

Willkommen zur



```
<fortbildung titel='xml'>
  <fortbildner>
    <name>Michael Dienert</name>
    <schule>Walther-Rathenau GS Freiburg</schule>
  </fortbildner>
  <fortbildner>
    <name>Sascha Mertens</name>
    <schule>Walther-Rathenau GS Freiburg</schule>
  </fortbildner>
  <fortbildner>
    <name>Jürgen Schnaiter</name>
    <schule>Friedrich-August-Haselwander GS Offenburg</schule>
  </fortbildner>
</fortbildung>
```

Zeitplan heute



```
<tag dow='Montag' datum='2014-11-10' fortbildner='S.Mertens'>
  <material>http://web.wara.de/~mertens/fortbildungxml</material>
  <termin zeit='10:00:00'>Begrüßung</termin>
  <thema>XML Grundlagen</thema>
  <thema>Document Type Definition (DTD)</thema>
  <termin zeit='12:30:00'>Mittagessen</termin>
  <termin zeit='13:30:00'>Beginn Nachmittagsession</termin>
  <thema>XML Schema (XSD)</thema>
  <termin zeit='15:30:00'>Kaffeetrinken</termin>
  <thema>XPath</thema>
  <termin zeit='18:00:00'>Abendessen</termin>
</tag>
```

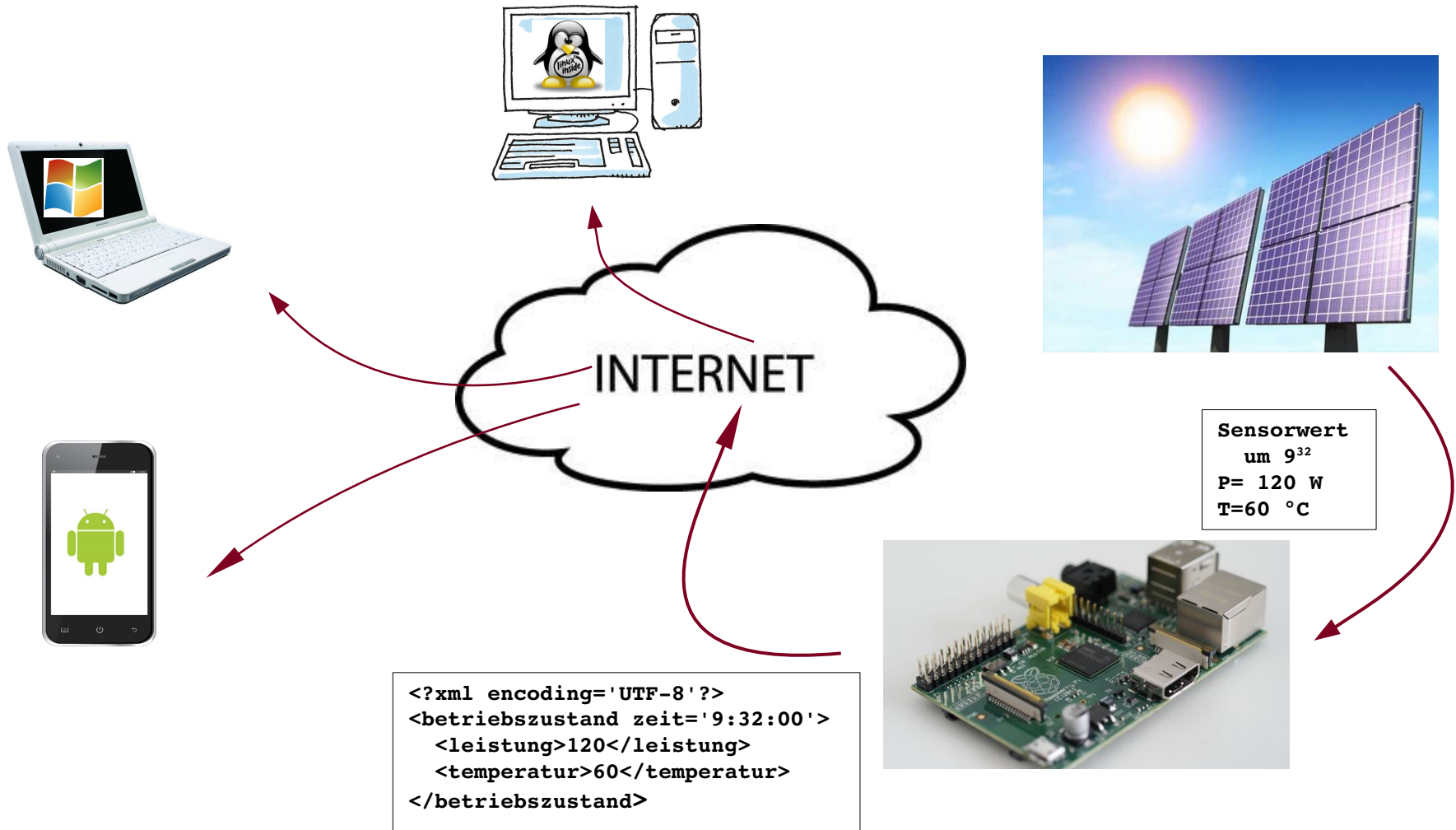
XML – Wieso, weshalb, warum ?



XML

- steht für EXtensible Markup Language
- ist eine Sprache, um Daten in für Menschen lesbarer Form zu speichern
- wird eingesetzt, um Daten *unabhängig* von der verwendeten *Software, Plattform* und *Hardware* auszutauschen

XML im Einsatz



XML – You use it !



- SVG = Scalable Vector Graphics → Beliebtes Grafikformat im Internet (Wikipedia!)

```
<g inkscape:label="Ebene 1"
  inkscape:groupmode="layer"
  id="layer1">
  <rect style="fill:none;stroke:url(#linearGradient5203);"
    id="rect2985"
    width="231.27069"
    height="299.8421"
    x="71.507515"
    y="229.58398" />
  <text xml:space="preserve"
    style="font-size:40px;font-style:normal;font-weight:normal;"
    x="94.285713"
    y="395.21933"
    id="text3755"
```

XML – You use it !



- „Open Document Format“ (Open/LibreOffice) und „Office Open XML“ (Microsoft Office) → zip-komprimiertes XML

```
<office:presentation>
  <draw:page draw:name="page1" >
    <office:formsn />
    <draw:frame presentation:style-name="pr1">
      <draw:text-box/>
    </draw:frame>
    <draw:frame presentation:style-name="pr2"
      draw:layer="layout" svg:width="25.199cm"
      svg:height="12.692cm" svg:x="1.4cm"
      svg:y="4.4cm" presentation:class="subtitle">
      <draw:text-box>
        <text:p>Willkommen zur Fortbildung</text:p>
        <text:p>XML</text:p>
      </draw:text-box>
    </draw:frame>
    <presentation:notes draw:style-name="do2">
```

Wohlgeformtes XML (1)



- ein XML Datei enthält genau ein Wurzelement
- ein XML Element besteht aus einem öffnenden und einem schließenden „Tag“ (engl. start-tag, end-tag)
- ein „Tag“ ist der Elementname umschlossen von spitzen Klammern. Ein schließendes Tag wird durch einen slash vor dem Namen gekennzeichnet

<thema>...</thema>

- kein Leerzeichen vor dem Tagname !
< thema> ist falsch
- XML ist case-sensitive – bei allen Namen ist Groß- und Kleinschreibung zu berücksichtigen

Wohlgeformtes XML (2)



Ein Element (=Tag)

- kann ein weiteres Element enthalten oder einen Inhalt in Textform

```
<fortbildner>
```

```
  <name>Sascha Mertens</name>
```

```
</fortbildner>
```

- kann Attribute enthalten. Diese werden im Tag als Name/Wert-Paar angegeben. Der Wert wird in doppelte oder einfache Anführungszeichen gesetzt

```
<termin zeit = '12:00'>Mittagessen</termin>
```

- kann selbstschließend sein, also nur aus einem Tag bestehen:

```
<termin zeit='6:00' aktion='Mittagessen' />
```


Falsche Schachtel



```
<fortbildner>  
  <name>Sascha Mertens  
</fortbildner>  
  </name>
```

- ...ist ein Verschachtelungsfehler. Das End-Tag eines Unterelements muss vor dem End-Tag des umschließenden Element stehen.

Wohlgeformtes XML - Optionales



Eine XML Datei kann

- und sollte eine „Präambel“ enthalten, in der zumindest die Versionsnummer und der verwendete Zeichensatz deklariert werden

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

- (mehrzeilige) Kommentare enthalten

```
<!-- Zeile wird ignoriert -->
```

- Strukturinformation einbinden (DTD, XSD)

Aufgabe 0



- Installieren Sie auf dem Fortbildungsraum-PC das Java-JDK und die Entwicklungsumgebung Netbeans. Die Installationsdateien `jdk-8u25..exe` und `netbeans-8.0.1...exe` finden Sie im Ordner
`T:\\R012\\xml_fobi\\tools`
- man braucht zwar keine Java-Entwicklungsumgebung, um XML-Dateien zu schreiben, aber IDEs wie z.B. Netbeans oder Eclipse erleichtern das Arbeiten mit XML, DTD und XSchema ungemein
- das es aber auch ganz ohne geht, probieren Sie Aufgabe 1, die Sie bearbeiten, während die Installation läuft...

Aufgabe 1: XML (10 min)



- Schreiben Sie einen kurzen XML-Steckbrief zu sich selbst. Benutzen Sie Stift und Papier und erzählen Sie uns von sich, was sie möchten – einzige Vorgabe:

Sie müssen wohlgeformtes XML produzieren



Document Type Definiton DTD

Wohlgeformtes XML – Alles gut ?



- ein lediglich wohlgeformtes XML-Dokument ist für den Austausch von Daten zwischen Programmen nutzlos - für erfolgreichen Datenaustausch müssen die möglichen Inhalte eines XML-Dokumentes festgelegt werden
- zur expliziten Festlegung der erlaubten Dokumentenstruktur dienen

Document Type Definition (DTD)

XML Schema (XSD)

- man spricht von „gültigem“ XML, wenn eine Datei dem beigefügten DTD bzw. XML Schema entspricht

Document Type Definition → DOCTYPE



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

<!DOCTYPE fortbildung [
    <!ELEMENT fortbildung (fortbildner+,tag*,beschreibung?)>
    <!ELEMENT fortbildner (name,schule)?>
    <!-- ... -->
]>

<fortbildung>
  <fortbildner>
    <name>Sascha Mertens</name>
    <schule>
      Walther Rathenau GS Freiburg
    </schule>
  </fortbildner>
</fortbildung>
```

- die DTD Befehle werden über den DOCTYPE-Befehl eingebaut:

<!DOCTYPE root-element **[element-declarations]>**

DTD in eigener Datei



```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE fortbildung SYSTEM "fobi1.dtd">
<fortbildung>
  <!-- ... -->
</fortbildung>
```

fobi.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT fortbildung (fortbildner+,tag*,beschreibung?)>
<!ELEMENT fortbildner (name,schule)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT schule (#PCDATA)>
```

fobi1.dtd

<!DOCTYPE root-element **SYSTEM** 'dateiname'>

Auch DTD-Dateien sollten ein Präambel enthalten (Encoding !)

DTD - Element



- Ein Element definiert man mit
<!ELEMENT name (content)**>**
- „content“ gibt an, ob das Element
 - weitere Elemente enthält (kommagetrennte Liste)
 - Daten enthält (**#PCDATA**)
 - nichts (**EMPTY**) bzw. etwas beliebiges (**ANY**) enthält

```
<!ELEMENT fortbildung (fortbildner+,tag*)>  
<!ELEMENT fortbildner (name,schule)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT schule (#PCDATA)>
```

Leerzeichen !!

„Parsed Character Data“ = Daten



```
<!ELEMENT fortbildner (name,schule)>
```

- eine Elementliste (Sequenz) ist verbindlich: „fortbildner“ *muss* die Elemente „name“ und „schule“ enthalten und zwar in dieser Reihenfolge

```
<!ELEMENT fortbildung (fortbildner+,tag*,beschreibung?)>
```

- die erlaubte Anzahl der Elemente kann genauer spezifiziert werden:
 - beliebig (keines oder mehr) *
 - mindestens eins (eins oder mehr) +
 - optionales Element (eins oder keins) ?

DTD Auswahlliste



- Unterelemente eines Elements können auch als Auswahlliste angegeben werden:
- **<!ELEMENT** tag (termin | thema)* **>**
→ „tag“ enthält beliebig viele Termine und Themen
- Achtung: innerhalb der Auswahlliste keine Kardinalitäten
- Sequenzen und Auswahllisten kann man mischen:
<!ELEMENT kontakt (name, (email | telefon))**>**
→ „kontakt“ enthält einen Namen und danach *entweder* eine Email *oder* eine Telefonnummer

DTD - Attribute



```
<!-- Elementname -->
<!-- Attributname Attributwert Attributtyp -->
<!-- Attributname datum CDATA #REQUIRED -->
<!-- Attributname dow (Montag|Dienstag|Mittwoch) #IMPLIED -->
```

- **<!--** element attr₁ attrTyp₁ wert₁ attr₂ attrTyp₂ wert₂ ... **-->**
- wichtigste Attributstypen:
 - CDATA beliebiger Text
 - (wert1 | wert2 | ...) Liste möglicher Werte
- wichtigste Attributswerte
 - #REQUIRED Attribut muss angegeben werden
 - #IMPLIED Attribut kann angegeben werden

Aufgabe 2: DTD → XML (15 min)



- Öffnen Sie mit Netbeans die Datei aufgabeMessung.xml

Ergänzen Sie XML-Daten, die die DTD erfüllen, die bereits in der Datei steht.

Benutzen Sie „Run“ → „Validate XML“ um Netbeans die Gültigkeit prüfen zu lassen.

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

<!ELEMENT messung (zeitpunkt,ort,messwert+)>

<!ELEMENT zeitpunkt EMPTY>
<!ATTLIST zeitpunkt
    zeit CDATA #REQUIRED
    datum CDATA #REQUIRED
    >
<!ELEMENT ort (stadt?,gps)>
<!ELEMENT stadt (#PCDATA)>
<!ELEMENT gps EMPTY>
<!ATTLIST gps long CDATA #REQUIRED lat CDATA #REQUIRED>
<!-- Für name z.B. Temperatur, Luftdruck, Strom-->
<!ELEMENT messwert (name,((einheit,wert)|fehlermeldung))>
<!ELEMENT name (#PCDATA)>
<!ELEMENT einheit (#PCDATA)>
<!ELEMENT wert (#PCDATA)>
<!ELEMENT fehlermeldung (#PCDATA)>
<!ATTLIST einheit praefix (f|p|n|m|k|M|G|T) #IMPLIED>
```

Aufgabe 3 – XML → DTD (15 min)



- Schreiben Sie eine DTD für die XML-Datei „musiksammlung.xml“

(!! „File → New File“,
nicht „File → New
Project“ !!)

```
<musiksammlung>
  <album titel='Ten'>
    <artist>Pearl Jam</artist>
    <songs>
      <song>Once</song>
      <song>Even Flow</song>
      <song>Alive</song>
      <!--...-->
    </songs>
  </album>
  <album titel='G3 live' erscheinungsjahr="1991">
    <artist>Joe Satriani</artist>
    <artist>Steve Vai</artist>
    <artist>Eric Johnson</artist>
    <songs>
      <song>Cool #9</song>
      <song>Flying in a blue dream</song>
      <!--...-->
    </songs>
  </album>
  <soundtrack filmtitel="Matrix">
    <track>
      <artist>Rage against the machine</artist>
      <song>Wake up</song>
    </track>
    <track>
      <artist>Prodigy</artist>
      <song>Mindfields</song>
    </track>
    <!-- ... -->
  </soundtrack>
</musiksammlung>
```

Mehr über XML



- <http://wiki.selfhtml.org/wiki/XML>
- <http://de.selfhtml.org/xml/index.htm> (veraltet, aber in der Prüfung verfügbar)
- <http://www.sitepoint.com/really-good-introduction-xml/>
- [http://www.w3.schools.com/\(xml|dtd\)](http://www.w3.schools.com/(xml|dtd))



Danke für Ihre Aufmerksamkeit