

# Einführung in XML

## Eine Sprache in einer Stunde

*Christiane Fritze*



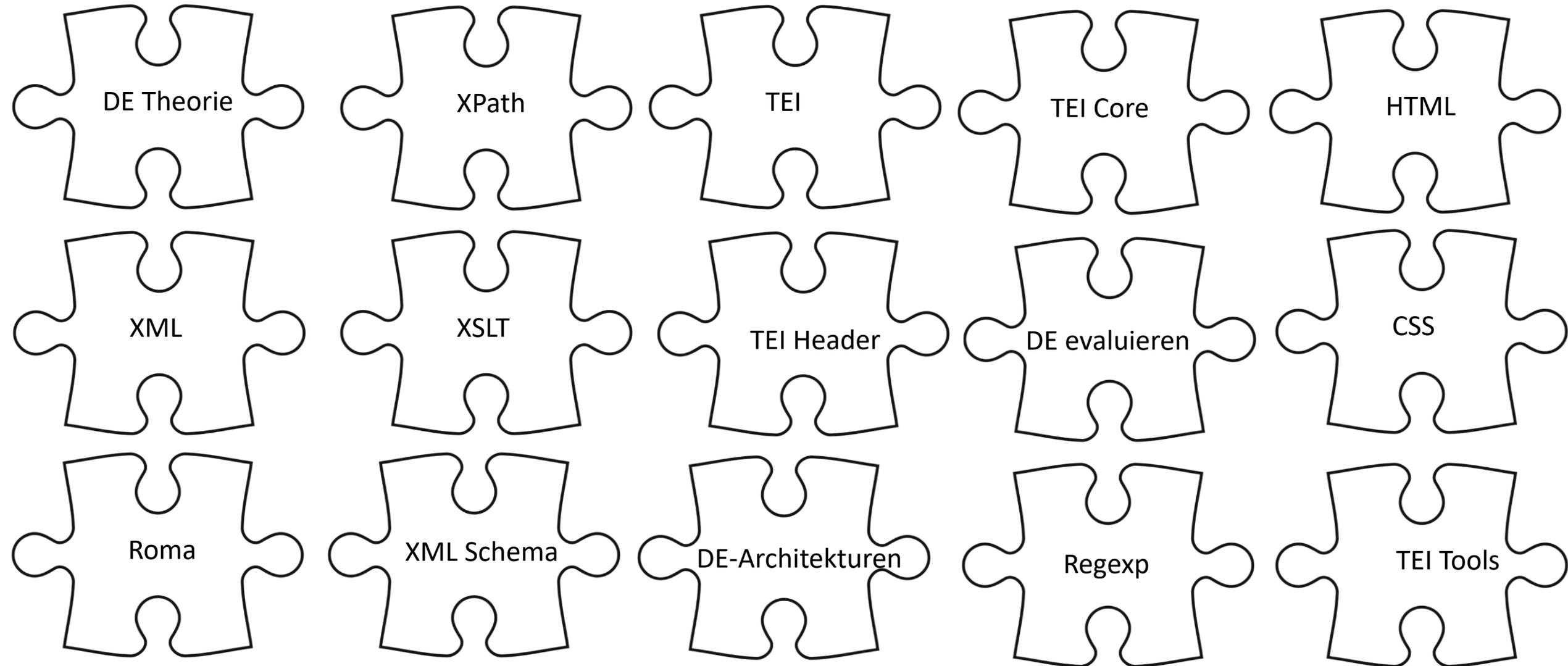
Leopoldina  
Nationale Akademie  
der Wissenschaften



berlin-brandenburgische  
AKADEMIE DER WISSENSCHAFTEN



# IDE School Puzzle



# Übersicht – Einführung in XML

---

- Über XML
  - Was ist das
  - Warum
  - Vor- und Nachteile
- Regeln, Regeln, Regeln
- XML-Editor kennen lernen
- Üben

# Über XML

Was? Warum und wofür? Vor- und Nachteile

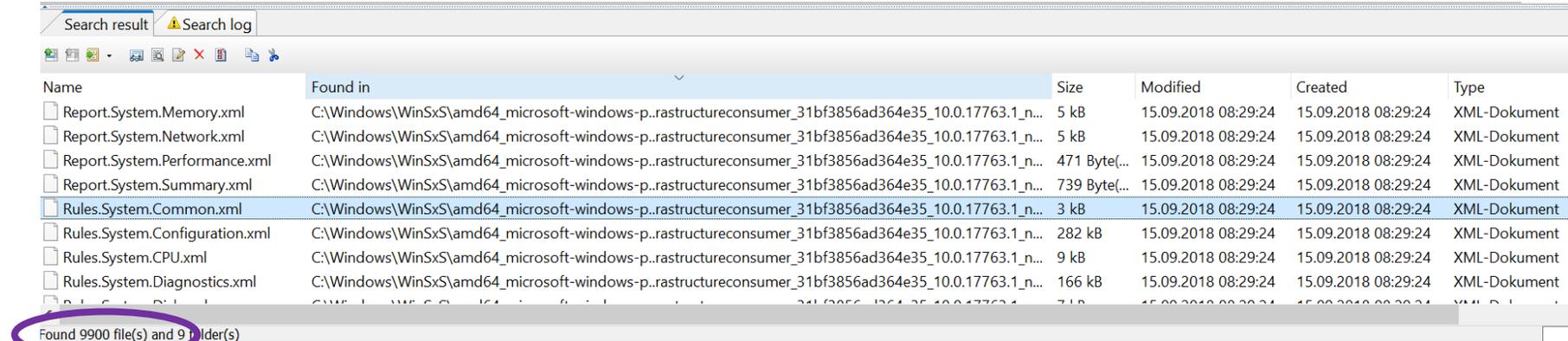
# Warum XML?

---

- XML ist ein weit verbreiteter Standard für die Beschreibung und den Austausch von Daten.
  - Standard des W3C seit 1998
- XML umfasst eine ganze Familie von begleitenden Standards.
- XML wird von einer breiten Softwarepalette unterstützt.
- XML ist die Grundlage vieler Anwendungsstandards („XML is everywhere“).
- XML ist menschen- und maschinenlesbar.
- XML ist einfach.

# XML als Grundlage von ...

- Einzelne Datenformate
  - z.B. xHTML, TEI, EAD, METS/MODS, SVG, KML etc.
- Datenaustausch
- Verschiedene Nutzungen
  - Webpräsentation
  - Visualisierungen
  - Druckausgaben
  - Ebooks
  - etc.



The screenshot shows a Windows search result window titled "Search result" with a "Search log" icon. The window displays a list of XML files found in the Windows system directory. The files are listed in a table with columns for Name, Found in, Size, Modified, Created, and Type. The file "Rules.System.Common.xml" is highlighted in blue. At the bottom of the window, a status bar indicates "Found 9900 file(s) and 9 folder(s)".

Name	Found in	Size	Modified	Created	Type
Report.System.Memory.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	5 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Report.System.Network.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	5 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Report.System.Performance.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	471 Byte(...)	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Report.System.Summary.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	739 Byte(...)	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Rules.System.Common.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	3 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Rules.System.Configuration.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	282 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Rules.System.CPU.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	9 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument
Rules.System.Diagnostics.xml	C:\Windows\WinSxS\amd64_microsoft-windows-p...rastructureconsumer_31bf3856ad364e35_10.0.17763.1_n...	166 kB	15.09.2018 08:29:24	15.09.2018 08:29:24	XML-Dokument

# Definition

---

- „XML definiert eine Syntax, um strukturierte Datenbestände jeder Art mit einfachen, verständlichen Auszeichnungen zu versehen, die zugleich von Anwendungen der unterschiedlichsten Art ausgewertet werden können.“

H. Vonhoegen, Einstieg in XML, Bonn : Galileo Press, 2007, S. 31

# XML ...

---

- **XML trennt Struktur und Darstellung** voneinander, Inhalt und Präsentation.
- XML-Dokumente werden nach einem **Dokumentenmodell** entwickelt.

# Vorteile von XML

---

- XML-Dokumente sind langzeitarchivierbar.
- XML-Dokumente sind plattformunabhängig.
- Browser können XML-Dokumente anzeigen und gegebenenfalls transformieren.
- Zur Dokumenterstellung reicht theoretisch ein simpler Texteditor aus.
- besonders gut geeignet, um Texte zu strukturieren.
- erweiterbar, jeder kann sein eigenes Vokabular modellieren.
- große internationale Nutzer- und Entwicklergemeinschaft.

# Nachteile von XML

---

- sehr „geschwätzig“, d.h. speicherintensiv.
- Overlapping-Problem.
- relativ langsamer Zugriff auf Daten, verglichen mit relationalen Datenbanken.

# Regeln, Regeln, Regeln

---

# Alles über ein XML-Dokument

---

- Ein XML-Dokument kann enthalten:
  - Elemente, eventuell mit Attributen
  - Verarbeitungsanweisungen
  - Kommentare
  - Entitätsreferenzen
- Ein XML-Dokument muss wohlgeformt sein und kann validiert werden.
- XML Attributwerte müssen in " " stehen.
- XML-Dokumente sind als lineare Zeichenketten kodiert.
- XML-Dokumente beginnen mit einer besonderen Verarbeitungsanweisung.
- Mehrere Leerzeichen oder Leerzeilen in Folge werden zu einem Leerzeichen.

# Mini XML-Dokument

---

- XML Deklaration `<?xml` und ein Element `<gruß>`

```
<?xml version="1.0" encoding="UTF-8"?>  
<gruß>Hallo Halle!</gruß>
```

# Mini XML-Dokument

- XML Deklaration `<?xml` und ein Element `<gruß>` und ein Attribut `modus`

```
<?xml version="1.0" encoding="UTF-8"?>  
<gruß modus="freundlich">Hallo Halle!</gruß>
```

# Mini XML-Dokument

- XML Deklaration `<?xml` und ein Element `<gruß>` und ein Attribut `modus` und ein `<!-- Kommentar -->`

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- kömmer kommentieren -->
<gruß modus="freundlich">Hallo Halle!</gruß>
```

# Mini XML-Dokument

- XML Deklaration `<?xml` und Elementen (`<winterschool>`, `<gruß>`, `<pause>`) und ein Attribut `modus` und ein `<!--` Kommentar `-->` und eine Entitätsreferenz `&amp;`;

```
<?xml version="1.0" encoding="UTF-8"?>
<winterschool>
  <!-- kömmer kommentieren und schachteln-->
  <gruß modus="freundlich">Hallo Halle!</gruß>
  <pause>Bald gibt es Kaffee &amp; Hallorenkugeln.</pause>
</winterschool>
```

# Übung

---

# Übung 1: XML selber schreiben, am Beispiel

- ein Beispiel: eine Postkarte
- Andere Beispiele?



# Übung 1: XML selber schreiben, am Beispiel

- Bilddateien im GoogleDrive öffnen oder herunterladen  
<http://tinyurl.com/IDESchool2019>



- oXygen öffnen; **STRG N** klicken; *Neues Dokument* anklicken; *XML-Dokument* wählen

# Übung 1: XML selber schreiben, am Beispiel

- Aufgaben
  - Versuchen Sie, Ihr Wissen über das Dokument festzuhalten
  - Textbestand?
  - Strukturen?
  - Zusätzliches Wissen?
  - Welche Informationen werden gebraucht, um später eine Edition daraus zu machen?
- Lassen Sie sich helfen: gibt es rote Unterkringelungen?

# Alles über XML

- Alles ist Text, Zeichendaten
- Markup bedeutet, **Text mit Auszeichnungen zu versehen...**
- Es gibt (Auszeichnungs-)Elemente
  - Öffnendes Tag ... Text ... Schließendes Tag
  - `<elementname>Elementinhalt</elementname>`
- Elemente können Attribute haben
  - `<elementname attributname="attributwert">Elementinhalt</elementname>`
- Elemente enthalten Elemente, Text, beides – oder nichts
- → Elemente können geschachtelt sein; es gibt *ein* Wurzelement

# XML Regeln

---

- Zeichen
- Aufbau einer XML-Datei
- Schachtelung
- Syntax / Regeln für Namen
- Attribute
- Wohlgeformtheit / Validität

# XML Regeln: Zeichen

---

- Alle Zeichen (z.B. aus Unicode) sind erlaubt ...
- außer ...
  - `<`     `&lt;`
  - `>`     `&gt;`
  - `&`     `&amp;`
- „whitespace“ ... spielt keine Rolle

# XML-Regeln: Dokumentenaufbau

- Bestandteile einer XML-Datei

- *XML-Deklaration:*

```
<?xml version="1.0" encoding="UTF-8"?>
```

- *Processing instructions:*

```
<?xml-stylesheet type="text/xsl" href="abc.xsl"?>
```

- *Elemente, geschachtelt:*

```
<root> ... </root>
```

- Kommentare

- *<!-- hier steht ein Kommentar -->*

# XML Element

- Ein Element hat immer einen Start-Tag und einen End-Tag.
  - Das ist ein `<StartTag>`.
  - Das ist das Ende vom `</StartTag>`.
- Attribute stehen im Starttag. `<gruß modus=„freundlich“>Hallo Halle!</gruß>`
- Die Tagnamen sind case-sensitive.
  - `<tag></tag>` und nicht `<tag></Tag>`.
- Leere Elemente können abgekürzt notiert werden:
  - `<LeererTag />`
- Alle Elemente müssen richtig geschachtelt werden.



# XML Elementnamen

---

- Elementnamen müssen mit einem **Buchstaben, Unterstrich** oder **Doppelpunkt** beginnen.
- Elementnamen können Buchstaben, Zahlen, Bindestriche, Punkte oder Unterstriche enthalten sowie Umlaute und Akzente.
- Elementnamen können beliebig lang sein.
- Nicht erlaubt:
  - `xml` und *Zahlen* zu Beginn
  - `<` `>` `&` `'` und `"`

# Element oder Attribut?

- Bei der Dokumentmodellierung muss immer abgewogen werden, welche Informationen elementwürdig, welche attributwürdig sind.
  - Daumenregel: alles, was automatisch auswertbar sein soll: normiert in Attribute
- Pro Element sind mehrere Attribute erlaubt.
- Aber: nicht gleiche Attribute in einem Element
  - nicht `<school wann="Februar" wann="März">25.2.-1.3.2019</school>`
- Jedoch: gleich benannte Attribute in allen Elementen möglich.
  - jedoch `<school>  
    <datum typ="start">25.2.2019</datum>  
    <datum typ="end">01.03.2019</datum>  
</school>`

# XML Prüfung

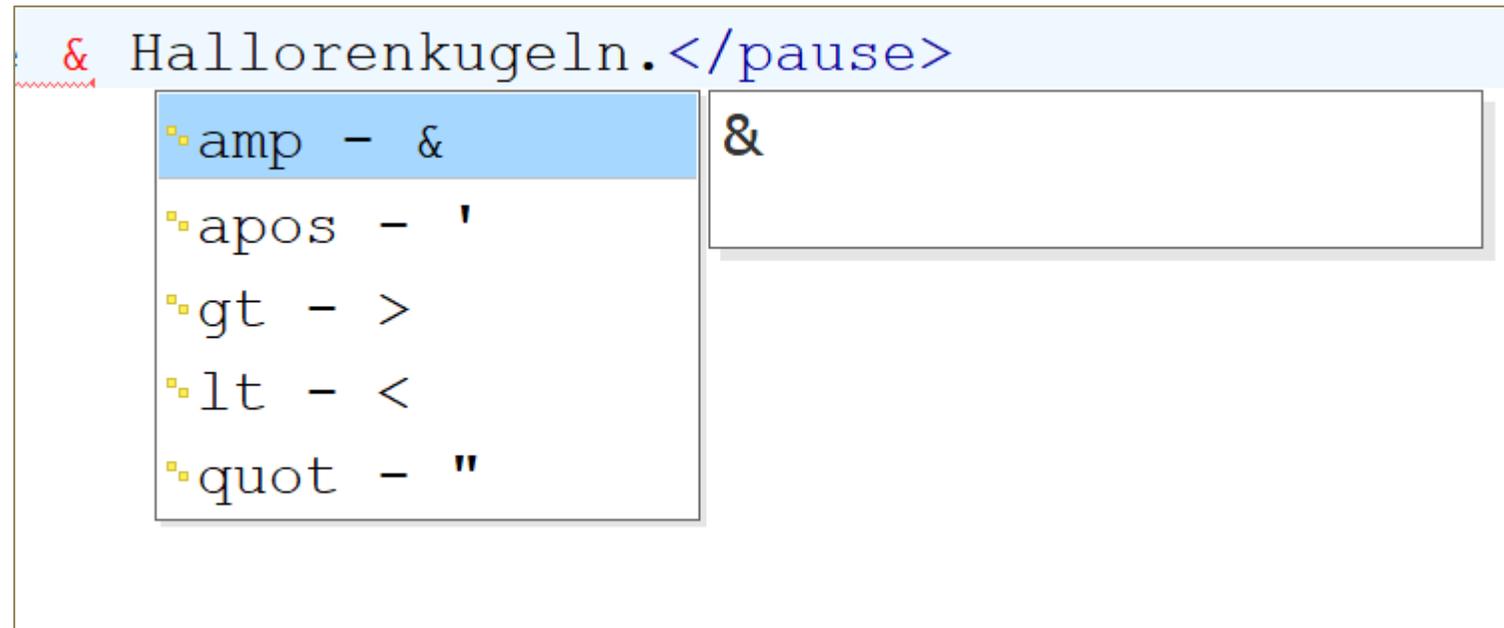
---

- Die Prüfung erledigt ein XML-Parser.
- XML-Dokument prüfen auf Wohlgeformtheit
  - Ein XML-Dokument ist **wohlgeformt**, wenn es den Regeln des XML-Standards entspricht.
    - ... *alle Syntaxregeln eingehalten, alle Elemente geschlossen, keine Überlappungen*
    - ... *wohlgeformte Dokumente können geparkt und verarbeitet werden*
- XML-Dokument prüfen auf Gültigkeit/Validität
  - Ein XML-Dokument ist **valide**, wenn es wohlgeformt ist und der Grammatik des XML-Schemas entspricht.

# Entitäten

- Entitäten stehen für etwas anderes.
- XML erlaubt, mit Referenzen Bezüge zu Entitäten herzustellen.
- Referenz wird vom XML-Prozessor durch die Entität ersetzt.
- bereits reservierte Entitätsreferenzen:

- `&lt;`           <
- `&gt;`           >
- `&quot;`         "
- `&amp;`         &
- `&apos;`       '



# Namespaces – Namensräume

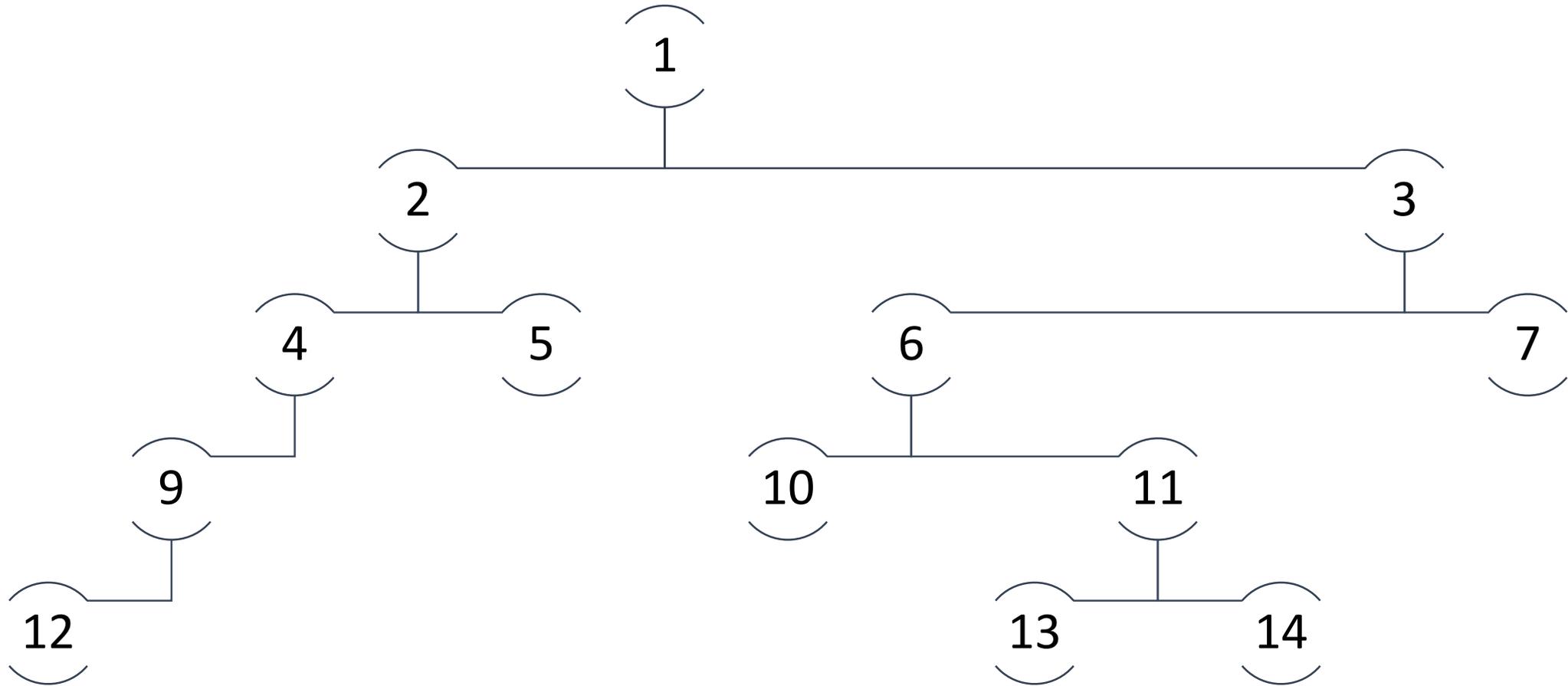
- Notwendig, wenn zwei Vokabularien in einem Dokument verwendet werden
  - `<title>PhD</title> ≠ <title>Einer flog übers Kuckucksnest</title>`
- Mit der Angabe des Namensraums wird also der Kontext angegeben, in dem ein bestimmter Name gilt.
- Namensräume werden über eine URI-Referenz angegeben.
- Es gibt Namensräume für Elemente und für Attribute.
- Wie werden Namensräume angegeben:
  - Zeichenfolge vor Elementnamen, mit `:` abgeschlossen
    - Beispiel: `<xsl:template match="msDesc"/>`
  - Namensräume müssen im Dokument deklariert werden
    - Mit Attribut `xmlns` im Wurzelement, URI als Attributwert
    - `xmlns="http://www.w3.org/1999/XSL/Transform"`

# Alles über ein XML-Dokument ...

- ... in anderen Worten:
- XML-Dokumente repräsentieren einen **Baum**.
- XML-Dokumente müssen **eine** – und nur eine – **Wurzel** haben, und viele Knoten.
- Jeder **Knoten** kann sein
  - ein Teilbaum
  - ein einzelnes Element (evtl. mit Attributen)
  - eine Zeichenkette
- Es gibt **Elternknoten** und **Kindknoten** und **Geschwisterknoten**



# XML Baum



# Referenzen

---

- <https://www.w3.org/XML/>
- <https://www.w3.org/TR/xml/>
- <https://www.w3.org/TR/xml-names/>
- <https://wiki.selfhtml.org/wiki/XML>
- H. Vonhoegen, Einstieg in XML, Bonn : Galileo Press, 2007