

# Austauschbare Datenformate & Parsing

Referat im Rahmen des Programmierpraktikums 2012

Peet Cremer

Lehrstuhl für Physik der weichen Materie  
Heinrich-Heine-Universität Düsseldorf,  
Universitätsstraße 1, D-40225 Düsseldorf, Germany

6. Juli 2012

# Gliederung

1 Einleitung

2 XML

3 Parser

# Gliederung

1 Einleitung

2 XML

3 Parser

# Datenformate

- Plattformunabhängige Datenformate besonders für das Internet wichtig
- Hierarchische Baumstrukturen
- Menschenlesbar
- Müssen geschrieben und ausgelesen werden: Parser

# Gliederung

1 Einleitung

2 XML

3 Parser

# XML

- **eXtensible Markup Language**
- Herausgegeben vom World Wide Web Consortium (W3C)
- Darstellung hierarchisch strukturierter Daten in Textdateien
- ASCII codiert und damit menschenlesbar
- Definiert strukturelle und inhaltliche Einschränkungen um als Grundlage für anwendungsspezifische Sprachen zu dienen
- Beispiele für XML-Sprachen: RSS, MathML, GraphML, XHTML, XAML, SVG

# Inhalte

- Elemente beginnen mit Start-Tag enden mit End-Tag
- Attribute, die zu einem Element gehören
- Verarbeitungsanweisungen
- Kommentare

# XML–Syntax

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <Level1 xsize="3" ysize="3">
    <fields>
      <field type="f" x="1" y="1"/>
      <field type="f" x="1" y="2"/>
      <field type="e" x="1" y="3"/>
      <field type="f" x="2" y="1"/>
      <field type="s" x="2" y="2"/>
      <field type="e" x="2" y="3"/>
      <field type="e" x="3" y="1"/>
      <field type="e" x="3" y="2"/>
      <field type="e" x="3" y="3"/>
    </fields>
    <spawnpoints>
      <spawnpoint x="1" y="1"/>
    </spawnpoints>
    <exits>
      <exit x="3" y="3"/>
    </exits>
    <Gelaber>
      BlaBla
    </Gelaber>
    <?Ziel--Name Parameter ?>
    <!-- Kommentar -->
  </Level1>

```



# XML-Regeln

XML-Dokument heißt **wohlgeformt** wenn es die XML-Regeln einhält

- Es gibt genau ein **Wurzelement** (das äußerste Element der Hierarchie)
- Alle Elemente besitzen ein **Begin-Tag** und ein **End-Tag** und den Inhalt dazwischen
- Elemente am ohne Unterelemente können alternativ auch in sich geschlossen werden
- Elemente enthalten nur ein Attribut mit demselben Namen

# Gliederung

1 Einleitung

2 XML

3 Parser

# Parser

XML-Dokumente sollen automatisch generiert und ausgelesen werden

- **Document Object Model**

- Das Ganze Dokument wird in den Speicher geladen
- Zugriff auf das Dokument als Baumstruktur
- Erlaubt Manipulation und Zurückschreiben in das XML-Dokument
- Speicherintensiv
- Für kleine Dokumente gut geeignet

- **Simple API for XML**

- XML-Dokument als sequentieller Datenstrom
- Auf Ereignisse im Datenstrom wird reagiert
- Wenig Speicherintensiv
- Keine Manipulation möglich
- Für große Dokumente gut geeignet