Introduction to XML

Kevin S. Hawkins
k.hawkins@dho.ie

07.04.2010 • National University of Ireland, Galway

~ FUNDED UNDER PRTLI . CYCLE 4 2008 - 2011 ~
HEA Higher Education Authority An tÚdarás um Ard-Oideachas
NDP National Development Plan 2007 - 2013
EUROPEAN UNION STRUCTURAL FUNDS

Slides from this two-day workshop will be available at http://dho.ie/node/679

raster images vs. vector images

proprietary vs. non-proprietary formats

closed vs. open standards

Plain text isn’t good enough

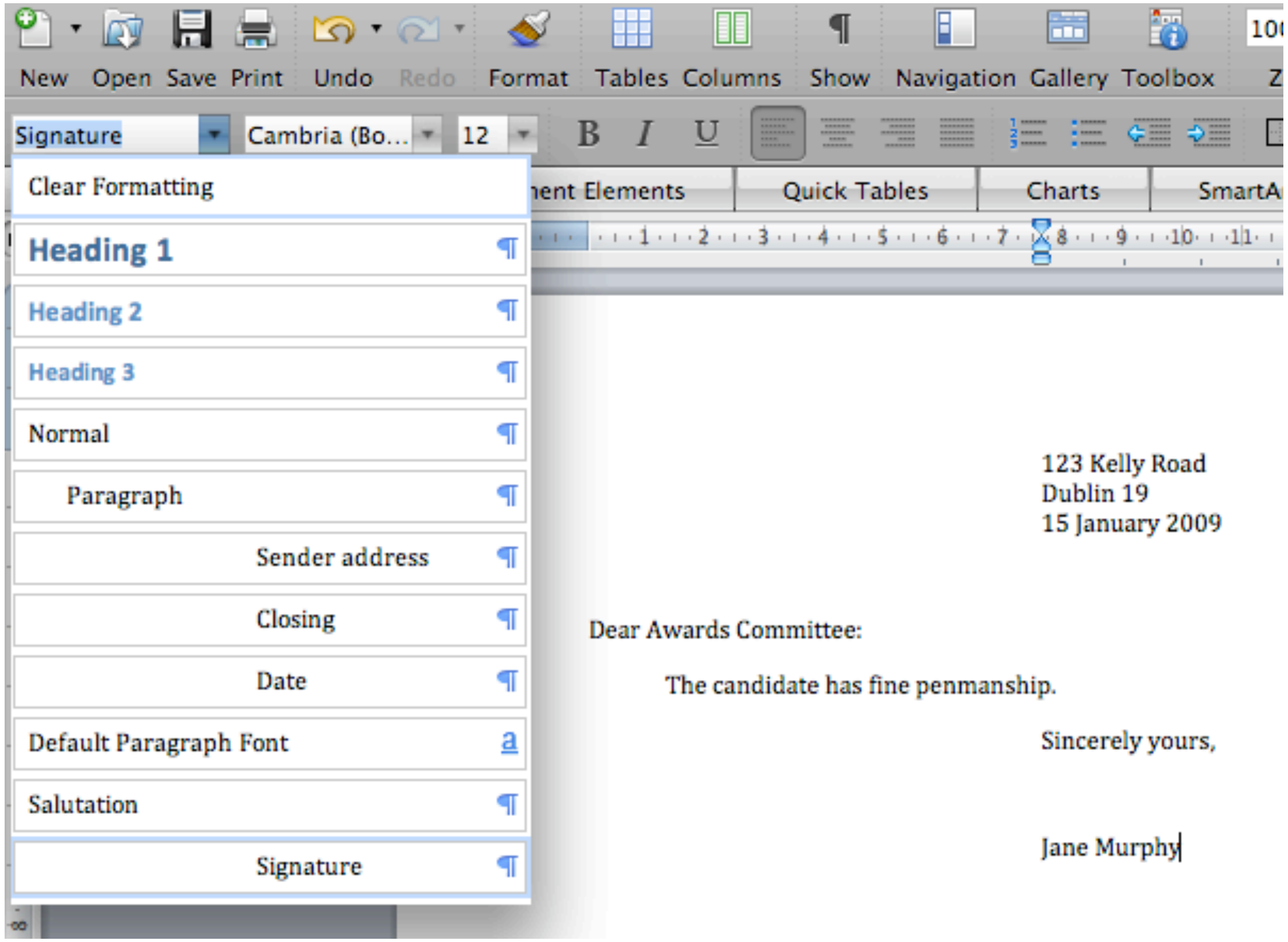
```
123 Kelly Road
Dublin 19
15 January 2009

Dear Awards Committee:

The candidate has fine penmanship.

Sincerely yours,

Jane Murphy
```

What if you want to …

- Publish a collection of letters and decide after beginning that you want to have the sender’s address and closing always right-aligned?
- Search your collection of letters to extract a list of all senders and another list of all recipients?

You need to make explicit certain features of text in order to aid the processing of that text by computer programs.

Word processor styles: create your own!



123 Kelly Road
Dublin 19
15 January 2009

Dear Awards Committee:

The candidate has fine penmanship.

Sincerely yours,

Jane Murphy

Extensible Markup Language (XML): word processor styles on steroids

- Can have one style inside another (‘nesting’)
- Can give properties to these styles, e.g.,
  - This salutation is formal.
  - This sentence contains sarcasm.
  - This word is misspelled.

XML in brief (1)

- Open, non-proprietary standard
- Stored in plain text but usually thought of as contrasting with it (as above)
- Marks beginning and ends of spans of text using tags:

  <sentence>This is a sentence.</sentence>

XML in brief (2)

- Spans of text must nest properly:

Wrong:

```
<sentence>Overlap is <emphasis>not allowed!</sentence></emphasis>
```

Right:

```
<sentence>Overlap is <emphasis>not allowed!</emphasis></sentence>
```

Elements (tags), attributes, values, content

```
<sentence type="declarative">This is a
   sentence.</sentence>

<sentence type="interrogative">Is this is a
   sentence?</sentence>
```

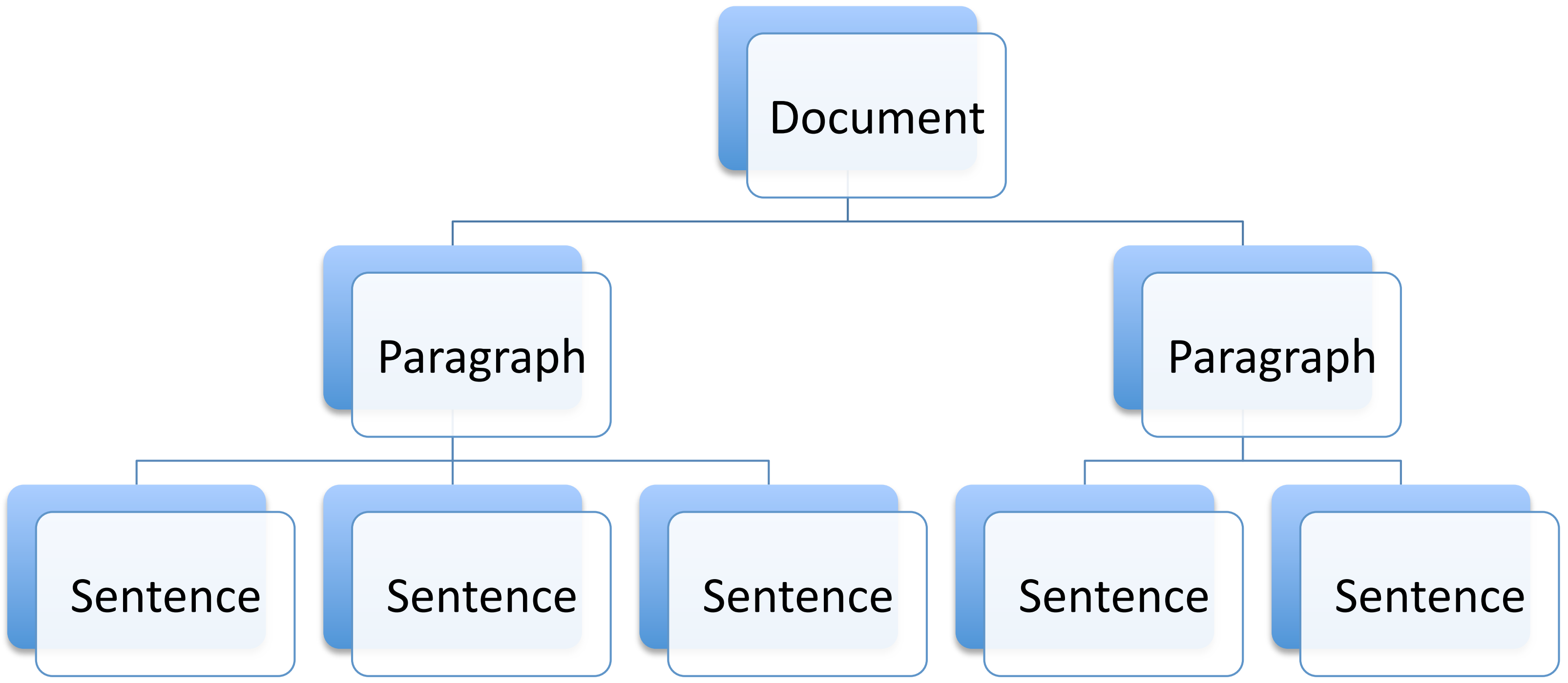
Elements may have zero, one, or more than one attribute.

Wait, this all looks a lot like HTML!

HTML is a specific implementation of XML (well, actually, its predecessor SGML) that has pre-defined elements and attributes. You can't create your own elements, so its usefulness is limited.

XML as a tree

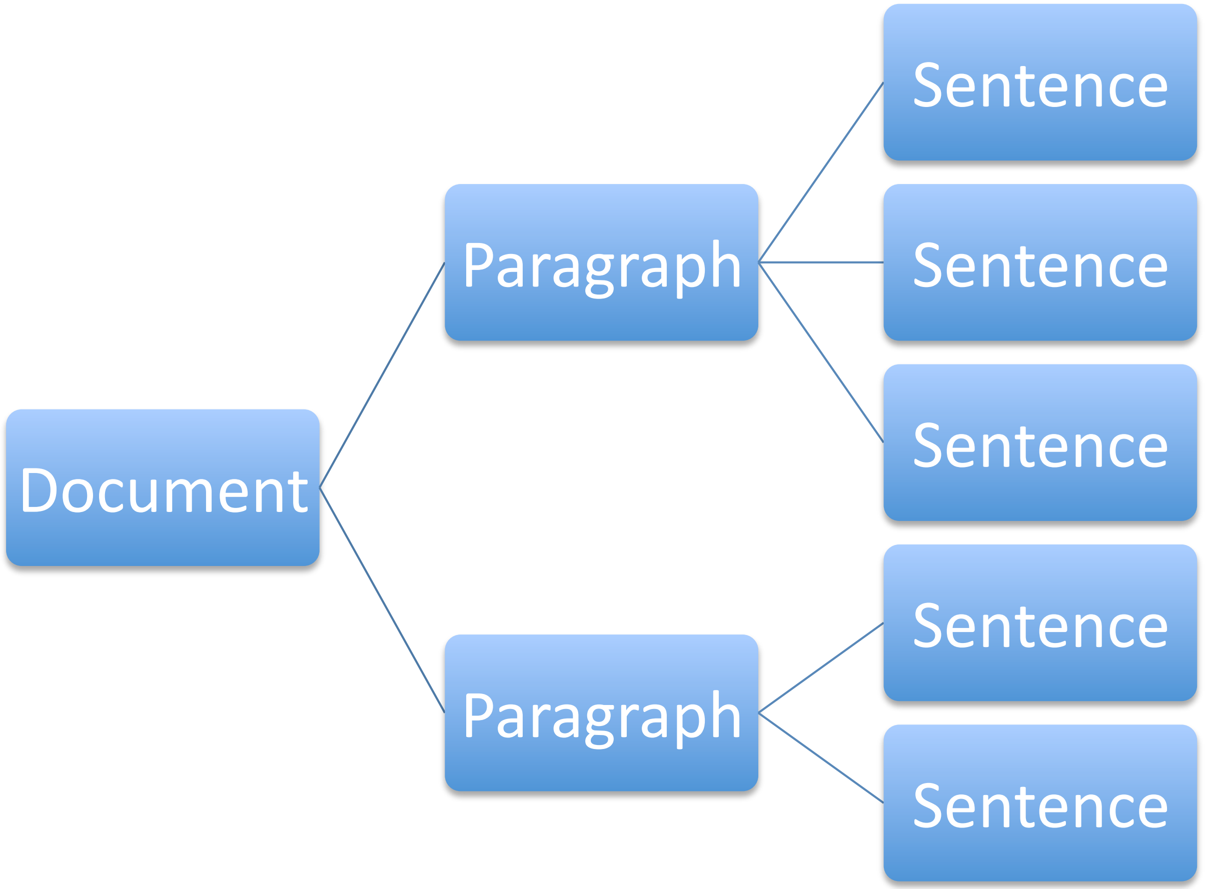
Remember, everything must nest properly!



We use family tree terms: parent, child, sibling, ancestor, and descendent.

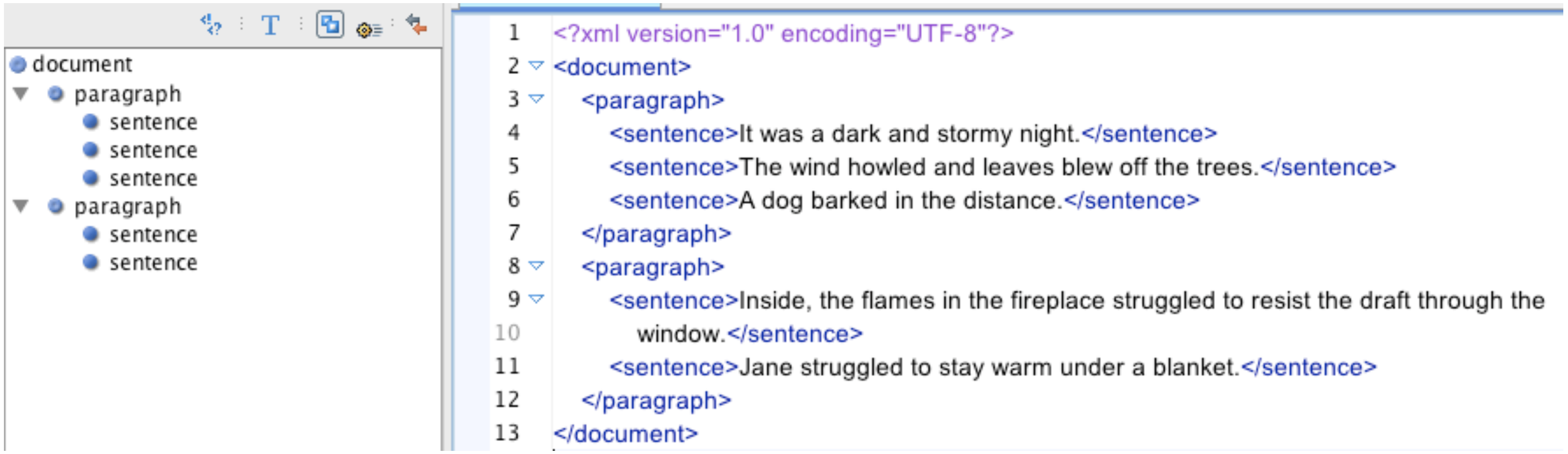
XML as a tree

Remember, everything must nest properly!



XML as a tree

Remember, everything must nest properly!



```xml
<?xml version="1.0" encoding="UTF-8"?>
<document>
   <paragraph>
      <sentence>It was a dark and stormy night.</sentence>
      <sentence>The wind howled and leaves blew off the trees.</sentence>
      <sentence>A dog barked in the distance.</sentence>
   </paragraph>
   <paragraph>
      <sentence>Inside, the flames in the fireplace struggled to resist the draft through the
         window.</sentence>
      <sentence>Jane struggled to stay warm under a blanket.</sentence>
   </paragraph>
</document>
```

Schemas (DTDs and others)

A syntax for your XML documents, specifying:

- Which elements may nest inside of others
- In what order these elements must occur
- How many times they may repeat
- What attributes they may have
- What values those attributes may have

Why would you want to constrain your document structure like this?

- Prevent errors in creating the XML
- Make it easier to search the text

  Remember we were going to extract names of senders and recipients? You know where to expect to find them within your XML documents.

Structure, not appearance

Most people use XML to describe the structure of a document rather than its appearance. Information about how to render various components of the document is usually stored separately, in a *stylesheet*.

Brilliant. But how do we keep from reinventing the wheel in determining good ways to constrain our document structure? And wouldn't it be good to make sure we use the same vocabulary of element and attribute names as our colleagues so that we can use each other's texts?

Use an existing schema!

TEI

TEXT ENCODING INITIATIVE

Questions?