```

- Add a new element publisherName as a child the publication, make it optional, but restrict its use to max one occurrence.

Homework

- Using the files from Exercise 3:
- Encode one more publication to your XML file, the details of the publication are in the comment at the bottom of the file.
- Make all the changes in your DTD that are necessary for you to be able to encode the second example (journal-specific info).
- Make sure your XML validates correctly.
- Send both files to Katarzyna by email (before 23:59 Tuesday 14/10):
  - katarzyna.kapitan [at] chartes.psl.eu

Encoding Project Portfolio: Part 1

- Instructions: https://github.com/KAKDH/TNAH_XML2025/tree/main/Encoding_Project_Portfolio_Instructions
- Due date: 20 October 2025
- Submission: Link to GitHub repo submitted through Moodle.

Useful links to explore (in addition to the reading list)

- XML DTD, *w3schools*:
  - https://www.w3schools.com/xml/xml_dtd_intro.asp
- Document type definition, *Wikipedia*:
  - https://en.wikipedia.org/wiki/Document_type_definition
- Document type declaration, *Wikipedia*:
  - https://en.wikipedia.org/wiki/Document_type_declaration