Document Object Model (DOM):

An Abstract Data Structure for XML data

Alex Dekhtyar

Department of Computer Science
University of Kentucky

About

- Jobs for B.S. graduates often require work with XML
- Teach XML in undergraduate curriculum
  - Databases
  - Web Programming
  - Data Structures

Outline

- XML Syntax
- XML as a tree
- Document Object Model (DOM)
- DOM API

XML

Meta-language for encoding information

Information, content

XML elements

Opening tags: <name>
Closing tags: <name/>
Emply elements: <p/>

ordered

```
<faculty>
    <name>
        <first>  Alexander </first>
        <middle> M.        </middle>
        <last>   Dekhtyar  </last>
    </name>
    <dept> Computer Science </dept>
    <course>
      <sem> Spring 2007  </sem>
        <code> CS405     </code>
        <title> Database systems
                                </title>
    </course>
</faculty>
```

XML

Meta-language for encoding information

Information, content

XML elements

Opening tags: <name>
Closing tags: <name/>
Emply elements: <p/>

ordered

Attributes

Additional information
unordered

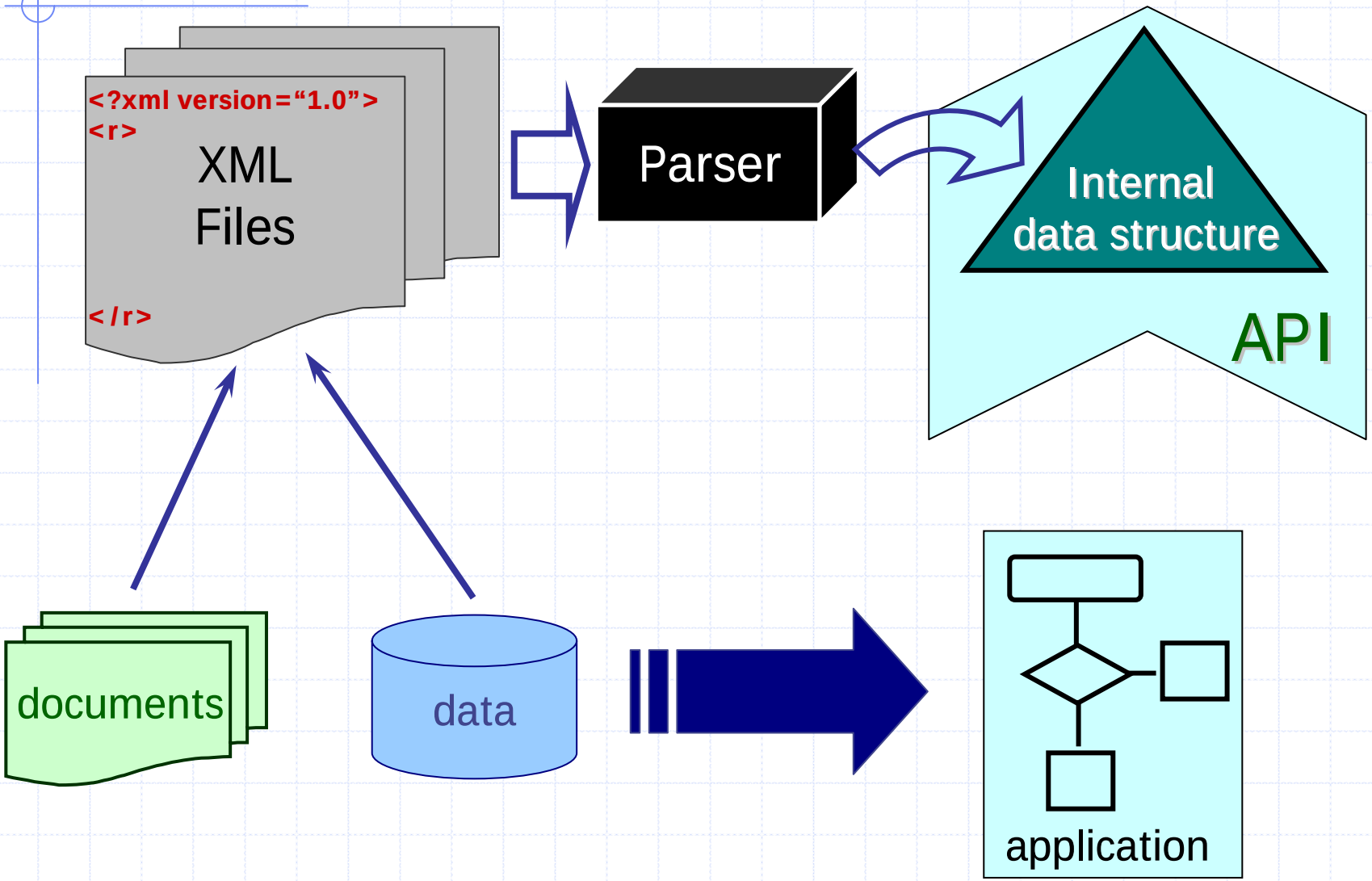
```
<faculty  id =“27” >
    <name origin=“Ukraine”>
          <first>   Alexander </first>
        <middle> M.          </middle>
          <last>     Dekhtyar </last>
     </name>
     <dept> Computer Science </dept>
     <course>
       <sem> Spring 2007  </sem>
         <code> CS405      </code>
         <title> Database systems
                                  </title>
     </course>
 </faculty>
```

XML

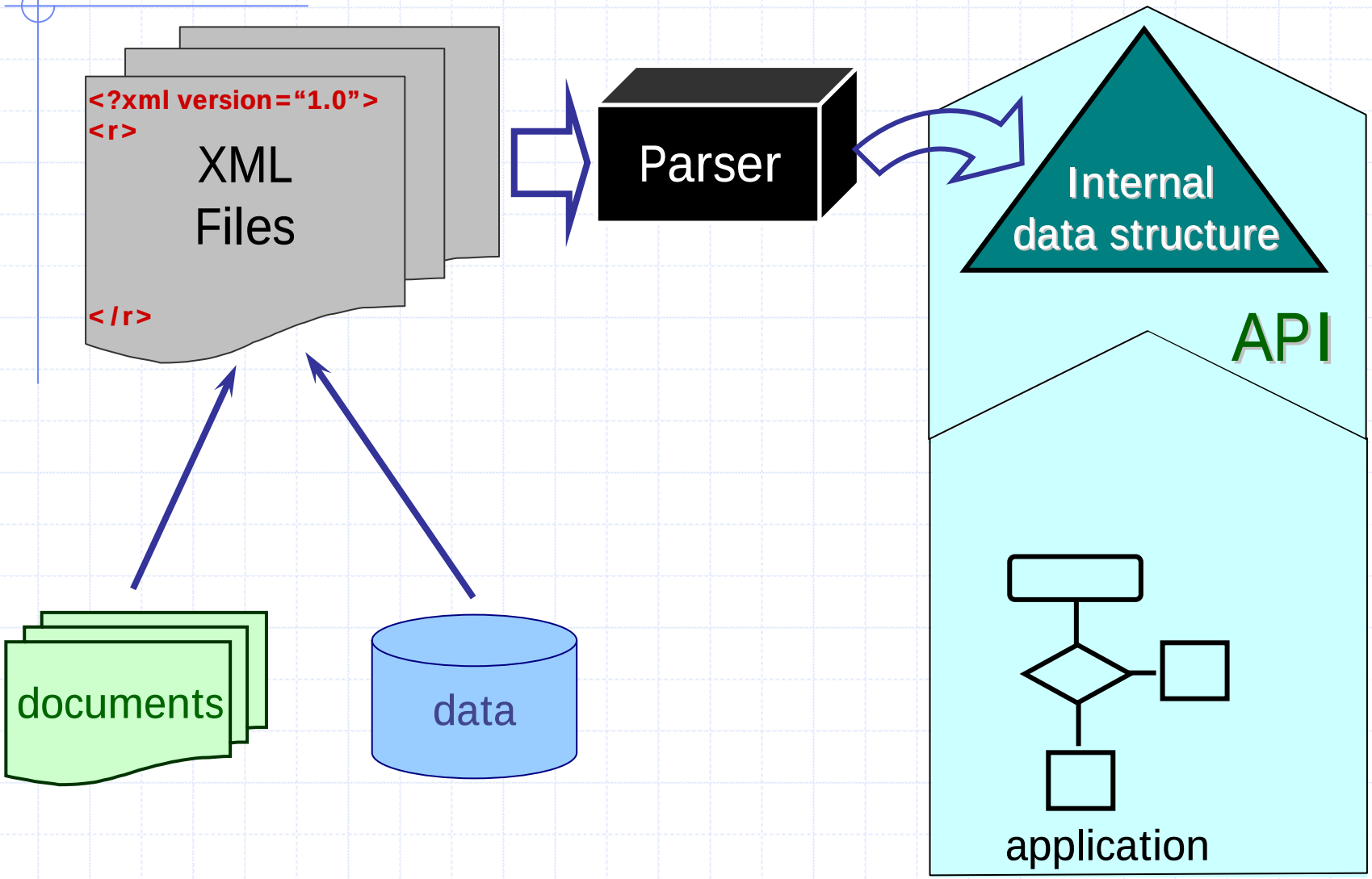
- Structure
  - nested elements
- Well-formed XML
  - correctly nested elements
- World Wide Web Consortium (W3C)
  - recommendation
  - XML 1.0, 1998

```
<faculty  id =“27” >
    <name origin=“Ukraine”>
          <first>   Alexander </first>
        <middle> M.          </middle>
        <last>     Dekhtyar </last>
    </name>
    <dept> Computer Science </dept>
    <course>
      <sem> Spring 2007  </sem>
        <code> CS405      </code>
        <title> Database systems
                                  </title>
    </course>
</faculty>
```

XML and applications



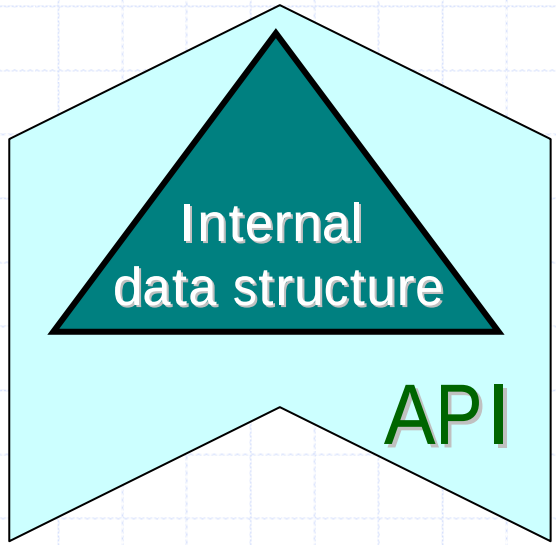
XML and applications



XML and applications

Document Object Model (DOM)

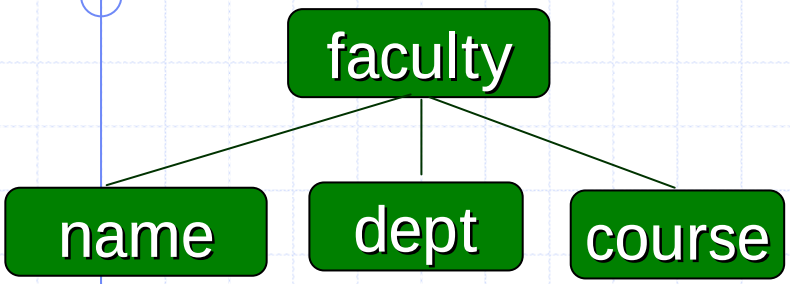
- W3C standard
- www.w3c.org/DOM



DOM Level 1 (core) – for XML

DOME Level 1 (HTML) – for HTML

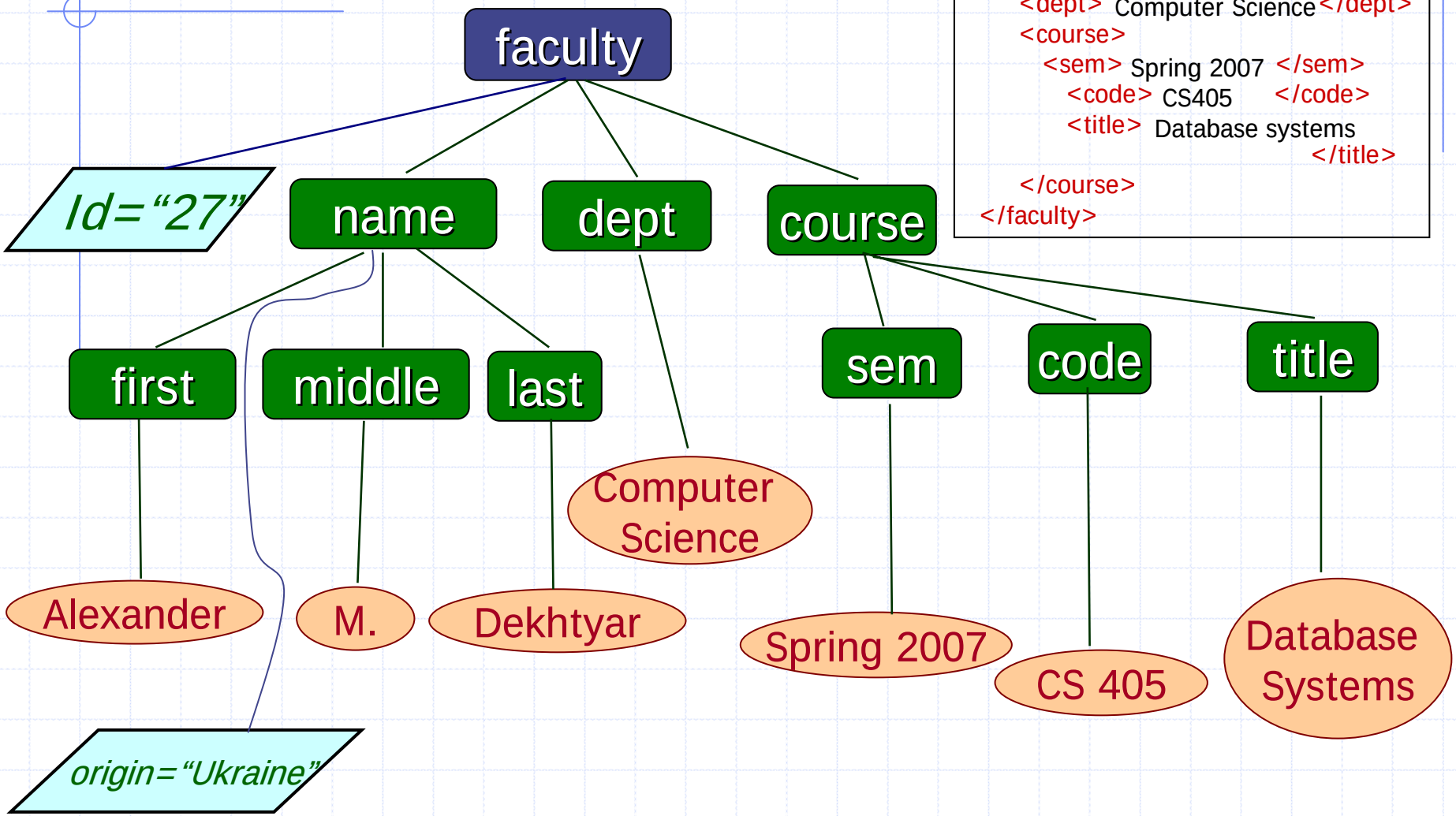
Trees for XML




```
<faculty  id =“27” >
    <name origin=“Ukraine”>
        <first>   Alexander </first>
        <middle> M.         </middle>
        <last>    Dekhtyar </last>
    </name>
    <dept> Computer Science </dept>
    <course>
      <sem> Spring 2007  </sem>
        <code> CS405     </code>
        <title> Database systems
                                  </title>
    </course>
</faculty>
```

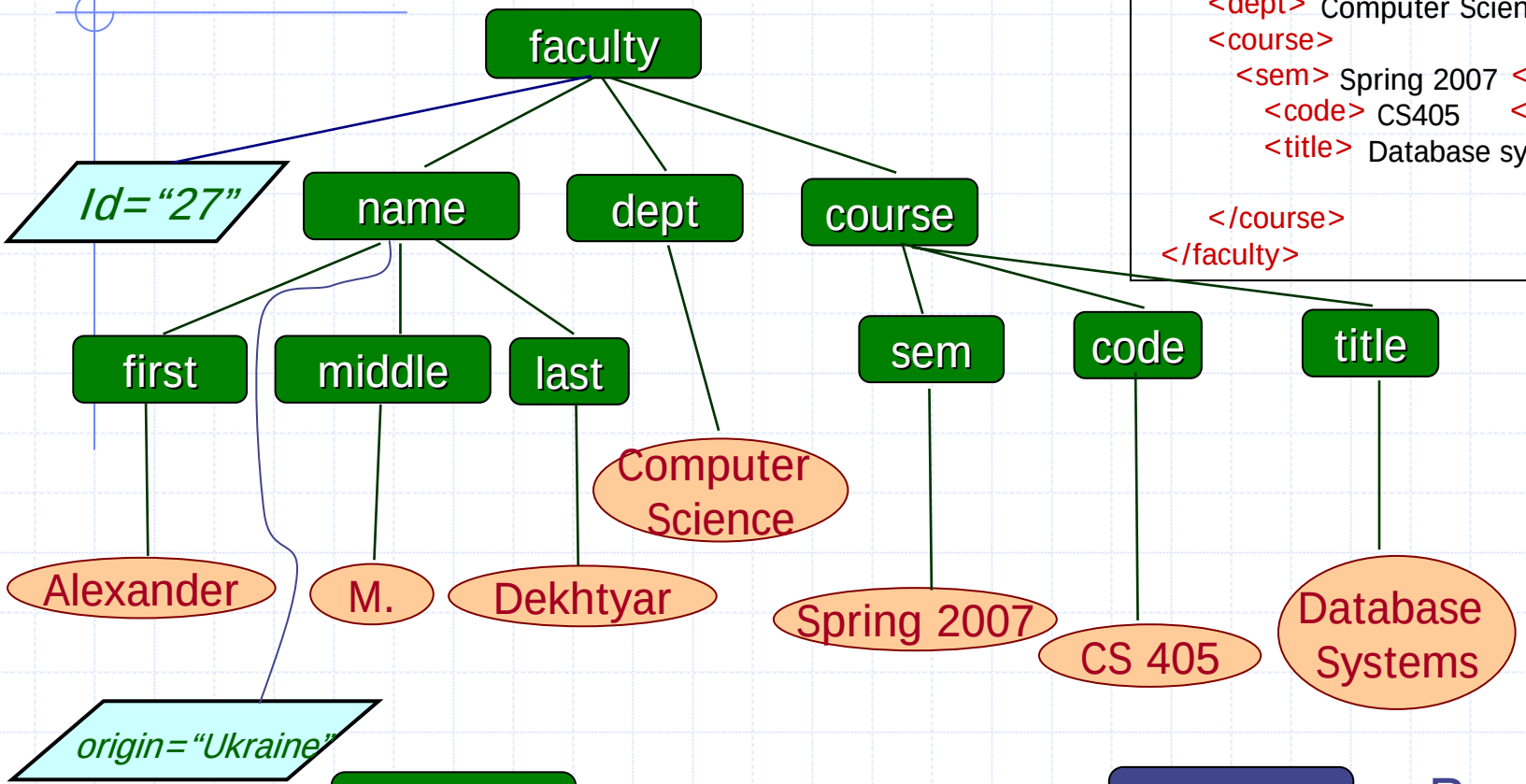
Trees for XML

```
<faculty  id =“27” >
   <name origin=“Ukraine”>
        <first>  Alexander</first>
      <middle> M.        </middle>
      <last>     Dekhtyar</last>
   </name>
   <dept> Computer Science</dept>
   <course>
     <sem> Spring 2007 </sem>
       <code> CS405     </code>
       <title> Database systems
                                </title>
   </course>
</faculty>
```



Trees for XML

```
<faculty  id =“27” >
   <name origin=“Ukraine”>
        <first>  Alexander</first>
      <middle>  M.        </middle>
      <last>    Dekhtyar</last>
   </name>
   <dept>  Computer Science</dept>
   <course>
     <sem>  Spring 2007  </sem>
       <code>  CS405     </code>
       <title>  Database systems
                                </title>
   </course>
</faculty>
```



name - Element node

faculty - Root node

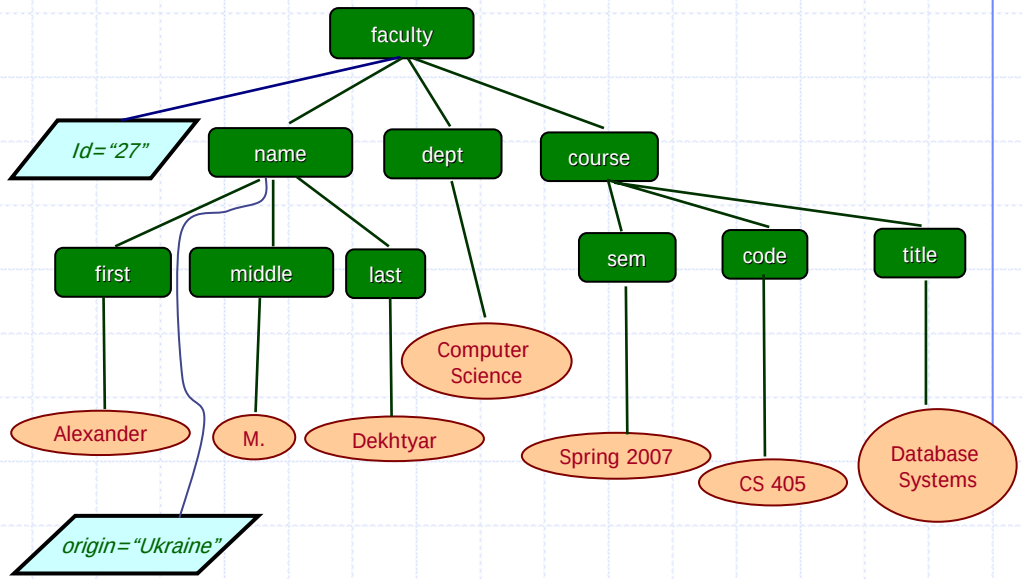
Computer Science - Text node

Id=“27” - Attribute node

Document Object Model

- Abstract Data Type
- Object-oriented
- System of types/interfaces
  - ★ Attributes
  - ★ Methods

Functionality

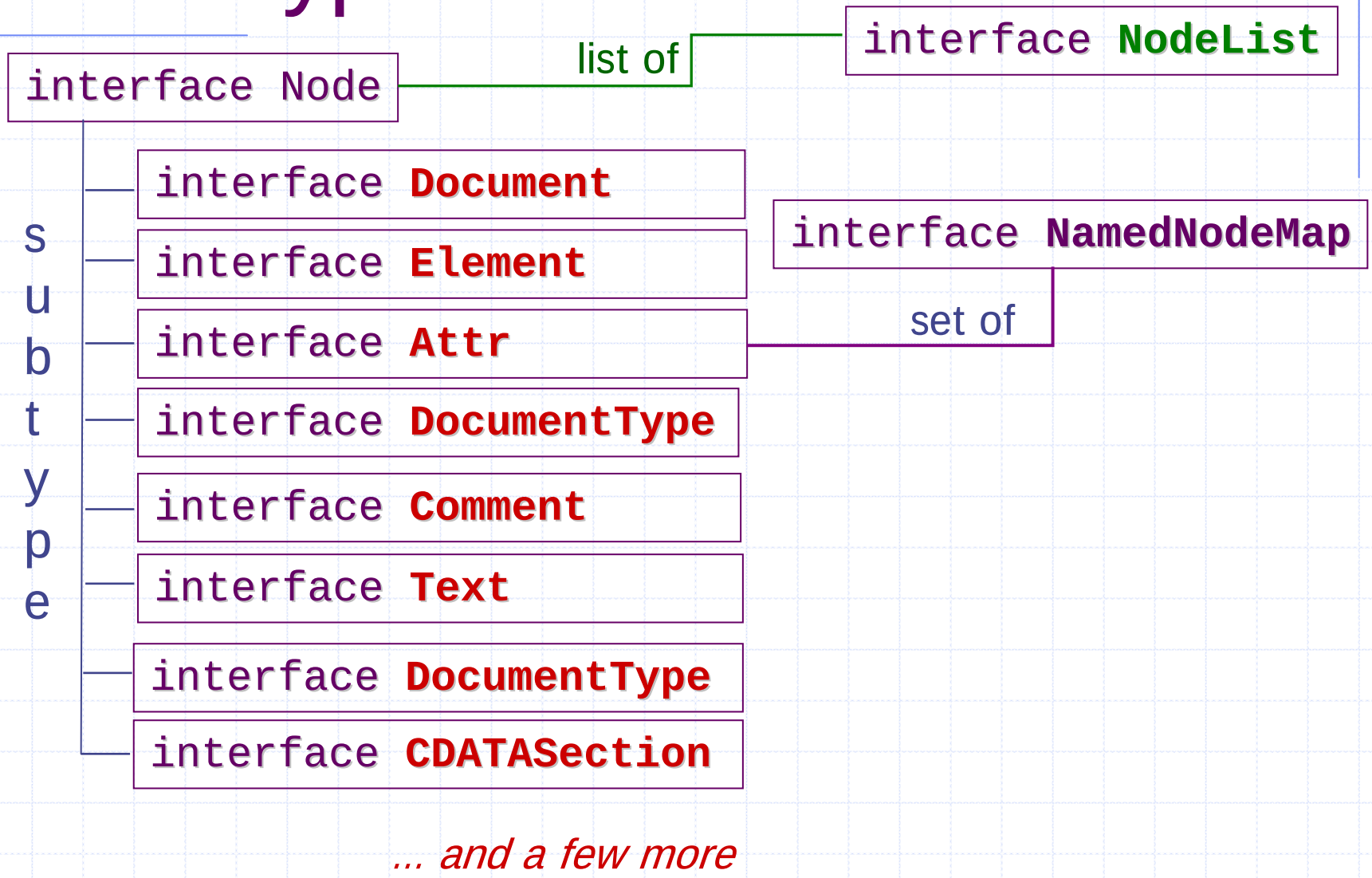


Creation of nodes

Insertion of nodes in into the DOM Tree

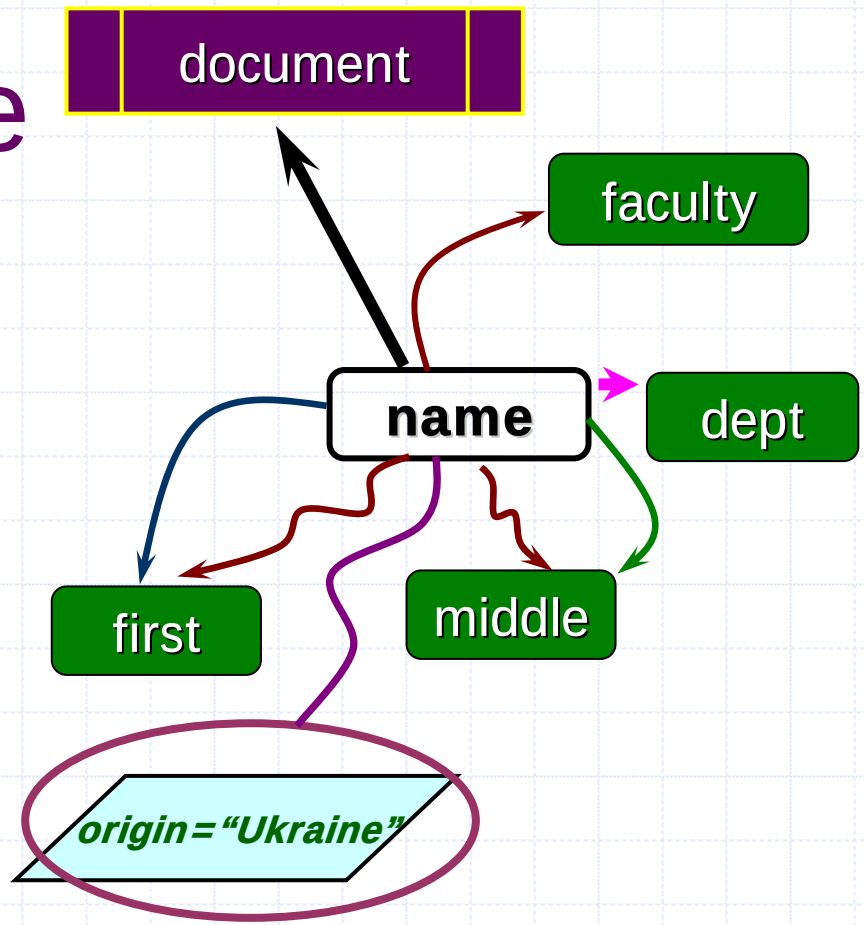
Traversal of the DOM Tree

DOM Type Structure

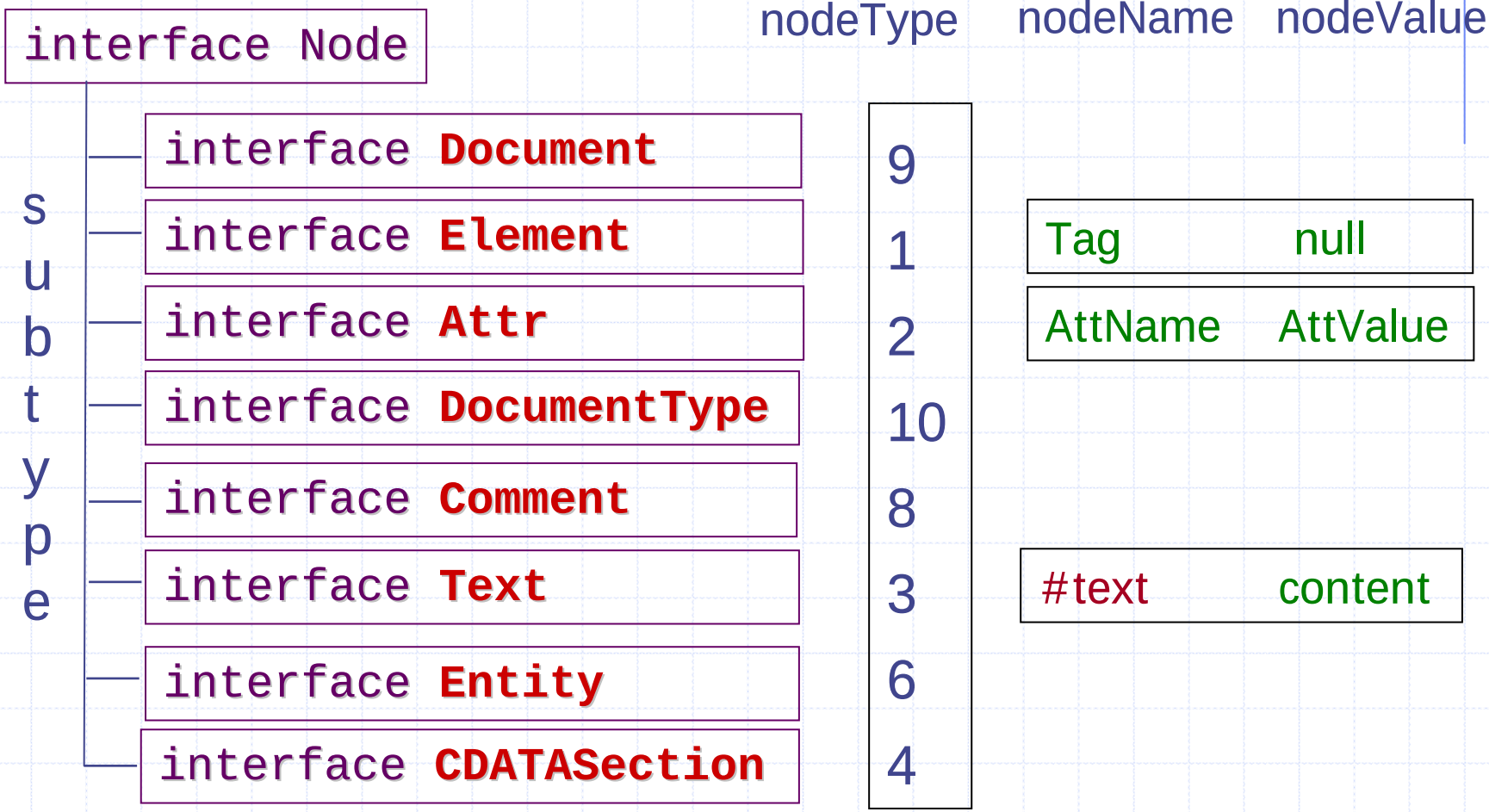


interface Node

DOMString nodeName
DOMString nodeValue
short nodeType
Node parentNode
NodeList childNodes
Node firstChild
Node lastChild
Node previousSibling
Node nextSibling
NamedNodeMap attributes
Document ownerDocument

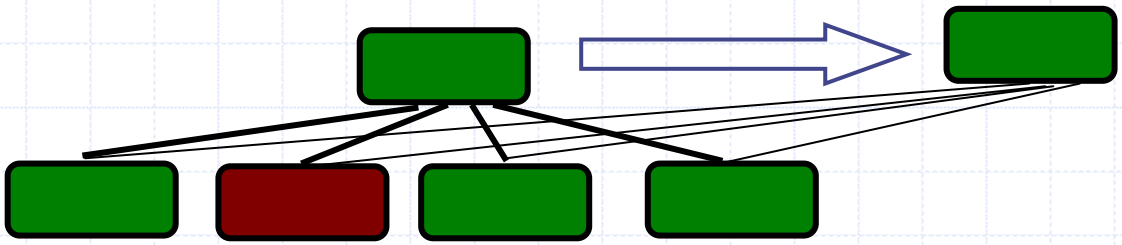


Node Types

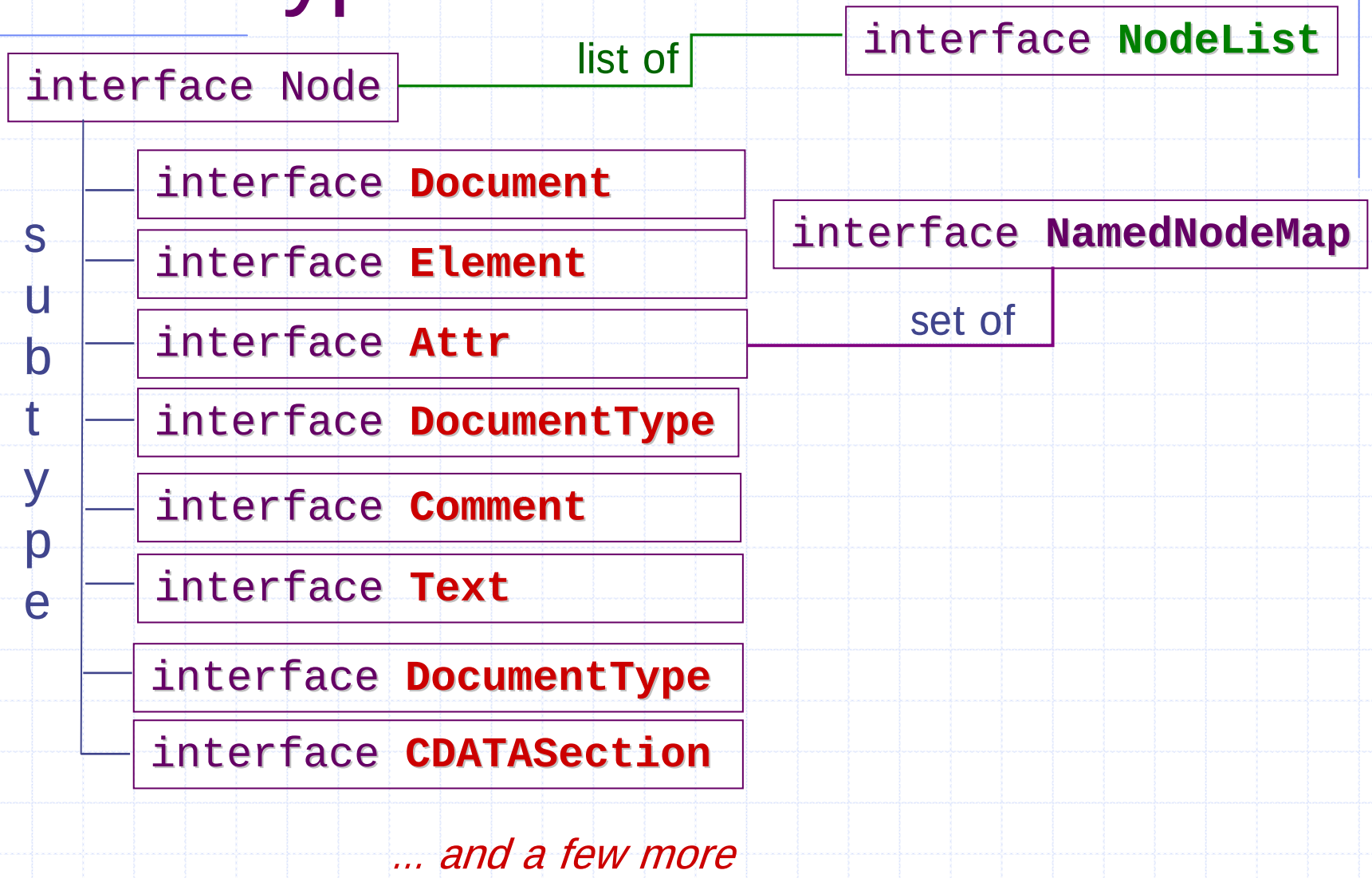

interface Node
subtype
interface Document
interface Element
interface Attr
interface DocumentType
interface Comment
interface Text
interface Entity
interface CDATASection
nodeType
nodeName
nodeValue
9
1
Tag
null
2
AttName
AttValue
10
8
3
#text
content
6
4
... and a few more
(12 nodetypes altogether)

interface Node methods

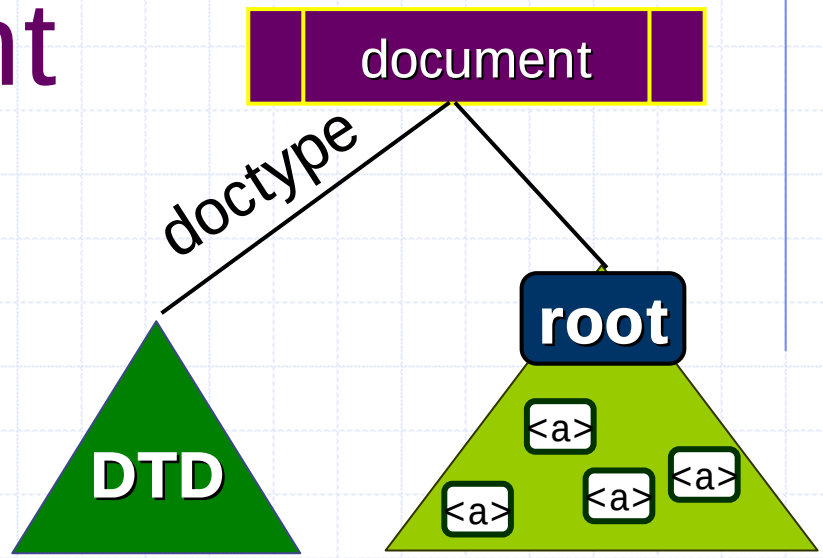
```
Node insertBefore(in Node newChild, in Node refChild)
Node replaceChild(in Node newChild, in Node oldChild)
Node removeChild(in Node oldChild)
Node appendChild(in Node newChild)
boolean hasChildNodes()
Node cloneNode(in boolean deep)
```



DOM Type Structure



interface Document



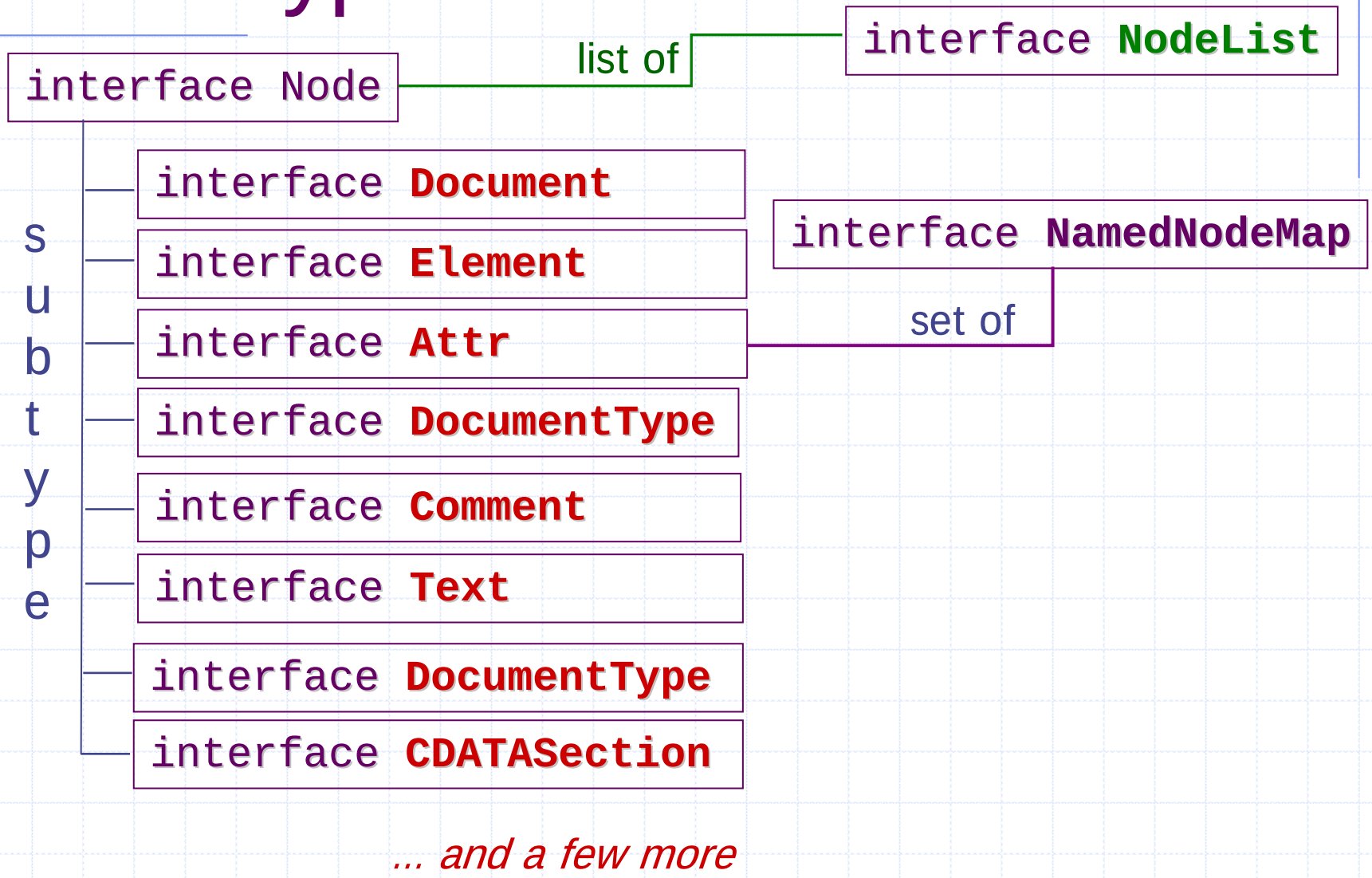
Attributes

DocumentType doctype
Element documentElement

Methods

```
Element          createElement(in DOMString tagName)
DocumentFragment createDocumentFragment()
Text             createTextNode(in DOMString data)
Comment          createComment(in DOMString data)
CDATASection     createCDATASection(in DOMString data)
Attr             createAttribute(in DOMString name)

NodeList getElementsByTagName(in DOMString tagname)
```

DOM Type Structure

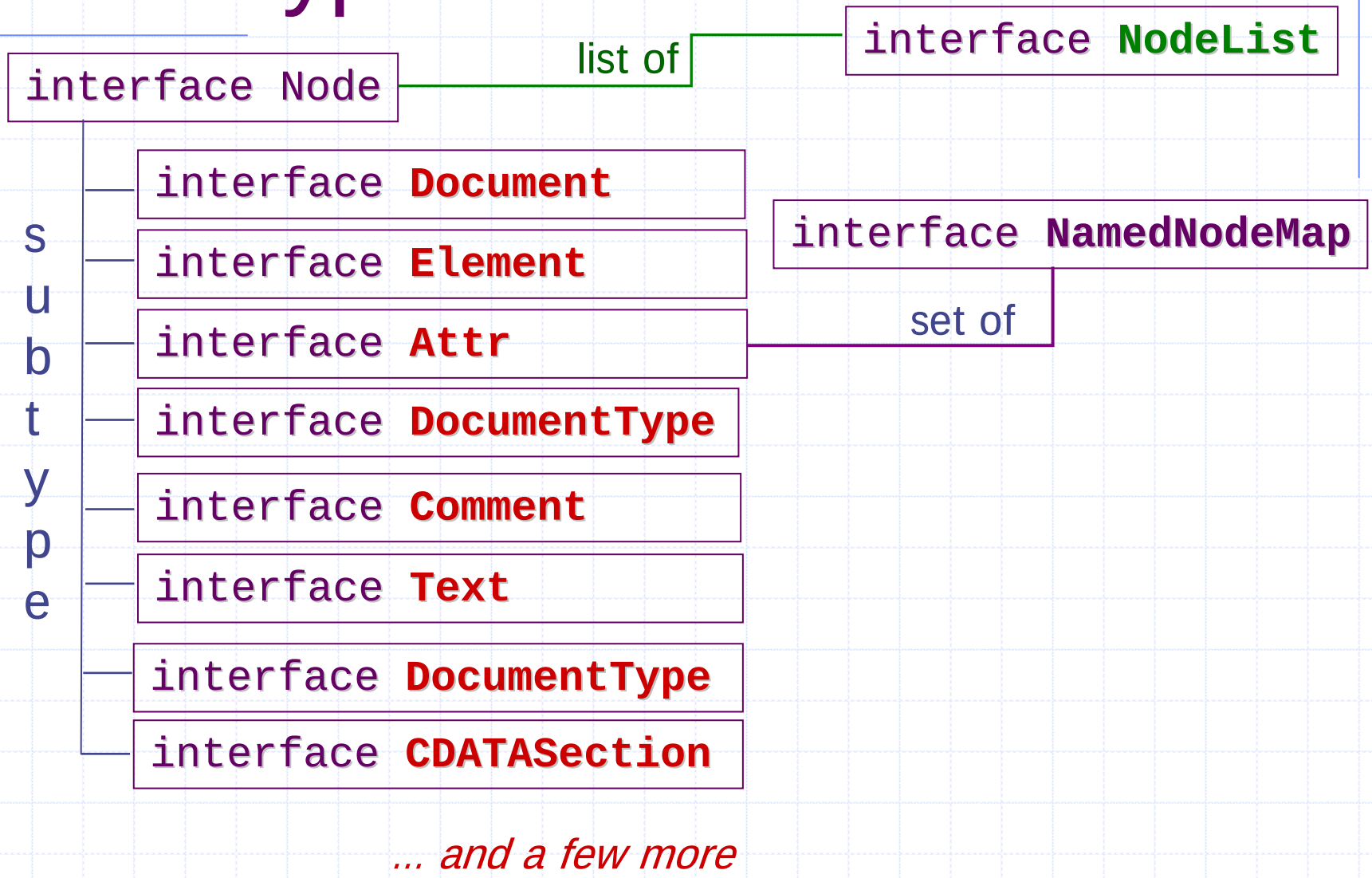


interface Element

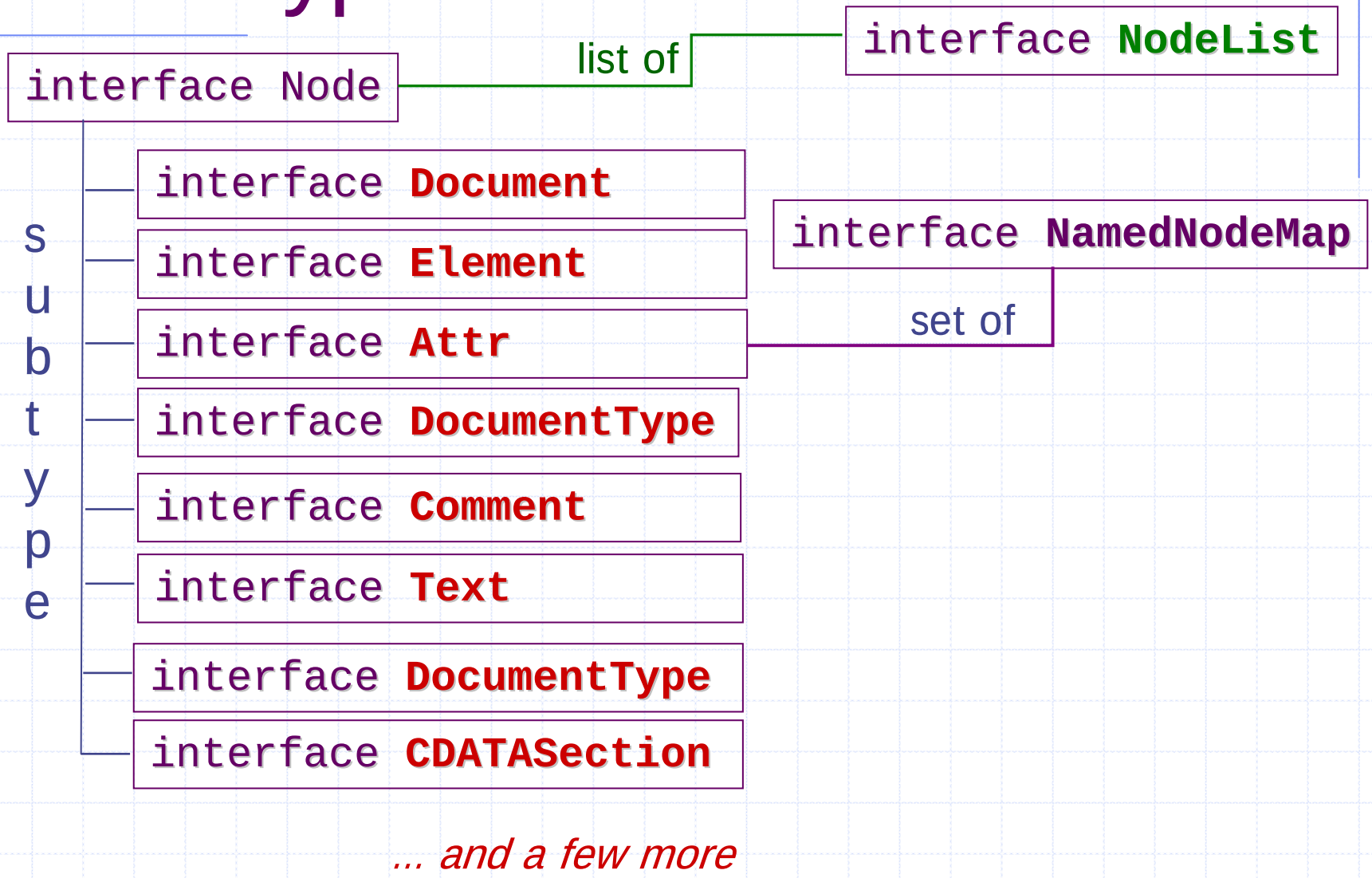
```
interface Element : Node
{
  DOMString tagName;

DOMString getAttribute(in DOMString name);

void setAttribute(in DOMString name, in DOMString value)
void removeAttribute(in DOMString name)

Attr getAttributeNode(in DOMString name);
Attr setAttributeNode(in Attr newAttr)
Attr removeAttributeNode(in Attr oldAttr)


NodeList getElementsByTagName(in DOMString name)

 };
```

Attribute management

DOM Type Structure



DOM Type Structure



interface NodeList

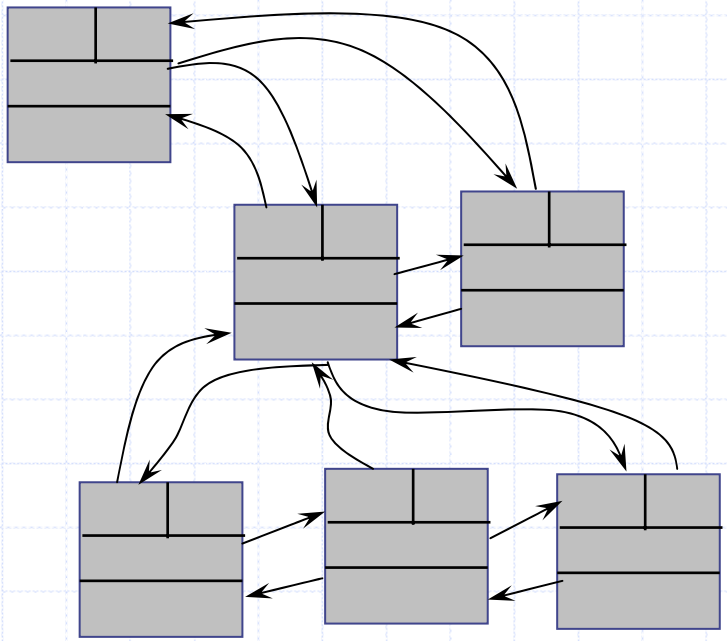
```
interface NodeList
{
   Node item(in unsigned long index);

   unsigned long length;
}
```

interface NamedNodeMap

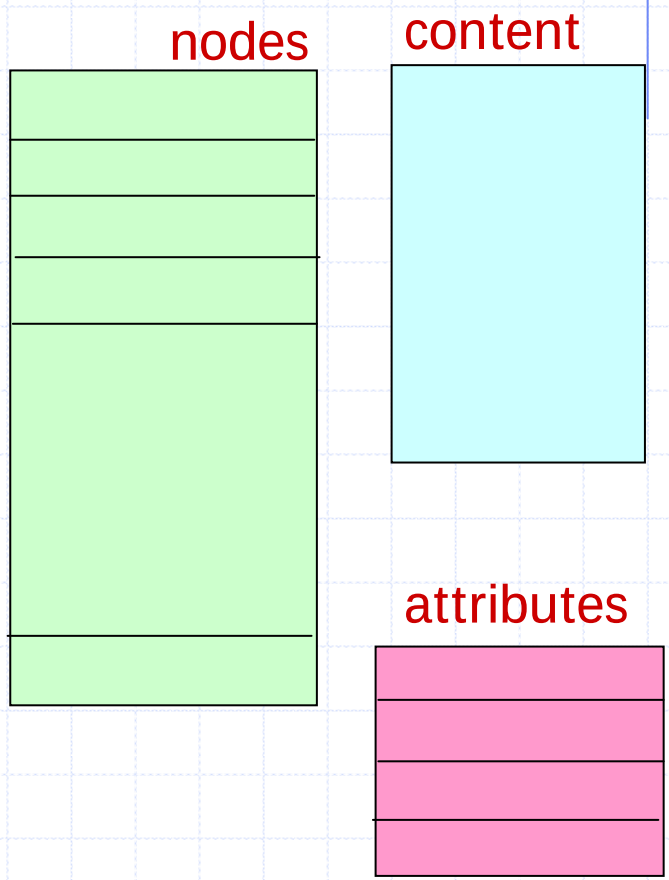
```
interface NamedNodeMap
{
  Node getNamedItem(in DOMString name);
  Node setNamedItem(in Node arg);
  Node removeNamedItem(in DOMString name);

  Node item(in unsigned long index);

  unsigned long length;
};
```

Next...

Implementations of DOM interfaces



1. Using pointers



2. Using record arrays

3. Wrappers over legacy implementations