Uppmärkningsspråk

TDP007 Konstruktion av datorspråk
Föreläsning 3

LINKÖPINGS UNIVERSITET

Pontus Haglund/Jonas Wallgren
Institutionen för datavetenskap
2022-01-20

Vad handlar det andra seminariet om?

- Strukturerad text
- Uppgifter
  - *Hämta information ur en textfil med regexp*
  - *Hämta information ur en XHTML-fil med hjälp av en parser*
- Hur testar man?
  - *Testa delfunktioner*
  - *Läsbarhet av testerna, kommunicera beteendet väl*
  - *Dokumentera tolkningen i bloggen*

Översikt över föreläsning 3-4

- Uppmärkningsspråk
- Struktur och specifikation av XML-dokument
- Parsning av XML-dokument
  - *Strömparsning*
  - *Trädparsning*
- Mikroformat
- Inför seminariet

Uppmärkningsspråk

Alice's Adventures in Wonderland — title, 16p

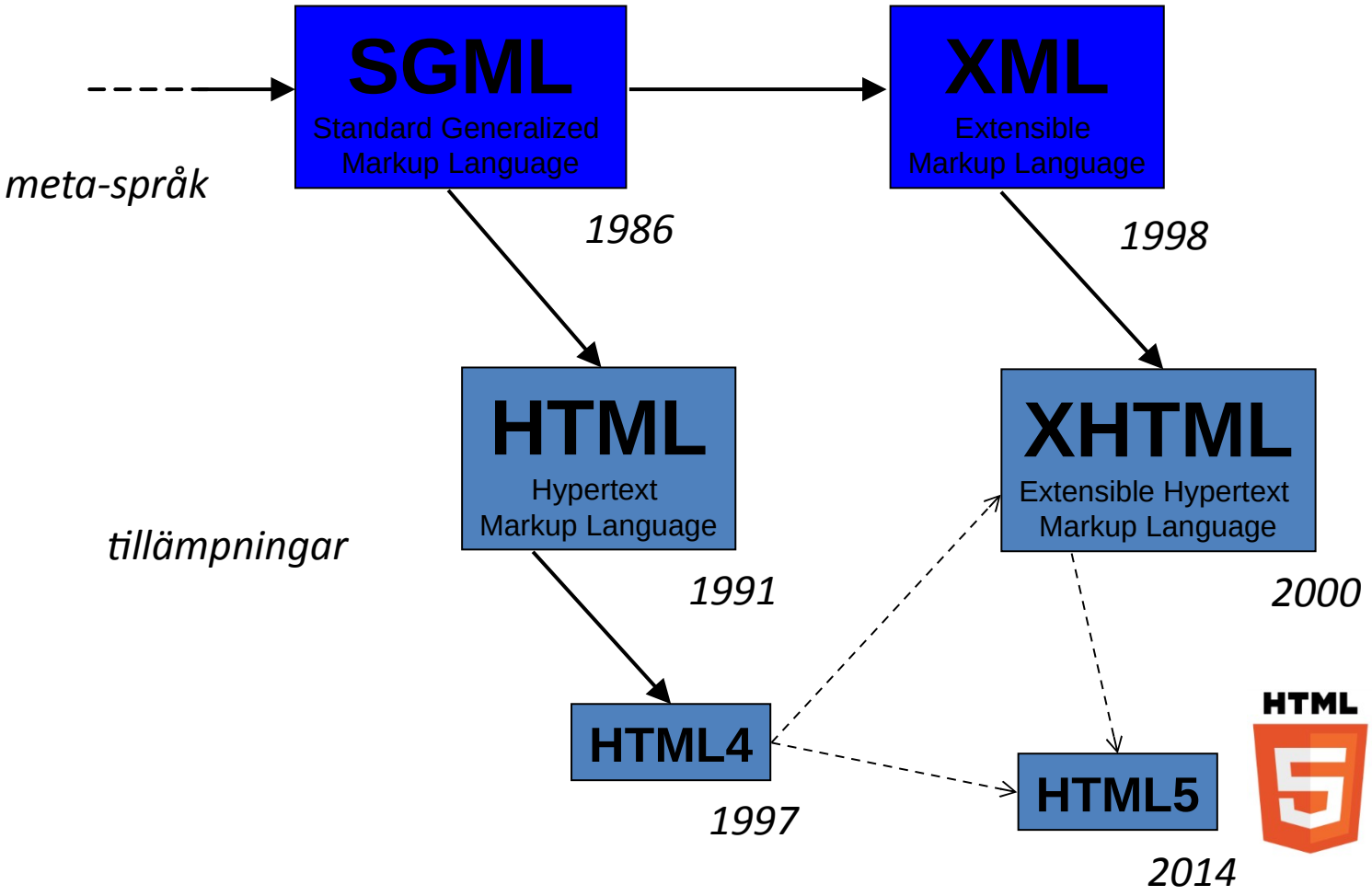
I. Down the rabbit-hole — chapter heading, 14p

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do. Once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, *"and what is the use of a book,"* thought Alice, *"without pictures or conversations?"*

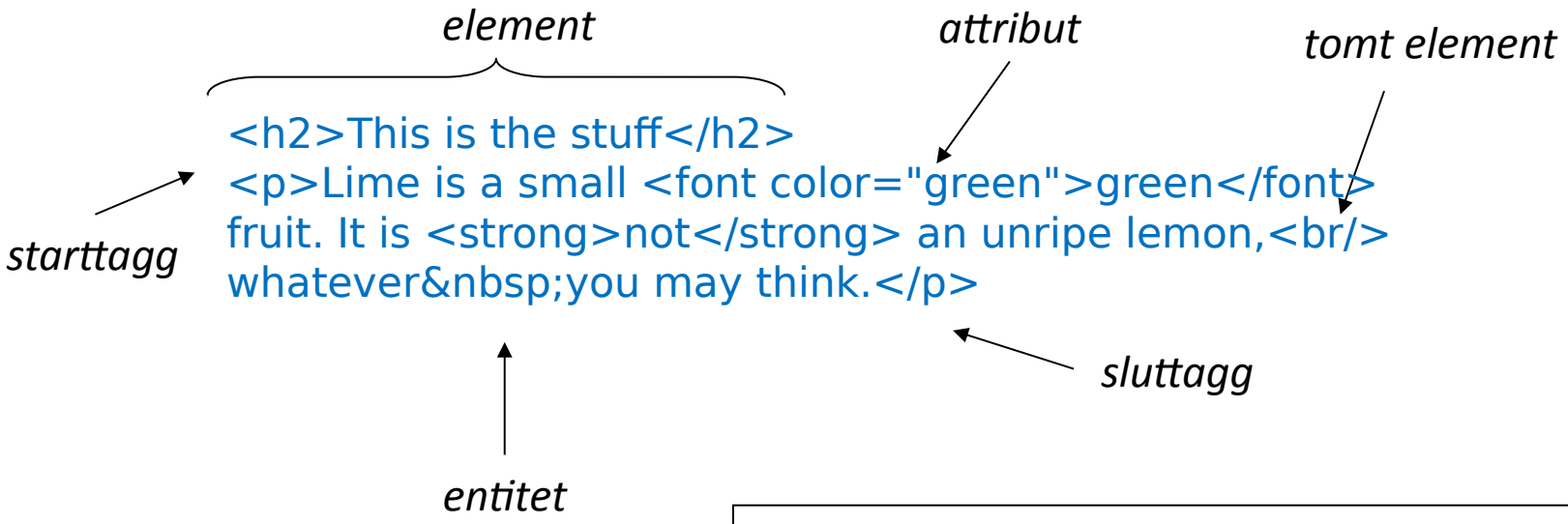
paragraph 8p

italic

Kort historisk översikt



Begrepp inom uppmärkning



This is the stuff

Lime is a small green fruit. It is **not** an unripe lemon, whatever you may think.

Exempel på XML

Well-formed
(grundnivå)

```
<?xml version="1.0"?>
<note>
  <to>Ola</to>
  <from>Peter</from>
  <heading>Öl på fredag?</heading>
  <body>
    <paragraph>Det har öppnat ett nytt holländskt ställe
    på S:t Larsgatan. De lär ha en massa sorters holländsk
    öl på fat. Vad sägs om att vi testar det på fredag
    kväll?</paragraph>
    <paragraph>Förresten, stället heter De Klomp.</paragraph>
  </body>
</note>
```

Exempel på DTD

```
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (paragraph+)>
  <!ELEMENT paragraph (#PCDATA)>
]>
```

Valid
(nästa nivå)

Detta är en DTD (Document Type Defintion), men det finns även andra sätt att specificera strukturen för XML-data. Mest populärt nuförtiden är bl.a. XML Schema.

Exempel på XML Schema


```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="note">
  <xs:complexType>
   <xs:sequence>
   <xs:element name="to" type="xs:string"/>
   <xs:element name="from" type="xs:string"/>
   <xs:element name="heading" type="xs:string"/>
   <xs:element name="body">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="paragraph" type="xs:string"
                  maxOccurs="unbounded"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Valid
(nästa nivå)

Koppla ihop XML och DTD

- *Inkludera DTD i XML-filen:*

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
 ...
]>
<note>...</note>
```

- *Hänvisa till DTD i separat fil:*

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>...</note>
```

Ytterligare exempel på DTD

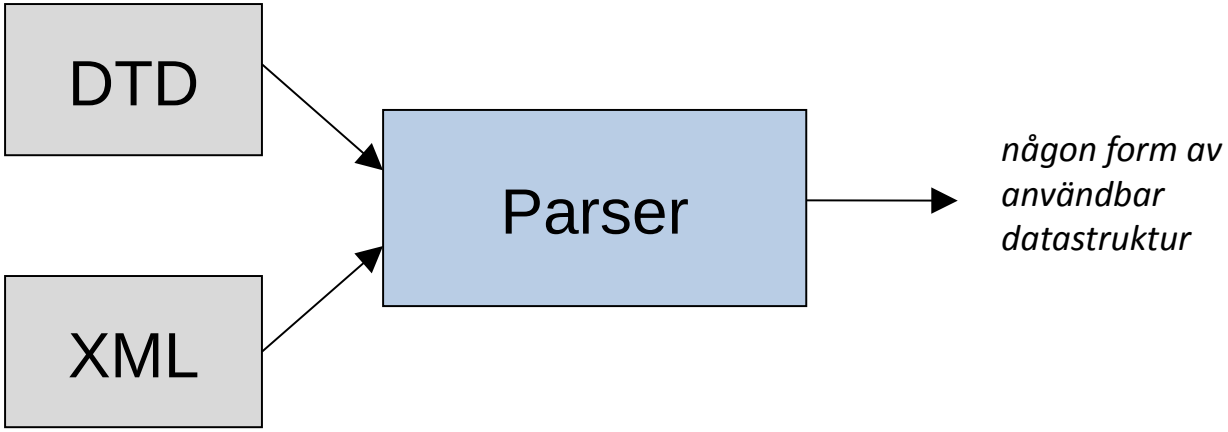
```
<!DOCTYPE RESULTS [
  <!ELEMENT RESULTS (ARTICLE+)>
  <!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
  <!ELEMENT HEADLINE (#PCDATA)>
  <!ELEMENT BYLINE (#PCDATA)>
  <!ELEMENT LEAD (#PCDATA)>
  <!ELEMENT BODY (#PCDATA)>
  <!ELEMENT NOTES (#PCDATA)>
  <!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
  <!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
  <!ATTLIST ARTICLE DATE CDATA #IMPLIED>
  <!ATTLIST ARTICLE EDITION CDATA #IMPLIED>
  <!ENTITY NEWSPAPER "Dagens Nyheter">
  <!ENTITY COPYRIGHT "Copyright 2008 Dagens Nyheter">
]>
```

Övning

- Kopiera DTD:n från föregående bild från kurswebben. Den finns i filen **newspaper.dtd** på föreläsningssidan.
- Skapa en XML-fil som inkluderar denna DTD och hitta på lite innehåll, d.v.s. fejka några nyheter.
- Validera hela filen mot **http://xmlvalidation.com/** när du är klar. (Observera att taggarna måste skrivas med versaler för att denna tjänst ska acceptera dokumentet.)
- Så här anger man teckenkodning för en XML-fil, men det ska inte behövas:
- <?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0" encoding="iso-8859-1"?>

Användning av XML-dokument

- Tolka med reguljära uttryck: *jobbigt och dåligt*
- Använda en parser: *lätt och bra*



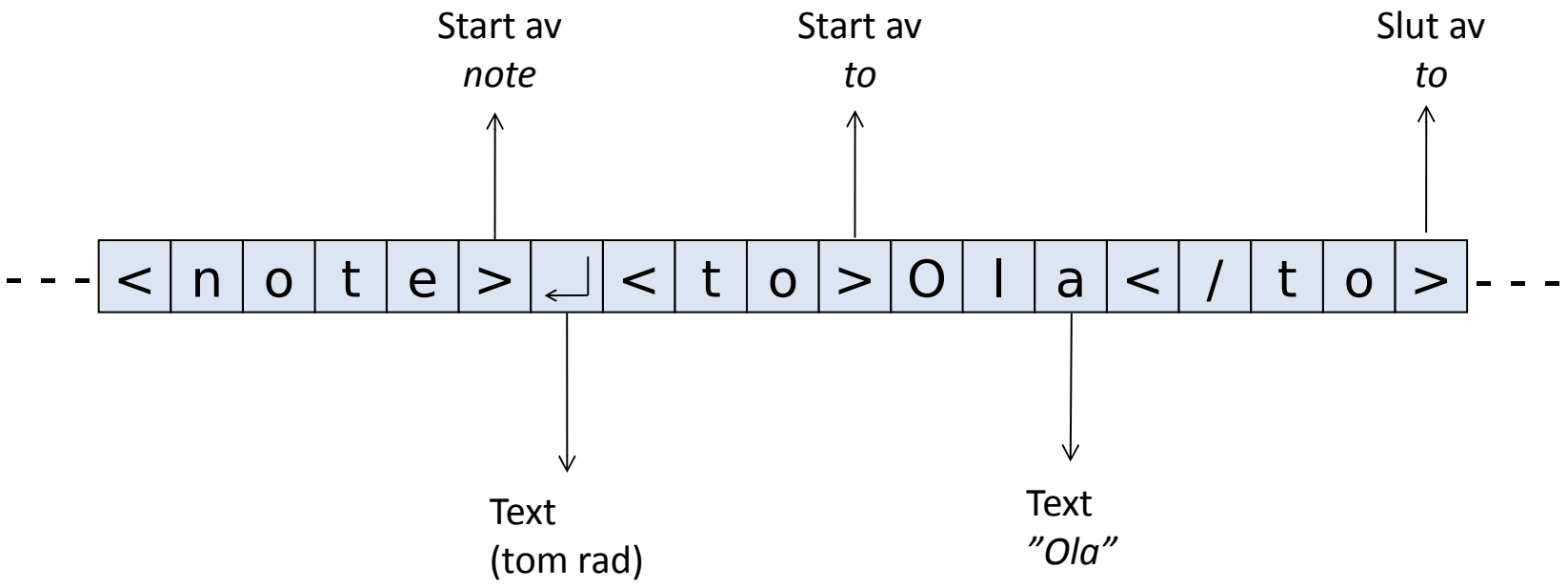
Vi ska titta på två olika sätt att parsa XML-dokument,
ett idag (strömparsning) och ett nästa föreläsning (trädparsning).

1. Strömparsning

- Läser XML-filen som en ström, från början till slut.
- Varje gång det ”händer något” (t.ex. att man träffar på en starttagg) anropas en viss funktion.
- Benämns ibland SAX (Simple API for XML) efter den ursprunliga implementationen i Java.

```
<?xml version="1.0"?>
<note>
  <to>Ola</to>
  <from>Peter</from>
  <heading>Öl på
fredag?</heading>
  ...
</note>
```

Hur strömparsningen går till



Exempel på strömparsning


```
require 'rexml/streamlistener'

class MyListener
  include REXML::StreamListener


  def tag_start(name, attrs)
    puts "Start of #{name}."
  end

  def tag_end(name)
    puts "End of #{name}."
  end

  def text(text)
    puts "Tag contains the text '#{text}'."
  end
end
```

```
>> require 'rexml/document'
=> true
>> lst = MyListener.new
=> #<MyListener:0x2f8d188>
>> src = File.new
"c:/lab/ruby/code/note2.xml"
=> #<File:c:/lab/ruby/code/note2.xml>
>>
REXML::Document.parse_stream(src,lst)
...
```

Övning

- Filen **inventory.xml** på kurswebben innehåller en lista med information om kontorsmateriel.
- Skriv en funktion **find_article** som tar en XML-fil som den ovan, samt ett kodnummer. Funktionen ska skapa en strömparser med vars hjälp den ska lokalisera och returnera namnet på den artikel som matchar kodnumret.
- Exempel:
- `>> find_article("inventory.xml", "off104")`
- `=> "Cable Ties"`
- Koden i exemplet på föregående sida finns också på kurswebben. Börja gärna med att testa den.

www.liu.se

LINKÖPINGS UNIVERSITET