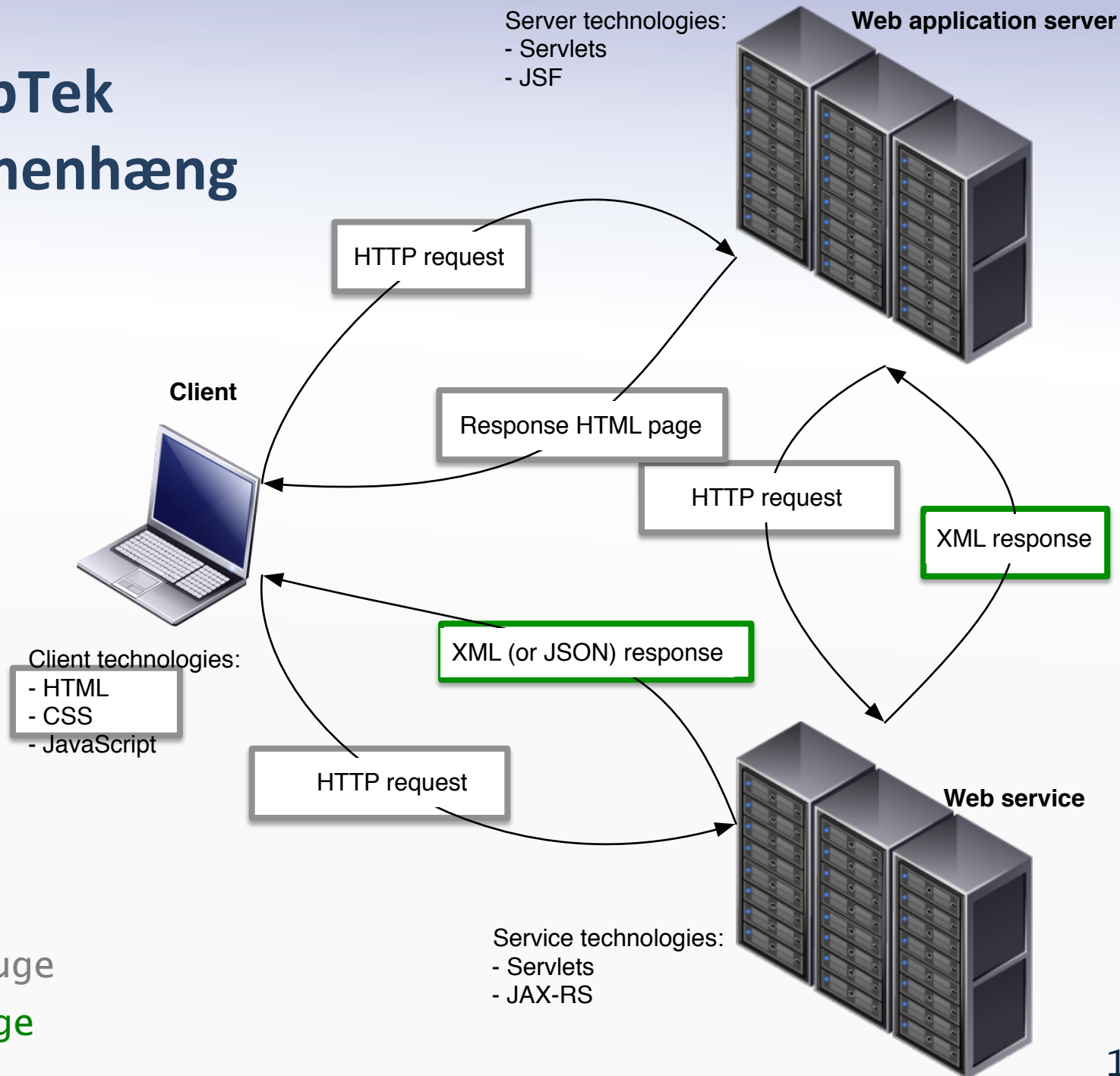


dWebTek sammenhæng



sidste uge

Denne uge

XML and Namespaces

Motivation

- We need a way to represent data
 - For storage
 - For transfer between machines
- We want it to be
 - Readable
 - Supported in many languages and on many platforms
- We want to be able to create our own formats.

What is XML?

- XML: *Extensible Markup Language*
- A **framework** for defining markup languages
- Each language is targeted at its own **application domain** with its own markup tags
- There is a common set of **generic tools** for processing XML documents
- **XHTML**: an XML variant of HTML
- Inherently **internationalized** and **platform independent** ([Unicode](#))
- Developed by W3C, standardized in 1998

Overview

- **Recipe ML**
- XML trees
- Textual representation
- Examples of XML languages
- Namespaces

Recipes in XML

- Define our own “***Recipe Markup Language***”
- Choose markup tags that correspond to concepts in this application domain
 - *recipe, ingredient, amount, ...*
- No canonical choices
 - granularity of markup?
 - structuring?
 - elements or attributes?
 - ...

Example (1/2)

```
<collection>
  <description>Recipes suggested by Jane Dow</description>

  <recipe id="r117">
    <title>Rhubarb Cobbler</title>
    <date>Wed, 14 Jun 95</date>

    <ingredient name="diced rhubarb" amount="2.5" unit="cup"/>
    <ingredient name="sugar" amount="2" unit="tablespoon"/>
    <ingredient name="fairly ripe banana" amount="2"/>
    <ingredient name="cinnamon" amount="0.25" unit="teaspoon"/>
    <ingredient name="nutmeg" amount="1" unit="dash"/>

    <preparation>
      <step>
        Combine all and use as cobbler, pie, or crisp.
      </step>
    </preparation>
```

Example (2/2)

```
<comment>
  Rhubarb Cobbler made with bananas as the main sweetener.
  It was delicious.
</comment>

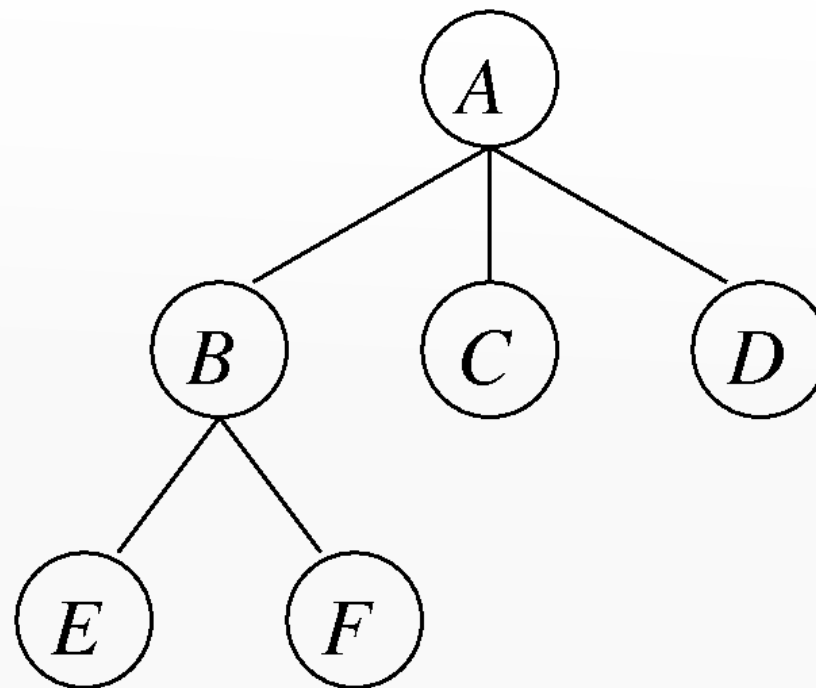
<nutrition calories="170" fat="28%"
           carbohydrates="58%" protein="14%"/>
  <related ref="42">Garden Quiche is also yummy</related>
</recipe>
</collection>
```


Overview

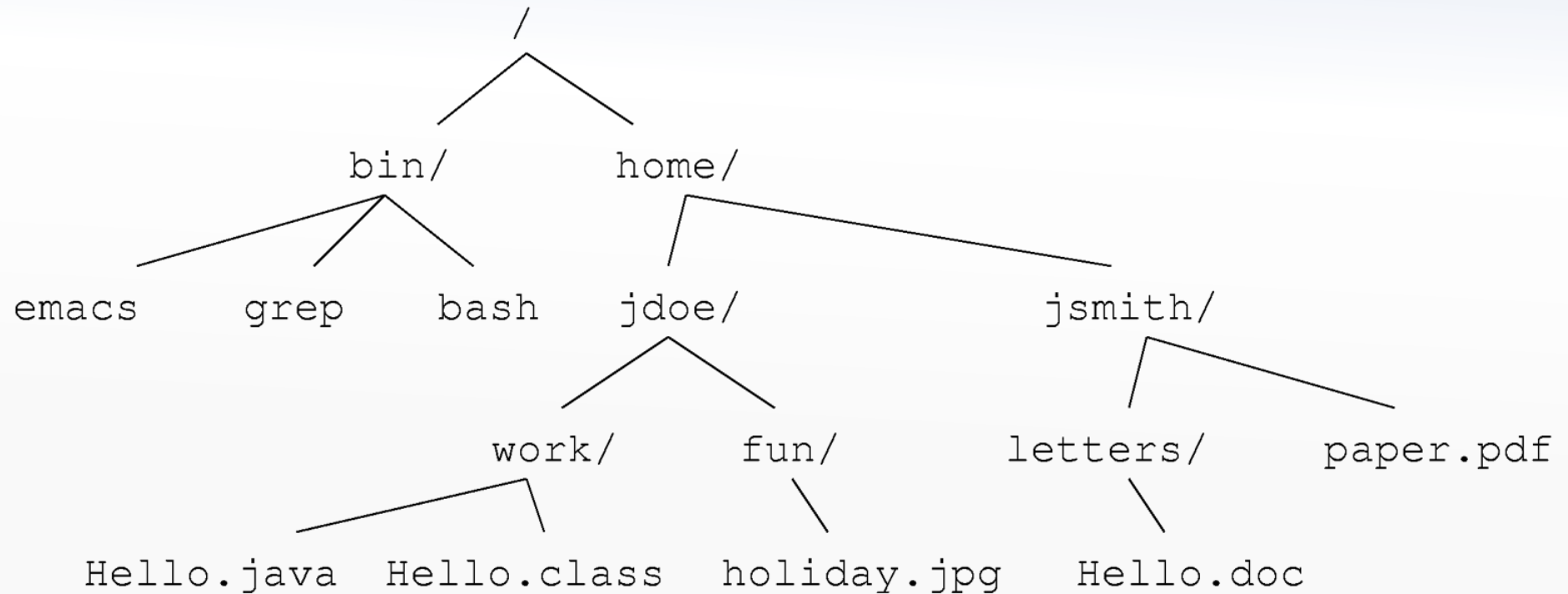
- Recipe ML
- **XML trees**
- Textual representation
- Examples of XML languages
- Namespaces

XML Trees

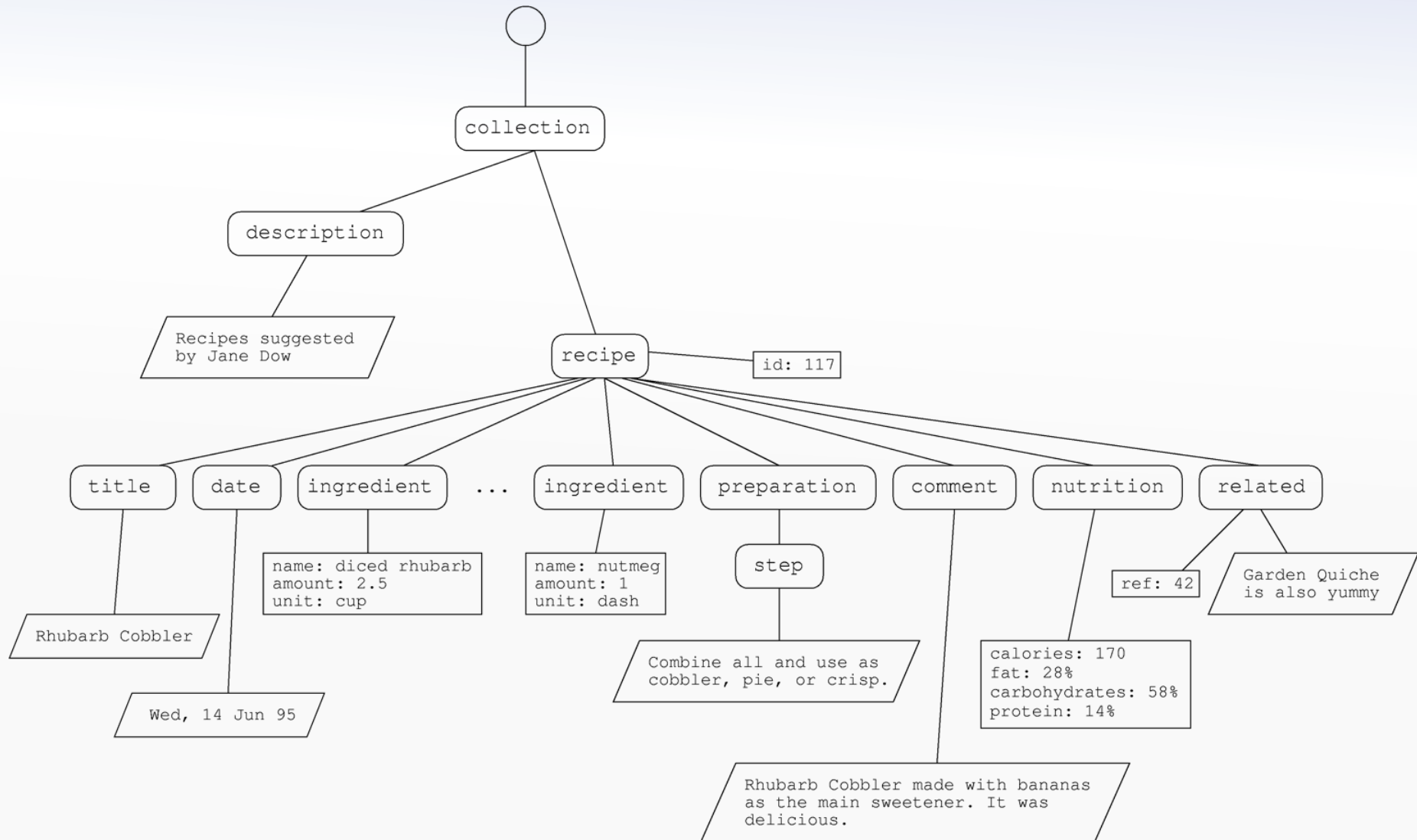
- Conceptually, an XML document is a **tree structure**
 - node, edge
 - root, leaf
 - child, parent
 - sibling (ordered), ancestor, descendant



An Analogy: File Systems



Tree View of the XML Recipes



Nodes in XML Trees

- **Text nodes:** carry the actual contents, leaf nodes
- **Element nodes:** define hierarchical logical groupings of contents, each have a *name*
- **Attribute nodes:** unordered, each associated with an element node, has a *name* and a *value*
- **Comment nodes:** ignorable meta-information
- **Processing instructions:** instructions to specific processors, each have a *target* and a *value*
- **Root nodes:** every XML tree has one root node that represents the entire tree

More Constructs

- XML declaration
- Character references
- CDATA sections
- Document type declarations and entity references explained later...
- Whitespace?

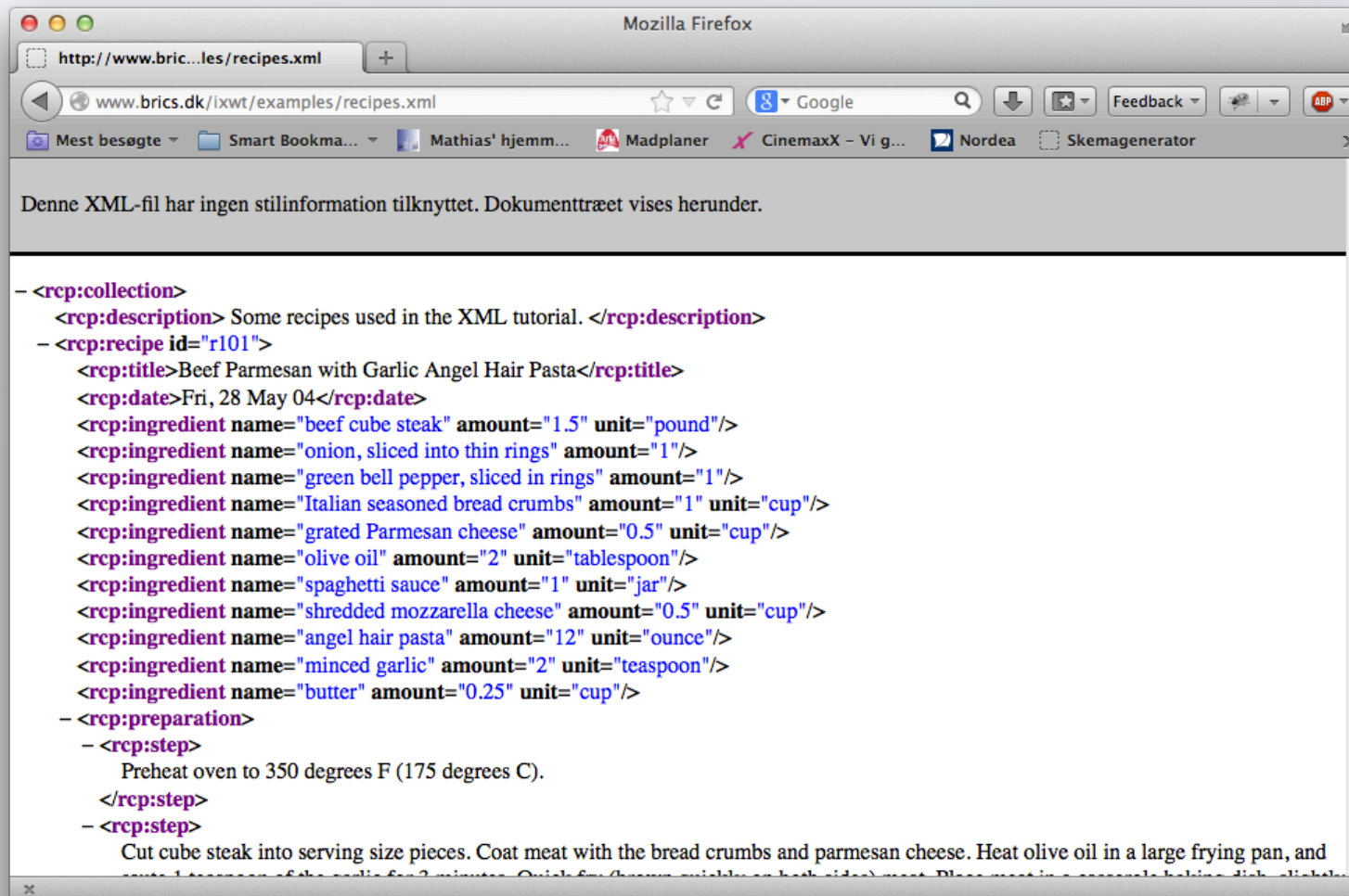
Overview

- Recipe ML
- XML trees
- **Textual representation**
- Examples of XML languages
- Namespaces

Textual Representation

- **Text nodes:** written as the text they carry
- **Element nodes:** start-end tags
 - `<bla ...> ... </bla>`
 - short-hand notation for empty elements: `<bla/>`
- **Attribute nodes:** `name="value"` in start tags
- **Comment nodes:** `<!-- bla -->`
- **Processing instructions:** `<?target value?>`
- **Root nodes:** implicit

Browsing XML



Example

```
<?xml version="1.1" encoding="ISO-8859-1"?>
<!DOCTYPE features SYSTEM "example.dtd">
<features a="b">
  <?mytool here is some information specific to mytool?>
  El señor está bien, garçon!
  Copyright &#169; 2005
  <![CDATA[ <this is not a tag> ]]>
  <!-- always remember to specify the
        right character encoding -->
</features>
```

Well-formedness

- Every XML document must be ***well-formed***
 - start and end tags must **match** and **nest** properly
 - ~~<x><y></y></x>~~ ✓
 - ~~</z><x><y></x></y>~~
 - exactly one **root element**
 - ...
- in other words, it defines a proper tree structure
- **XML parser**: given the textual XML document, constructs its tree representation

Alternatives?

S-expressions, 1958:

```
(collection  
  (recipe  
    (title "Rhubarb Cobbler") (date "Wed, 14 Jun 95")  
    ...  
  )  
)
```

You will see this in
dProgSprog

JSON (1990's/2000's):

```
{"recipes": [{"title": "Rhubarb Cobbler",  
              "date": "Wed, 14 Jun 95"},  
             ...]  
}
```

You will see this in
week 4 of this course

Clicker question

- Is this XML doc well-formed?

`<a><a>`

- No, it has two root elements
- No, "b" is not closed
- No, "a" is not closed
- Yes



Overview

- Recipe ML
- XML trees
- Textual representation
- **Examples of XML languages**
- Namespaces

Example: XHTML

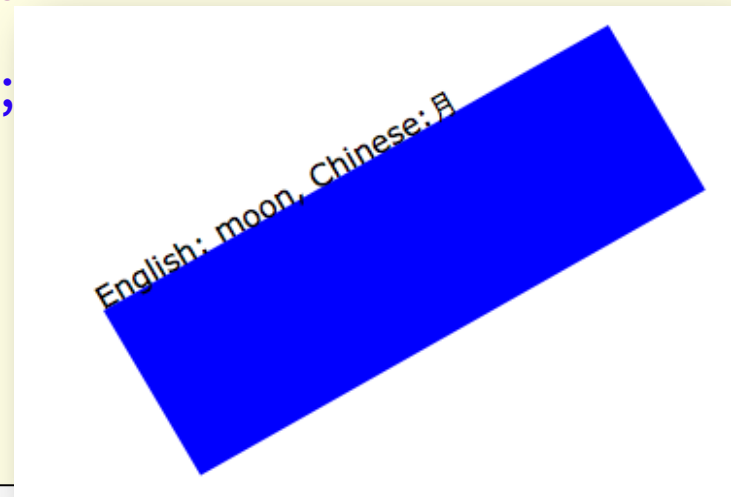
```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>Hello world!</title></head>
  <body>
    <h1>This is a heading</h1>
    This is some text.
  </body>
</html>
```

Example: SVG

```
<svg xmlns="http://www.w3.org/2000/svg">
<title>Chinese-English Unicode Table</title>

<g transform="translate(100,200)">
  <g transform="rotate(-30)">
    <rect width="300" height="100"
          style="fill:rgb(0,0,255);stroke-width:1"/>

    <text id="TextElement" x="0" y="0" style="font-
family:Verdana;font-size:24">
      English: moon, Chinese:&#x6708;
    </text>
  </g>
</g>
</svg>
```



Example: Ant

```
<?xml version="1.0"?>
<project basedir="." default="run">
  <path id="classpath">
    <pathelement path="classes" />
  </path>
  <target name="compile" description="Compile all Java source files">
    <echo message="Compiling the schema..." />
    <mkdir dir="classes" />
    <javac destdir="classes" debug="on">
      <src path="src" />
    </javac>
  </target>

  <target name="run" depends="compile" description="Run app">
    <echo message="Running the sample application..." />
    <java classname="Main" fork="true">
      <classpath refid="classpath" />
    </java>
  </target>
</project>
```

Overview

- Recipe ML
- XML trees
- Textual representation
- Examples of XML languages
- **Namespaces**

XML Namespaces

```
<widget type="gadget">
  <head size="medium"/>
  <big><subwidget ref="gizmo"/></big>
  <info>
    <head>
      <title>Description of gadget</title>
    </head>
    <body>
      <h1>Gadget</h1>
      A gadget contains a big gizmo
    </body>
  </info>
</widget>
```

- When combining languages, element names may become **ambiguous!**
- Common problems call for common solutions

The Idea

- Assign a URI to every (sub-)language

e.g. `http://www.w3.org/1999/xhtml`
for XHTML 1.0

- Qualify element names with URIs:

`{http://www.w3.org/1999/xhtml}head`

The Actual Solution

- *Namespace declarations* bind URIs to *prefixes*

```
<... xmlns:foo="http://www.w3.org/TR/xhtml1">  
  ...  
  <foo:head>...</foo:head>  
  ...  
</...>
```

- Lexical scope
- Default namespace (no prefix) declared with `xmlns="..."`
- Attribute names can also be prefixed

Widgets with Namespaces

```
<widget type="gadget" xmlns="http://www.widget.inc">
  <head size="medium"/>
  <big><subwidget ref="gizmo"/></big>
  <info xmlns:xhtml="http://www.w3.org/TR/xhtml1">
    <xhtml:head>
      <xhtml:title>Description of gadget</xhtml:title>
    </xhtml:head>
    <xhtml:body>
      <xhtml:h1>Gadget</xhtml:h1>
      A gadget contains a big gizmo
    </xhtml:body>
  </info>
</widget>
```

Namespace map: for each element, maps prefixes to URIs

Clicker question

```
<xyz xmlns="http://baz.net" xmlns:foo="http://www.foo.org">  
  <abc xmlns="http://www.bar.com">  
    <foo xmlns:dwebtek="http://www.cs.au.dk/dwebTek"/>  
  </abc>  
</xyz>
```



what is the namespace of foo?

1. http://www.cs.au.dk/dwebTek
 2. http://www.bar.com
 3. http://www.foo.org
 4. http://baz.net
- 

Clicker question

```
<xyz xmlns="http://baz.net" xmlns:foo="http://www.foo.org">  
  <abc xmlns="http://www.bar.com">  
    <foo:foo xmlns:dwebtek="http://www.cs.au.dk/dwebTek"/>  
  </abc>  
</xyz>
```



what is the namespace of foo?

1. http://www.cs.au.dk/dwebTek
2. http://www.bar.com
3. http://www.foo.org ✓
4. http://baz.net

Summary

- Recipe ML
- XML trees
- Textual representation
- Examples of XML languages
- Namespaces

Essential Online Resources

- <http://www.w3.org/TR/xml11/>
- <http://www.w3.org/TR/xml-names11>
- <http://www.unicode.org/>