

# Semi-structured Data

## 2 - XML

# 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> Wolfgang </first>
    <last> Dvořák </last>
  </name>
  <tel> 18441 </tel>
  <fax> 918441 </fax>
  <email> dvorak@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> Wolfgang </first>
    <last> Dvořák </last>
  </name>
  <tel> 18441 </tel>
  <fax> 918441 </fax>
  <email> dvorak@dbai.tuwien.ac.at </email>
</person>
```

e.g., **first** and **last** are associated with **name**, while **Wolfgang** is a first name and **Dvořák** 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> Bräuhausegasse </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



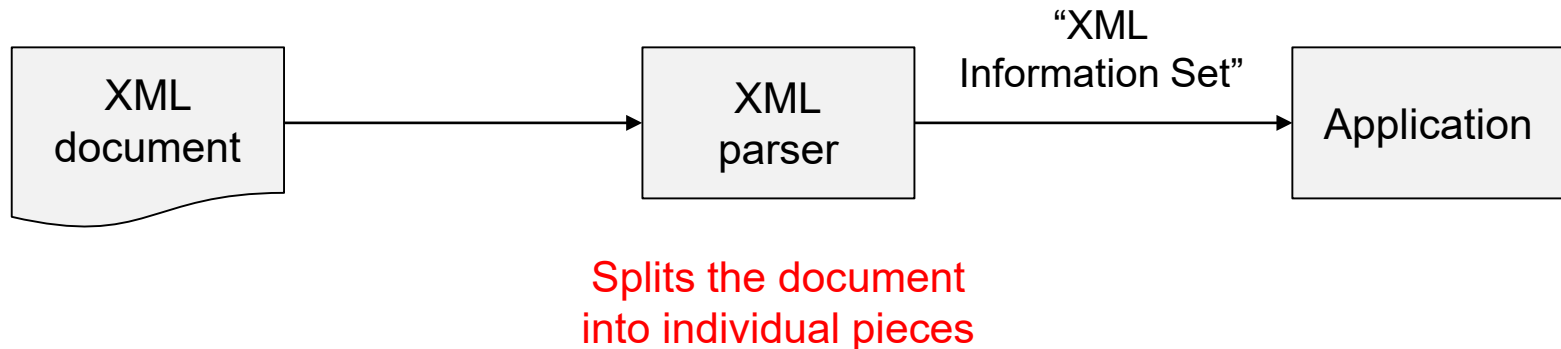
# What XML Is Not

- **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

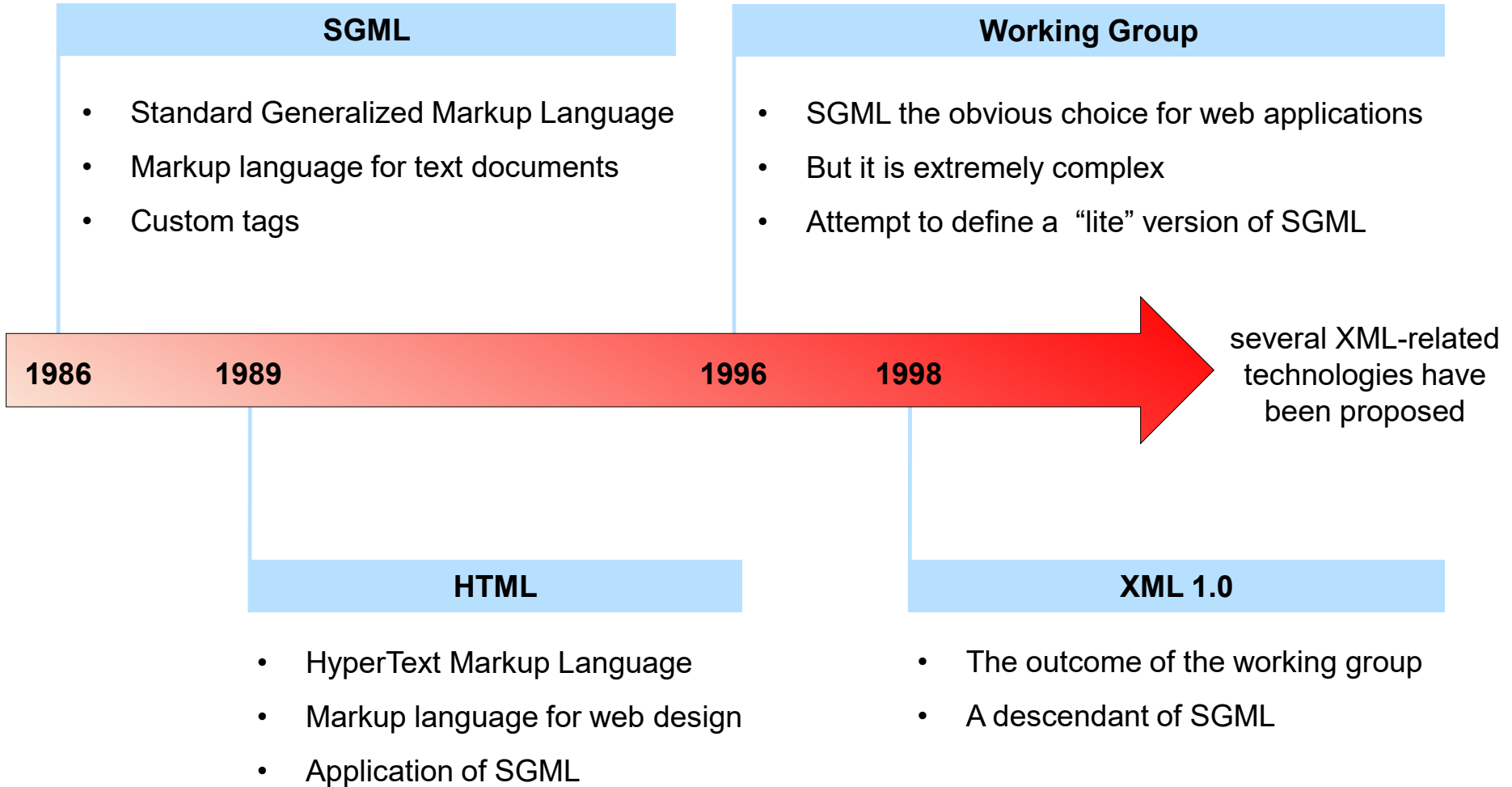
**ATTENTION:** XML documents simply exist - they do nothing

# 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



# The Evolution of XML



# Outline

- **XML at First Glance:**

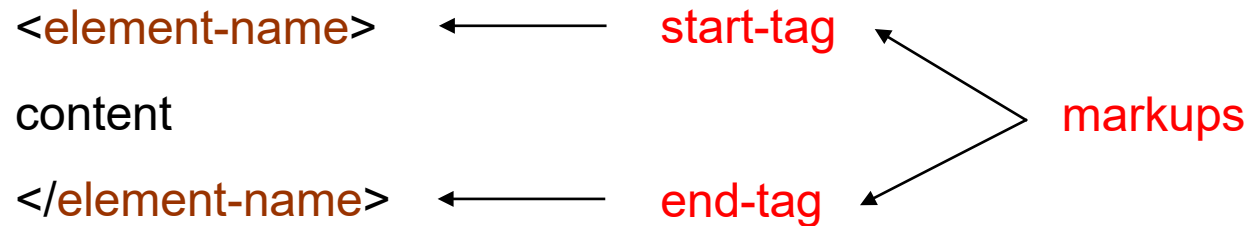
- The Benefits of XML
- XML vs. HTML
- What XML Is Not
- How XML Works
- The Evolution of XML

- **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



- 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

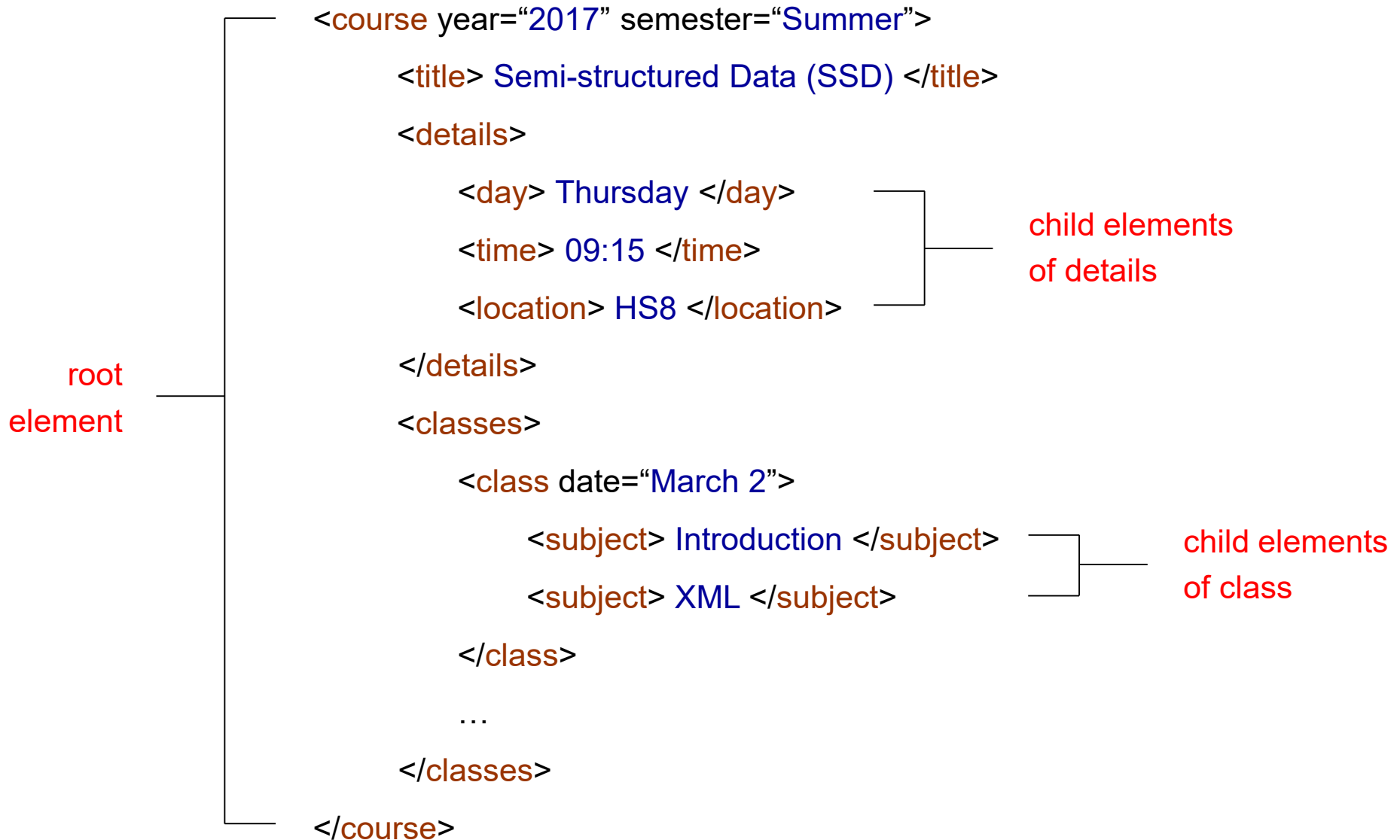
<course>

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



# XML Trees

- An element may have **several child elements**
- An element (apart from the root) has **exactly one 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

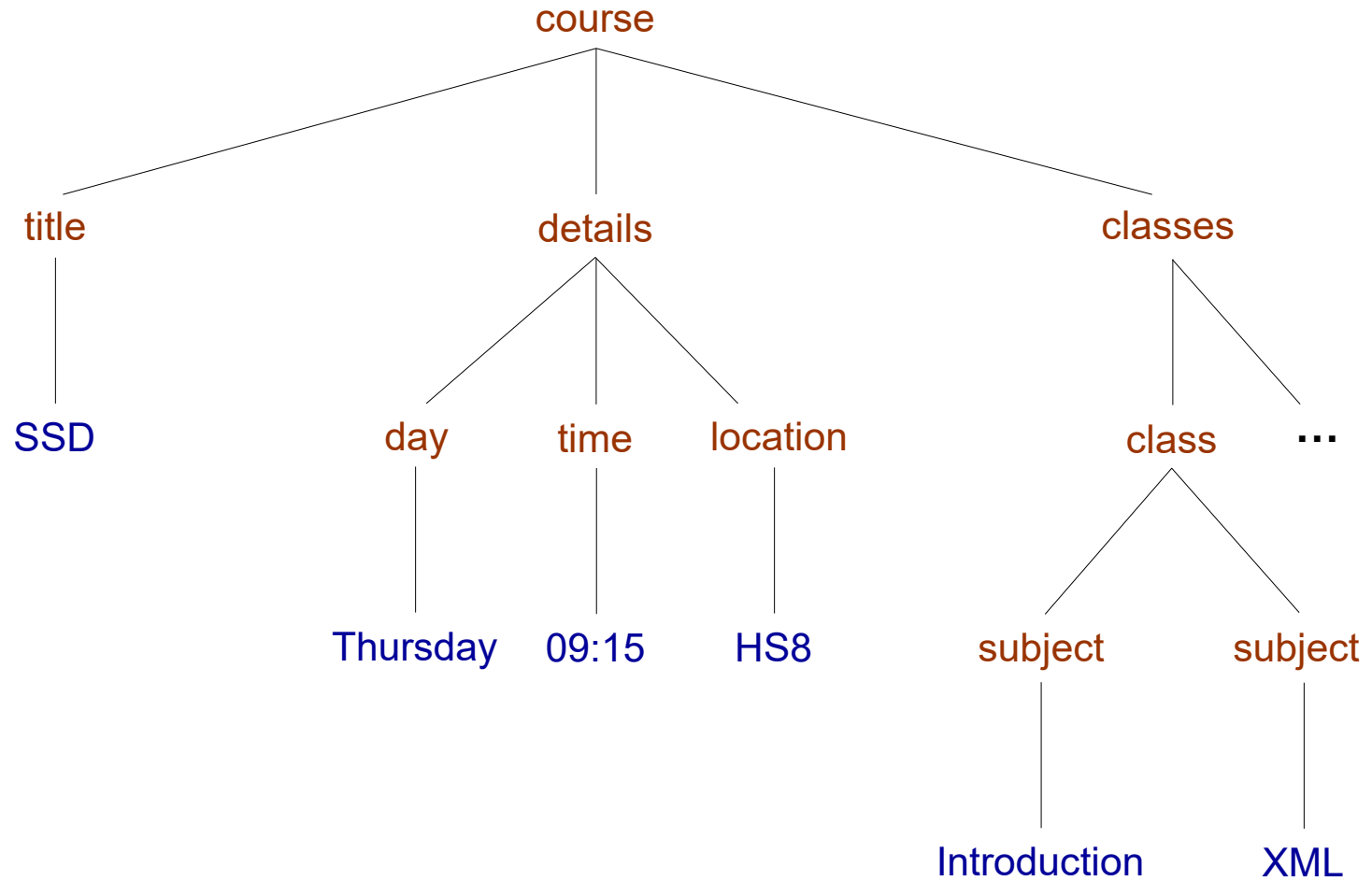
</course>

</title>





# XML Trees



# Attributes

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

```
<course year="2017" 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-name1="value1" ... attr-namen="valuen">`  
content  
`</element-name>`

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

- The order of attributes is not significant
- $\text{attr-name}_i = \text{value}_i$  &  $\text{attr-name}_j = \text{value}_j$  are the same

<code>&lt;course year="2017" semester="Summer"&gt;</code>	<code>&lt;course semester = 'Summer' year = '2017'&gt;</code>
<code>&lt;title&gt; Semi-structured Data &lt;/title&gt;</code>	<code>&lt;title&gt; Semi-structured Data &lt;/title&gt;</code>
<code>&lt;/course&gt;</code>	<code>&lt;/course&gt;</code>

# 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

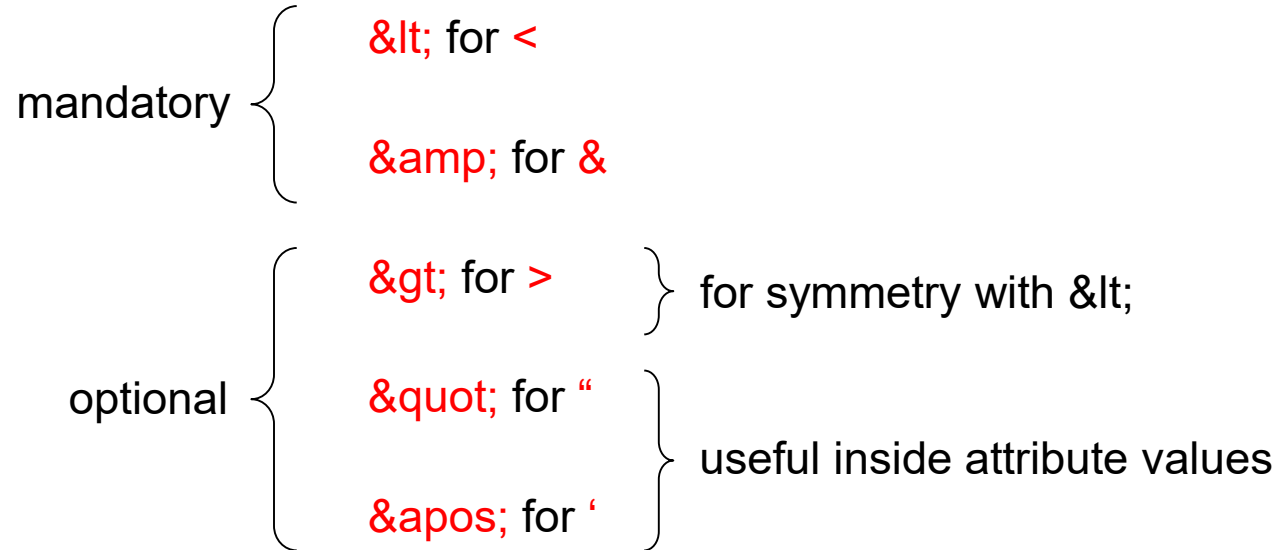
- The character data inside an element may not contain the symbol <

<less-than>		<less-than>
1 < 2	→	1 &lt; 2
</less-than>		</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**:



- 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**

`<course>`

`<!-- comment -->`

`</course>`



`<course <!-- comment --> >`

`</course>`

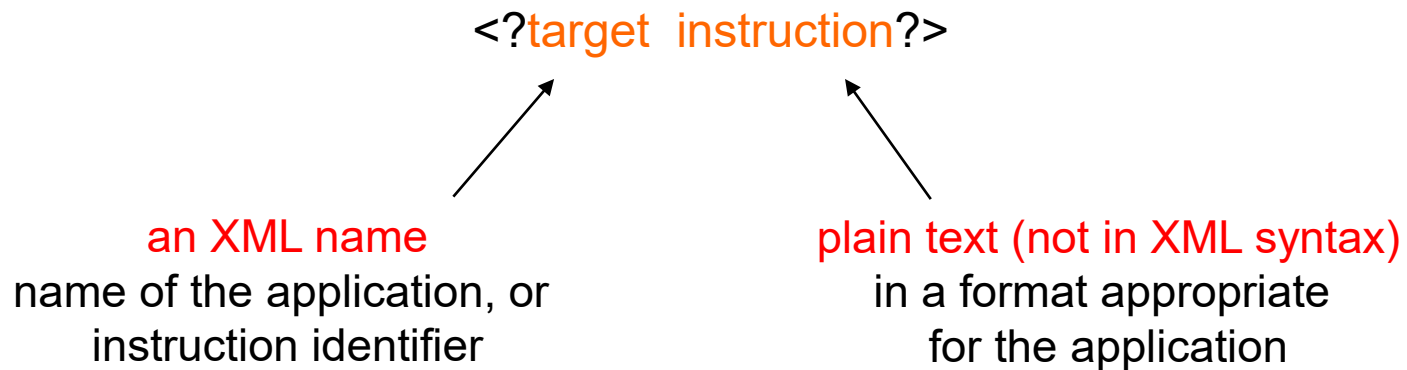


**ATTENTION:** Comments are not elements



# Processing Instructions

- A way of passing information to applications



- 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

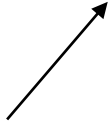
[http://www.w3schools.com/xml/xml\\_display.asp](http://www.w3schools.com/xml/xml_display.asp)

# 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 diagram shows three arrows pointing upwards from the descriptive text blocks to the corresponding attributes in the XML declaration above. The first arrow points from 'specifies the XML version...' to 'version="1.0"'. The second arrow points from 'the character encoding...' to 'encoding="ISO-8859-1"'. The third arrow points from 'whether the document is standalone...' to 'standalone="yes"'.

the character  
encoding that the  
document uses  
(default is UTF-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="2017" semester="Summer">
  <title> SSD </title>
  <details>
    <day> Thursday </day>
    <time> 09:15 </time>
    <location> HS8 </location>
  </details>
  <classes>
    <class date="March 2">
      <subject> Introduction </subject>
      <subject> XML </subject>
    </class>
  </classes>
</course>
```

```
<course year="2017" semester="Summer">
  <title> SSD </title>
  <details>
    <day> Thursday </day>
    <time> 09:15 </time>
    <location> HS8 </location>
  </details>
  <classes>
    <class date="March 2">
      <subject> Introduction </subject>
      <subject> XML </subject>
    </classes>
  </class>
</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="2017" semester="Summer">
  <title> Semi-structured Data (SSD) </title>
  <details>
    <day> Thursday </day>
    <time> 09:15 </time>
    <location> HS8 </location>
  </details>
  <classes>
    <class date="March 2">
      <subject> Introduction to the Module & Course </subject>
      <subject> Introduction to SSD </subject>
      <subject> XML </subject>
    </class>
    ...
  </classes>
</course>
```

... available at the TUWEL course