



# Praktikum zur Veranstaltung XML-Technologie: **Übung 08**

XPointer, XInclude, XLink



# Organisatorisches

---



- Arbeitsverzeichnis:

`~/lv/xmltech/08/`

- Dateinamen:

`08-news.xml`

`08-news.css`

`08-ads.xml`

`08-news[1-4].xml`

`08-fhlogo.svg`

- Abzugeben:

`alle`

- Werkzeuge:

`emacs`

`# oder X-Emacs`

`xmllint`

`# Zur Expansion von XInclude`

`firefox`

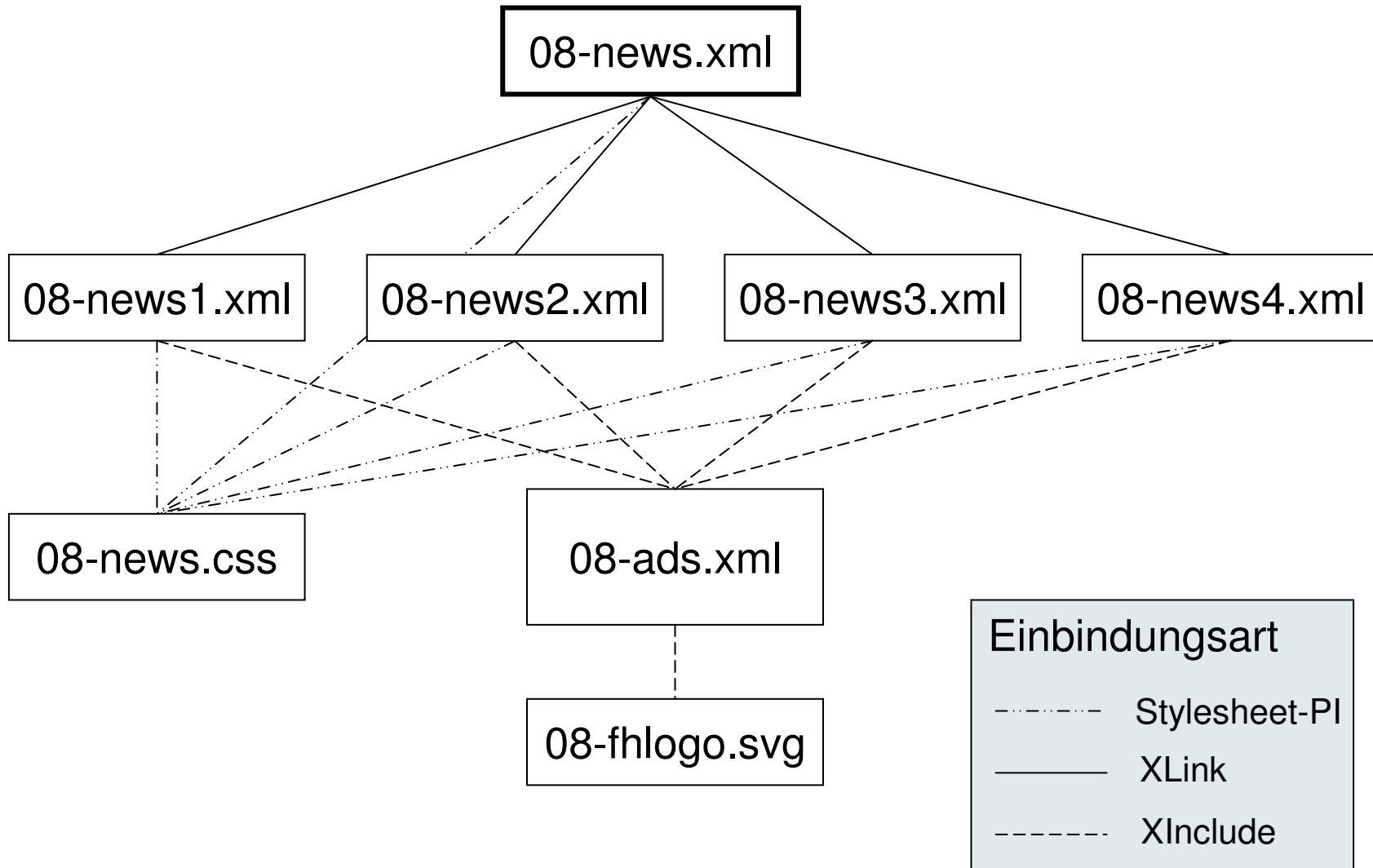
`# ab V 1.5`



- Aufgabe:
  - Simulieren Sie News-Seiten mit XML-Mitteln unter Verwendung von XInclude und XLink
  - Struktur
    - 1 Basis-Seite mit Überschrift „NEWS vom xx.mm.2008“ und (mindestens) 4 Einträgen.
    - 4 News-Artikel (oder mehr). Jeder Artikel genügt derselben DTD und enthält ein Element „title“. Die Einträge der Basis-Seite entsprechen diesen Titeln
    - 1 Datei mit „Werbung“. Jeder News-Artikel bindet mindestens einen Werbungsbaustein ein.
    - 1 SVG-Datei mit dem (einzelnen) FH-Logo
    - 1 gemeinsame CSS-Datei
- Abgabezeitpunkt:
  - Normal, spätestens zu Beginn der nächsten Übung.



# Dateiabhängigkeiten





- Verwenden Sie folgende Elemente:

news	Root-Element in 08-news.xml
ads	Root-Element in 08-ads.xml
article	Unterelement von „news“ in 08-news.xml, Root-Element von 08-news[1-9].xml
title	Erstes Unterelement von „article“
para	Weitere Unterelemente von „article“
ad	Unterelement von „article“, bindet einen Werbeblock aus Datei „08-ads.xml“ ein
list	Unterelement von „para“, leitet einen Listenblock ein
item	Unterelement von „list“
emph	Inline-Element zum „Betonen“ von Textabschnitten, z.B. kursiv in der Darstellung
div	Generisches blockbildendes Element
sp	Generisches Inline-Element, nützlich z.B. zum Setzen von XLink-Attributen



## A: Die Datei 08-news.xml

- Das Root-Element enthält so viele Unterelemente „article“, wie es Artikeldateien gibt.
- Jedes Unterelement „article“ besitzt einen XLink-Verweis auf die entsprechende Artikel-Datei.
  - Beim Anklicken soll das Zieldokument im selben Browser-Fenster/-Tab angezeigt werden.
- Jedes Unterelement „article“ enthält eine XInclude-Anweisung, mit der Element „title“ aus dem zugeordneten Artikel-Dokument eingebunden wird.
  - Die Darstellung von „08-news.xml“ besteht nur aus einer (per CSS erzeugten) Überschriftenzeile und je einer Zeile pro News-Artikel. Diese Zeilen werden aus den Inhalt der „title“-Elemente gebildet.
- Demonstrieren Sie mindestens einmal den Einsatz von „fallback“, etwa indem Sie auf eine fünfte, nicht vorhandene Artikeldatei verweisen und als „fallback“ den Text „FEHLER – Quelldatei fehlt“ anstelle vom Inhalt eines gefundenen „title“-Elements erzeugen.



B: Die Dateien 08-news1.xml, 08-news2.xml, ...

- Verwenden Sie als Nachrichtenquelle <http://www.heise.de/newsticker/>
- Wählen Sie (mindestens) 4 Nachrichten eines Tages aus
  - Eine dieser Nachrichten muss am Ende eine Liste mit Links auf frühere Artikel zum Thema haben
    - Jeder dieser Artikel muss mindestens 2 Links auf fremde Seiten enthalten
- Pro Artikel übertragen Sie den Text (per copy/paste) in die entsprechende XML-Datei und ergänzen ihn mit Markup
- Rekonstruieren Sie die beiden Links auf fremde Seiten mittels XLink. Beim Anklicken soll jeweils ein neues Browser-Fenster/-Tab geöffnet werden.
- Binden Sie zwischen dem ersten und zweiten Absatz einen Werbeblock aus Datei „08-ads.xml“ ein. Jede Datei soll einen eigenen Werbeblock erhalten: 08-news1.xml den ersten, 08-news2.xml den zweiten, usw.



## C: Die Datei 08-ads.xml

- Der Inhalt des Root-Elements „ads“ besteht aus einer Folge von Container-Elementen „div“
- In „div“ können Sie beliebige XML-Inhalte schreiben, auch SVG! Wichtig dabei ist nur, dass der Browser diese auch anzeigen kann, z.B. indem Sie entsprechende CSS-Regeln hinterlegen.
- Einer der Einträge soll aus dem FH-Logo (SVG-Datei) bestehen. Verwenden Sie dazu das in Übung 07 entwickelte Symbol, hinterlegen Sie es (zusammen mit einem geeigneten „use“-Element) in Datei 08-fhlogo.svg, und binden Sie es per XInclude ein.
- Bei den anderen (mindestens 3) „Werbebotschaften“ seien Sie erfinderisch – Texte genügen.





## D: Die Datei 08-news.css

- Gestalten Sie die XML-Dateien mit CSS-Mitteln so, dass sie gut lesbare Anzeigen erzeugen
- Fixieren Sie „article“ auf die Breite „700px“
- Inhalte des Elements „ad“ sollen 400px breit, zentriert und von einem Rahmen mit 2px Breite umgrenzt werden
- Erzeugen Sie eine Überschrift mittels „news:before“ mit dem Wortlaut „News-Übersicht bei heise.de vom xx.mm.2008“  
(xx.mm.2008 durch aktuelles Datum ersetzen)



- Firefox unterstützt XLink (type=„simple“), aber leider nicht XInclude
- Abhilfe zum Testen:
  - Mit „xmllint –xinclude in.xml > out.xml“ können Sie XInclude-Elemente expandieren und die Ergebnisse mit Ihrem Browser betrachten
- Anregung: Tests mit weiteren Browsern
  - Wer Zugang zu weiteren Browsern hat (Konqueror, Safari, IE7/8, Opera ...), kann diese auf XLink- und XInclude-Unterstützung testen und der Gruppe darüber berichten, z.B. per E-Mail-Verteiler
  - Gleiches gilt für Plugins oder Addons, die XInclude-Fähigkeiten nachrüsten