FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Semi-structured Data

2 - XML

Andreas Pieris and Wolfgang Fischl, Summer Term 2016

Outline

- **XML at First Glance:**
  - The Benefits of XML
  - XML vs. HTML
  - What XML Is Not
  - How XML Works
  - The Evolution of XML

- **XML Fundamentals:**
  - Elements and Tags
  - Character Data
  - XML Trees
  - Attributes
  - XML Names
  - Character Reference
  - Comments
  - Processing Instructions
  - XML Declaration
  - Well-formed XML Documents

XML at First Glance

- eXtensible Markup Language
- W3C standard for document markup since 1998
- Generic syntax to markup data with human- and machine-readable tags

```
<person>
    <name>
        <first> Andreas </first>
        <last> Pieris </last>
    </name>
    <tel> 740072 </tel>
    <fax> 18493 </fax>
    <email> pieris@dbai.tuwien.ac.at </email>
</person>
```

The Benefits of XML

- Structural and semantic markup language - the markup describes the structure and the semantics of the document

```
<person>
        <name>
              <first> Andreas </first>
              <last> Pieris </last>
        </name>
        <tel> 740072 </tel>
        <fax> 18493 </fax>
        <email> pieris@dbai.tuwien.ac.at </email>
</person>
```

e.g., first and last are associated with name, while Andreas is a first name and Pieris is a last name

ATTENTION: XML is not a presentation language (like HTML)

The Benefits of XML

- Definition of application-specific document types - supports interoperability and extensibility

e.g., real estate domain

```
<house>
    <address>
        <street> Bräuhausgasse </street>
        <number> 49 </number>
        <postcode> A-1050 </postcode>
        <city> Vienna </city>
    </address>
    <rooms> 3 </rooms>
</house>
```

The Benefits of XML

- XML documents are plain text - offers platform-independent data formats (portable data)

- Suitable for storing and exchanging any data that can be encoded as text

ATTENTION: XML is unsuitable for digitized data (photos, sound, etc.)

XML vs. HTML

Superficially, the markup in XML looks like the markup in HTML

… but there are some crucial differences

XML	HTML
Structural and semantic language	Presentation language
No fixed set of elements that are supposed to work in every domain	Fixed set of elements with predefined semantics
Extensible - can be extended to meet different needs	Not extensible - it does web pages, but nothing else

XML vs. HTML

An HTML document - tags with predefined meaning

```
<html>
    <head>
        <title> This is an example </title>
    </head>
    <body>
        <p> Hello World! </p>
    </body>
</html>
```

<html> defines the whole document

<head> contains meta data that are not displayed

<body> describes the visible page content

<p> defines a paragraph

What XML Is Not

- Programing language - there is no XML compiler that reads XML files and produces executable code

- Network protocol - data sent across a network might be encoded in XML, but there is a protocol that actually sends the XML document

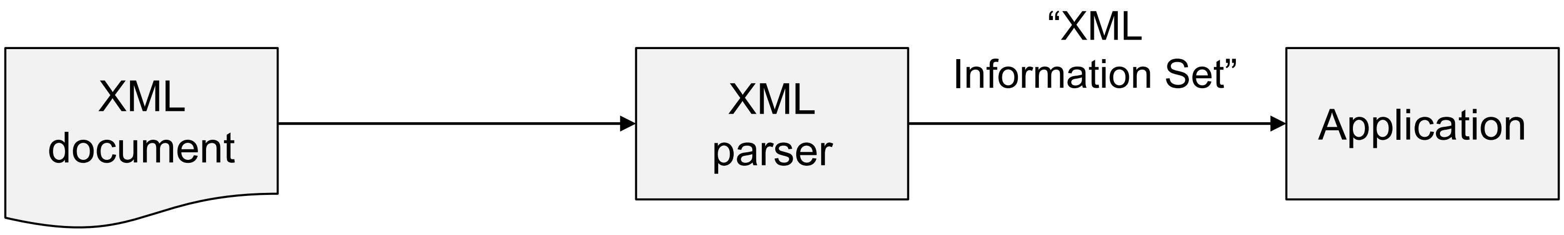
- Database - a database may contain XML data, but the database itself is not an XML document

ATTENTION: XML documents simply exist - they do nothing

How XML Works

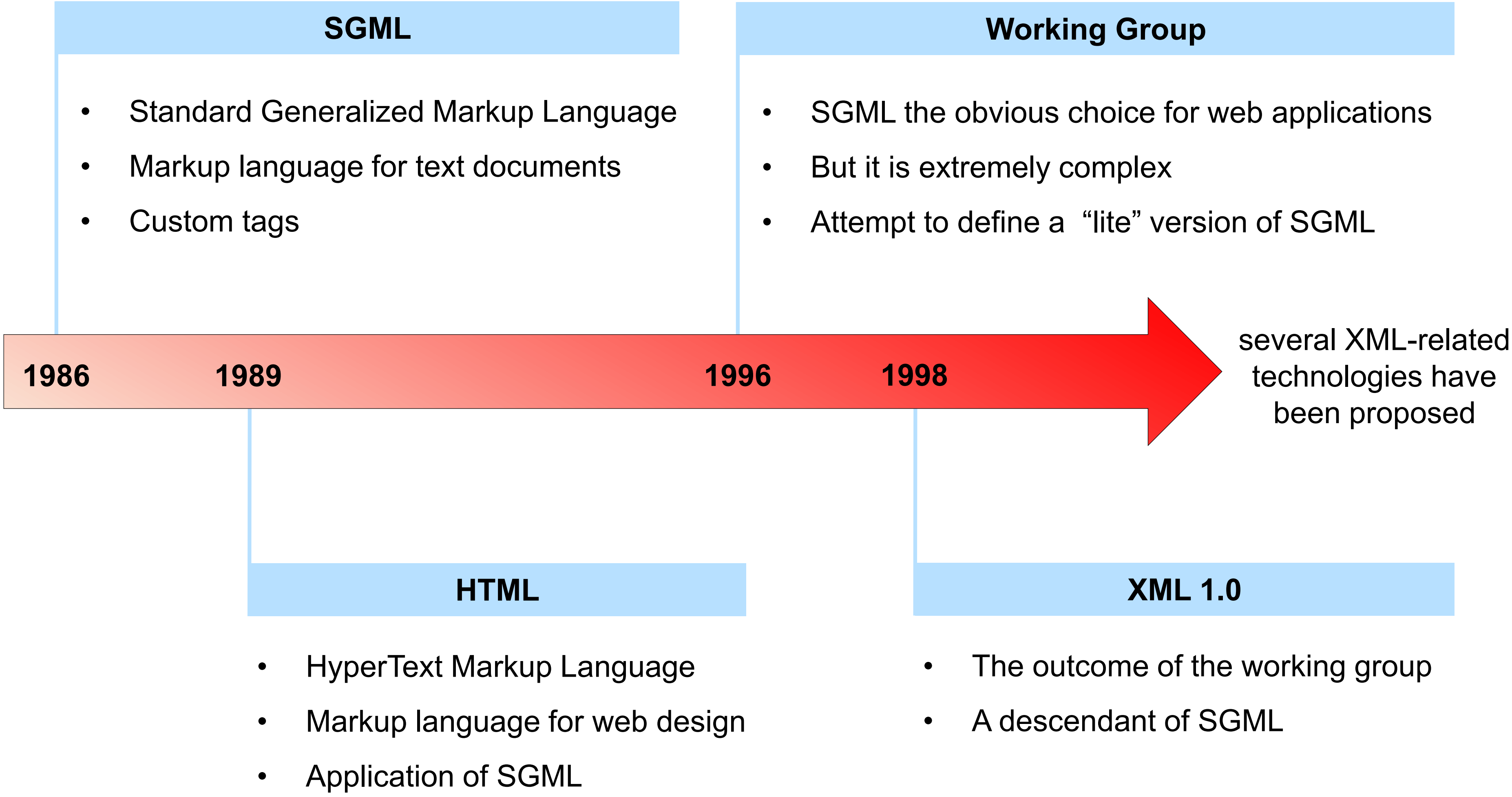
- Strict rules regarding the syntax of XML documents - allows for the development of XML parsers that can read documents

- Applications that need to understand an XML document will use a parser



Splits the document into individual pieces

The Evolution of XML



1986

SGML

- Standard Generalized Markup Language
- Markup language for text documents
- Custom tags

1989

HTML

- HyperText Markup Language
- Markup language for web design
- Application of SGML

1996

Working Group

- SGML the obvious choice for web applications
- But it is extremely complex
- Attempt to define a “lite” version of SGML

1998

XML 1.0

- The outcome of the working group
- A descendant of SGML

several XML-related technologies have been proposed

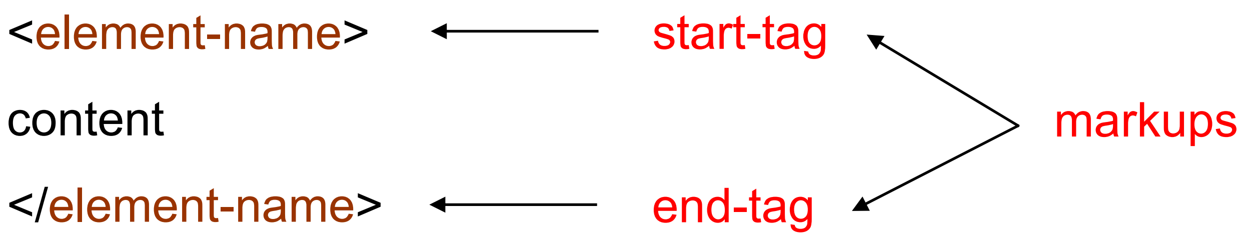
Outline

- **XML at First Glance:**
  - The Benefits of XML
  - XML vs. HTML
  - What XML Is Not
  - How XML Works
  - The Evolution of XML

- **XML Fundamentals:**
  - Elements and Tags
  - Character Data
  - XML Trees
  - Attributes
  - XML Names
  - Character Reference
  - Comments
  - Processing Instructions
  - XML Declaration
  - Well-formed XML Documents

Elements and Tags

- Element - the main concept of XML documents



<element-name> ← start-tag

content

</element-name> ← end-tag

markups

- The content can be
  - Empty - an empty element is abbreviated as <element-name/>
  - Simple content - consists of text
  - Element content - consists of one or more elements
  - Mixed content - consists of text and elements

ATTENTION: XML is case sensitive - <course> and <COURSE> are different

Character Data

```
<course>
    Semi-structured Data (SSD)   ←   character data
</course>
```

- Markup represent the structure of the document
- Character data represents the remaining information
- Both are stored as plain text

XML Trees

```
<course year="2015" semester="Summer">
    <title> Semi-structured Data (SSD) </title>
    <details>
        <day> Thursday </day>
        <time> 09:15 </time>
        <location> HS8 </location>
    </details>
    <classes>
        <class date="March 5">
            <subject> Introduction </subject>
            <subject> XML </subject>
        </class>
        …
    </classes>
</course>
```

root element

child elements of details

child elements of first

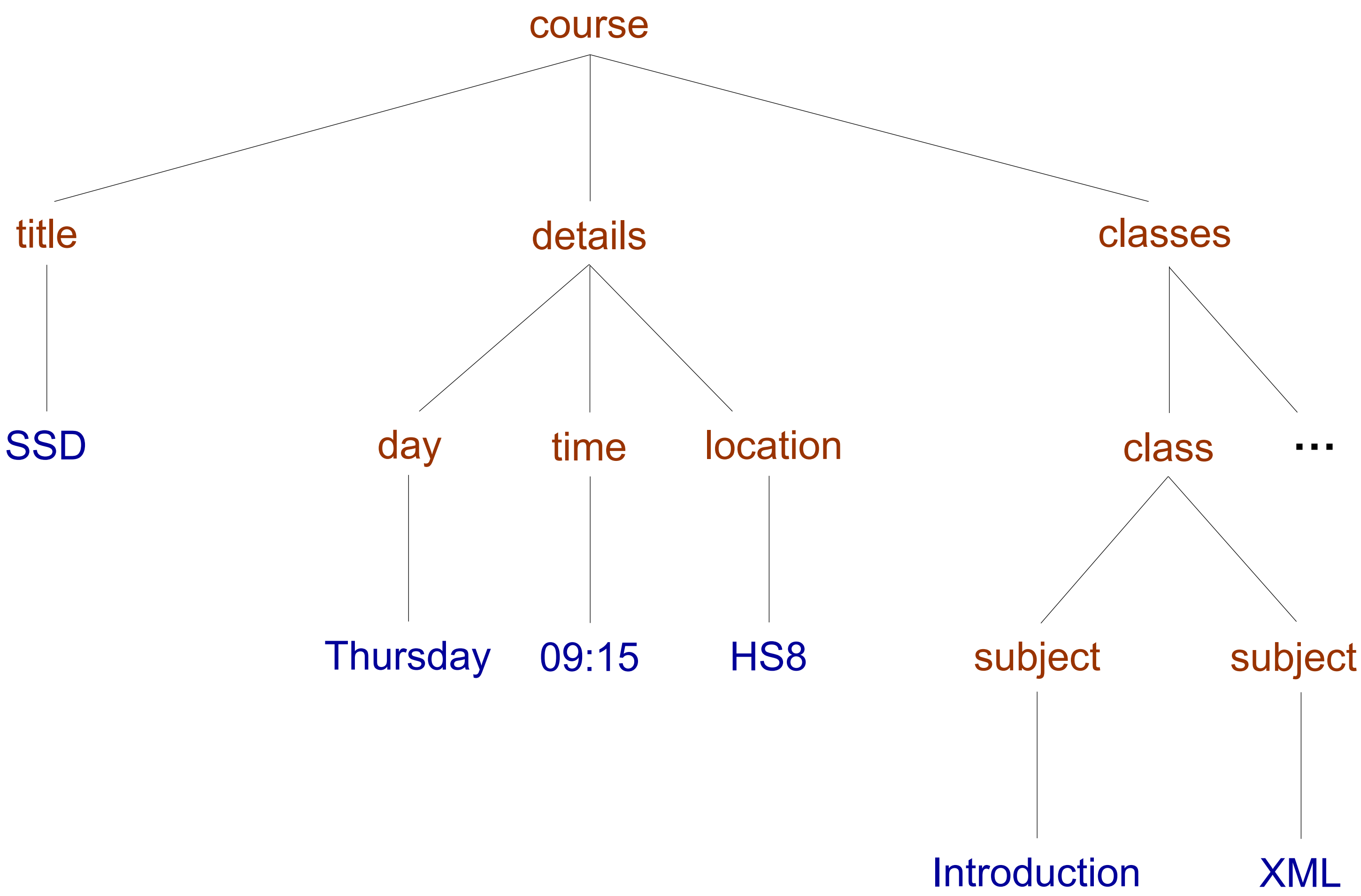
XML Trees

- An element may have several child elements
- An element (apart from the root) has exactly on parent element
- An element is completely enclosed by another element - overlapping tags are not allowed

```
<course>
    <title>
        Semi-structured Data
    </title>
</course>            ✓
```

```
<course>
    <title>
        Semi-structured Data
    </course>
</title>             ✗
```

XML Trees


course
title
details
classes
SSD
day
time
location
class
...
Thursday
09:15
HS8
subject
subject
Introduction
XML

Attributes

- We have already seen attributes in XML documents - for example,

```
<course year=“2015” semester=“Summer”>
    <title> Semi-structured Data </title>
</course>
```

- Specify properties of an element
- A name-value pair attached to the element’s start-tag

Attributes

- Elements with attributes have the following form:

  <element-name attr-name$_1$=“value$_1$” … attr-name$_n$=“value$_n$”>

  content

  </element-name>

  for each $i \neq j$, attr-name$_i$ $\neq$ attr-name$_j$

- The order of attributes is not significant
- attr-name$_i$=“value$_i$” & attr-name$_i$ = ‘value$_i$’ are the same

```
<course year=“2015” semester=“Summer”>
    <title> Semi-structured Data </title>
</course>
```

```
<course semester = ‘Summer’ year = ‘2015’>
    <title> Semi-structured Data </title>
</course>
```

XML Names

- But, what can be used as XML names?

- XML names are:
  - Element names
  - Attribute names
  - Names for other constructs (later)

- May contain:
  - Alphanumeric characters (A-Z, a-z, 0-9)
  - Non-English letters (δ, ü, ß, ж, etc.)
  - Numbers
  - Underscore (_), hyphen (-), period (.)

- May not contain:
  - Punctuation other than underscore (_), hyphen (-), period (.)
  - Whitespace of any kind

XML Names

ATTENTION:

- Names beginning with “XML” (in any combination of case) are forbidden
- XML names may only start with letters and underscore
- There is no limit to the length of an XML name
- Colon (:) is allowed, but its use is reserved for namespaces (later)

<course> ... </course>

<first_name> ... </first_name>

<_1st-class> ... </_1st-class>

✓

<xml_course> ... </ xml_course >

<first name> ... </first name>

<1st-class> ... </1st-class>

✗

Character References

- The character data inside an element may not contain the symbol <

```
<less-than>
        1 < 2
</less-than>
```

→

```
<less-than>
        1 &lt; 2
</less-than>
```

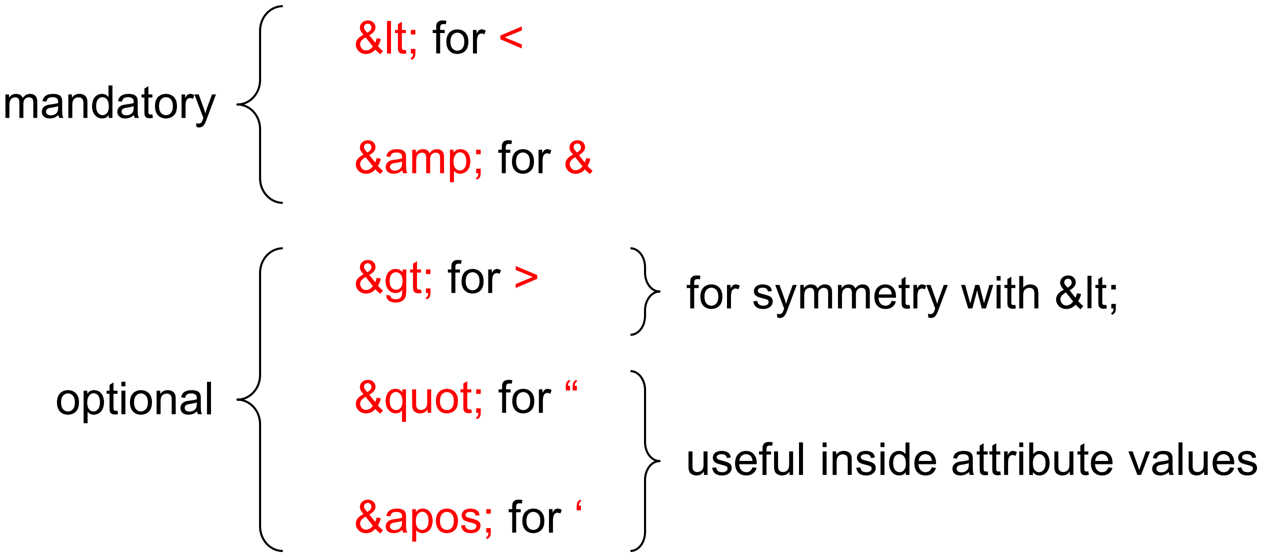
- &lt; is called entity reference

- But now the symbol ampersand (&) is problematic

- Use the entity reference &amp; instead of &

Character References

- XML predefines five entity references:



mandatory

&lt; for <

&amp; for &

optional

&gt; for >

&quot; for “

&apos; for ‘

for symmetry with &lt;

useful inside attribute values

- Additional references can be defined in the document type definition (later)

ATTENTION: Entity references cannot be used in XML names

Comments

- XML documents can be commented as follows:

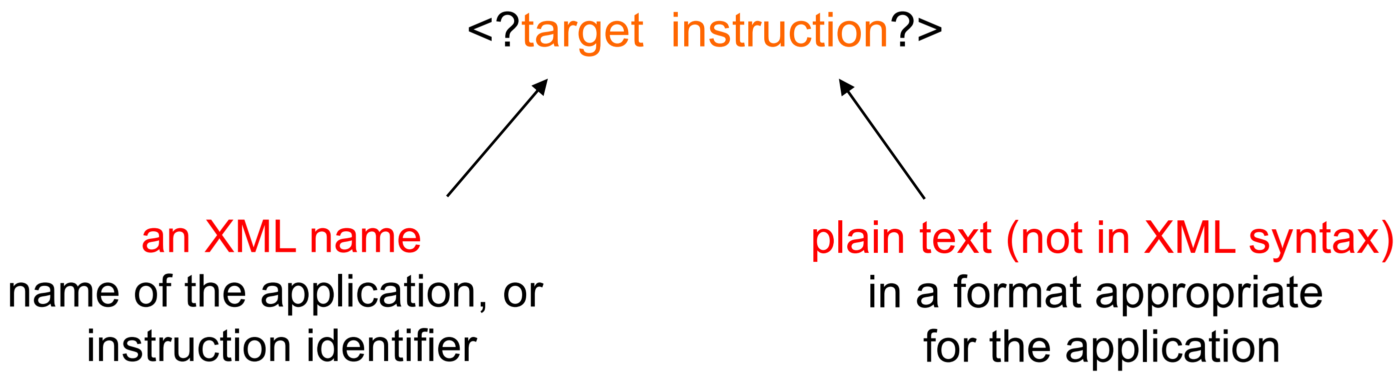
```
<!-- Here is my comment -->
```

- Double-hyphen (--) must not appear inside the comment
- Comments may appear anywhere outside tags and other comments
- XML parsers are free to completely ignore comments

ATTENTION: Comments are not elements

Processing Instructions

- A way of passing information to applications



<?target instruction?>

an XML name
name of the application, or
instruction identifier

plain text (not in XML syntax)
in a format appropriate
for the application

- May appear anywhere outside tags

ATTENTION: Processing instructions are not elements

Processing Instructions: Example

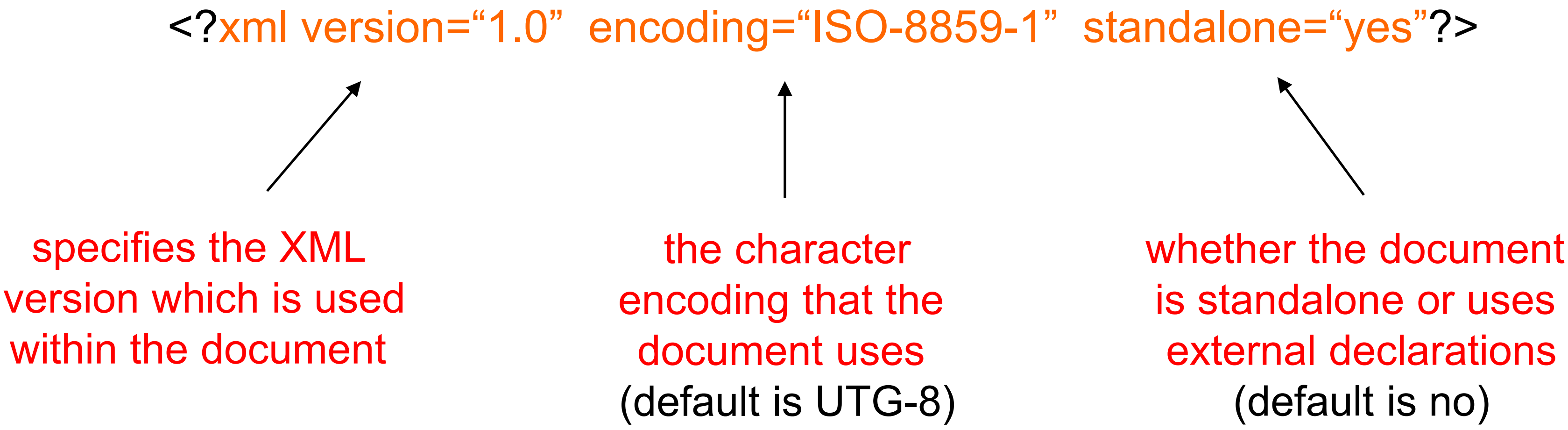
```
<?xml-stylesheet  href=“course.css”  type=“text/css”?>
```

Attach stylesheets to XML documents

http://www.w3schools.com/xml/xml_display.asp

XML Declaration

- XML should begin (but is optional) with an XML declaration:



```
<?xml version="1.0"  encoding="ISO-8859-1"  standalone="yes"?>
```

specifies the XML version which is used within the document

the character encoding that the document uses (default is UTG-8)

whether the document is standalone or uses external declarations (default is no)

- The XML declaration must be the first thing in the document

ATTENTION: XML declaration is not an element or processing instruction

Well-formed XML Documents

- Every XML document must be well-formed - no exception
- It must adhere to some rules including:
  - Every start-tag has a matching end-tag
  - Elements may nest but not overlap
  - Exactly one root element
  - Attribute values are quoted
  - Attribute names in an element are unique
  - Comments and processing instruction not inside tags
  - No < or & inside the data character of an element or attribute
  - …

ATTENTION: Before publishing an XML document, check it for well-formedness

Check for Well-formedness

```
<course year=“2015” semester=“Summer”>
    <title> SSD </title>
    <details>
        <day> Thursday </day>
        <time> 09:15 </time>
        <location> HS8 </location>
    </details>
    <classes>
        <class date=“March 5”>
            <subject> Introduction </subject>
            <subject> XML </subject>
        </class>
    </classes>
</course>
```

```
<course year=“2015” semester=“Summer”>
    <title> SSD </title>
    <details>
        <day> Thursday </day>
        <time> 09:15 </time>
        <location> HS8 </location>
    </details
    <classes>
        <class date=“March 5”>
            <subject> Introduction </subject>
            <subject> XML </subject>
        </class>
    </classes>
</course>
```

A Complete XML Document

```
<?xml version="1.0" encoding="UTF-8“ standalone=“yes”?>
<?xml-stylesheet href="course_style.css" type="text/css"?>
<!-- DBAI -->
<course year=“2015” semester=“Summer”>
    <title> Semi-structured Data (SSD) </title>
    <details>
        <day> Thursday </day>
        <time> 09:15 </time>
        <location> HS8 </location>
    </details>
    <classes>
        <class date=“March 5”>
            <subject> Introduction to the Module &amp; Course </subject>
            <subject> Introduction to SSD </subject>
            <subject> XML </subject>
        </class>
        …
    </classes>
</course>
```

… available at the webpage of the course