E-Applications

XML and DOM in Javascript

Michail Lampis
michail.lampis@dauphine.fr

Acknowledgment

- Much of the material on these slides follows the tutorial given in: http://www.w3schools.com/dom/

XML

- XML (eXtensible Markup Language) is a general language format for storing structured text data
- XML is not (exactly) a language itself, but a general format which a data-storing language can follow

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

XML

- The general syntax follows the conventions of HTML
  - Open and close tags: <tag-name> … </tag-name>
  - Attributes can be given inside the opening tag

    <tag-name attr1=”val1” attr2=”val2”>... </tag-name>
  - Proper nesting must be followed:

    <a> …

XML Meaning

- (X)HTML is a form of XML
  - A specific set of tags are allowed (e.g. <p>, <table>, <ol>,...)
  - Their meaning is well-defined
- Which tags are we allowed to use in XML?
- What do they mean?
  - **ANYTHING!**

XML languages

- XML allows us to write **structured** data
- This is useful in many applications:
  - Data becomes easier to read/write/process than plain text
- Depending on the application we have in mind we can **define** an XML format to store our data, or use a pre-existing one.

XML languages

- Example: MathML is a standardized XML format for writing mathematical formulas
- It contains appropriate tags, such as:
  - <mfrac>: denotes a fraction
  - <msqrt>: denotes a square root
  - ...

XML languages

- Example: SVG (scalable vector graphics) is an XML-based format for storing (scalable) image data
- It contains tags such as:
  - <rect>: rectangle
  - <circle>: cycle
  - ...

XML languages

- Example: GraphML is an XML-based format for describing graphs.
- It contains tags such as:
  - <node>
  - <edge>
  - ...

Why do we care?

- XML is one of the most common information interchange formats on the web today.
- Furthermore, in this class we care a lot about AJAX
  - Asynchronous Javascript and XML
- High-level idea: client-side programs which communicate with the server without refreshing the page (asynchronous) and send/receive data in XML.

Reading an XML document

- Javascript gives us facilities to request and read XML

```
xhttp = new XMLHttpRequest();
xhttp.open("GET","books.xml",false);
xhttp.send();
```

- What this means:
  - Send the server a GET request for the file books.xml
  - The request is not asynchronous (3rd parameter is false)...

XMLHttpRequest

- XMLHttpRequest is the general constructor we need to request XML files from the server
- Generally, for security reasons, we are only allowed to request XML files from the same domain
- We prepare a request, and send it
- The xhttp object then contains the response

XMLHttpRequest

xhttp = new XMLHttpRequest();

xhttp.open(“GET”,”books.xml”,false);

xhttp.send();

xhttp.responseXML → contains the XML document in DOM format

xhttp.responseText → contains the XML document in a string

Parsing XML

- The whole point of using XML is that we get the data structured in a nice object (DOM)
- We can still access the string XML
- If necessary, we can transform a string into a DOM object using the XML parser

```
parser=new DOMParser();
xmlDoc=parser.parseFromString(text,"text/xml");
```

Document Object Model

- The DOM is the abstract model that describes the structure of an XML document
- We have already used such an object:
  - The window.document object is a DOM representation of the current HTML document
  - DOM works similarly (but not exactly the same) for HTML and XML

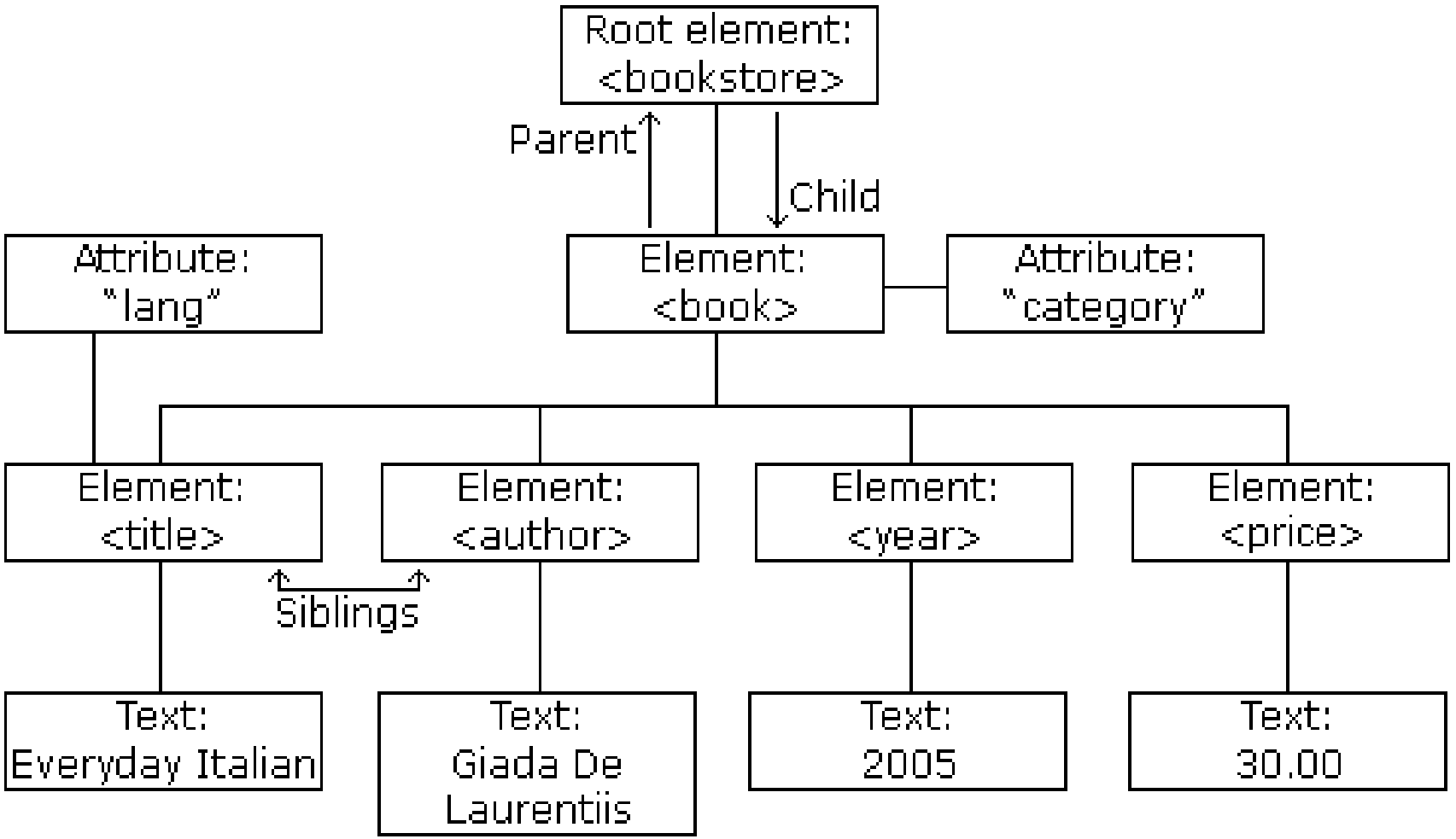
DOM Tree structure

- The DOM structure sees the XML document as a node-tree
- Each XML tag is a node
- ...which may contain other nodes as children

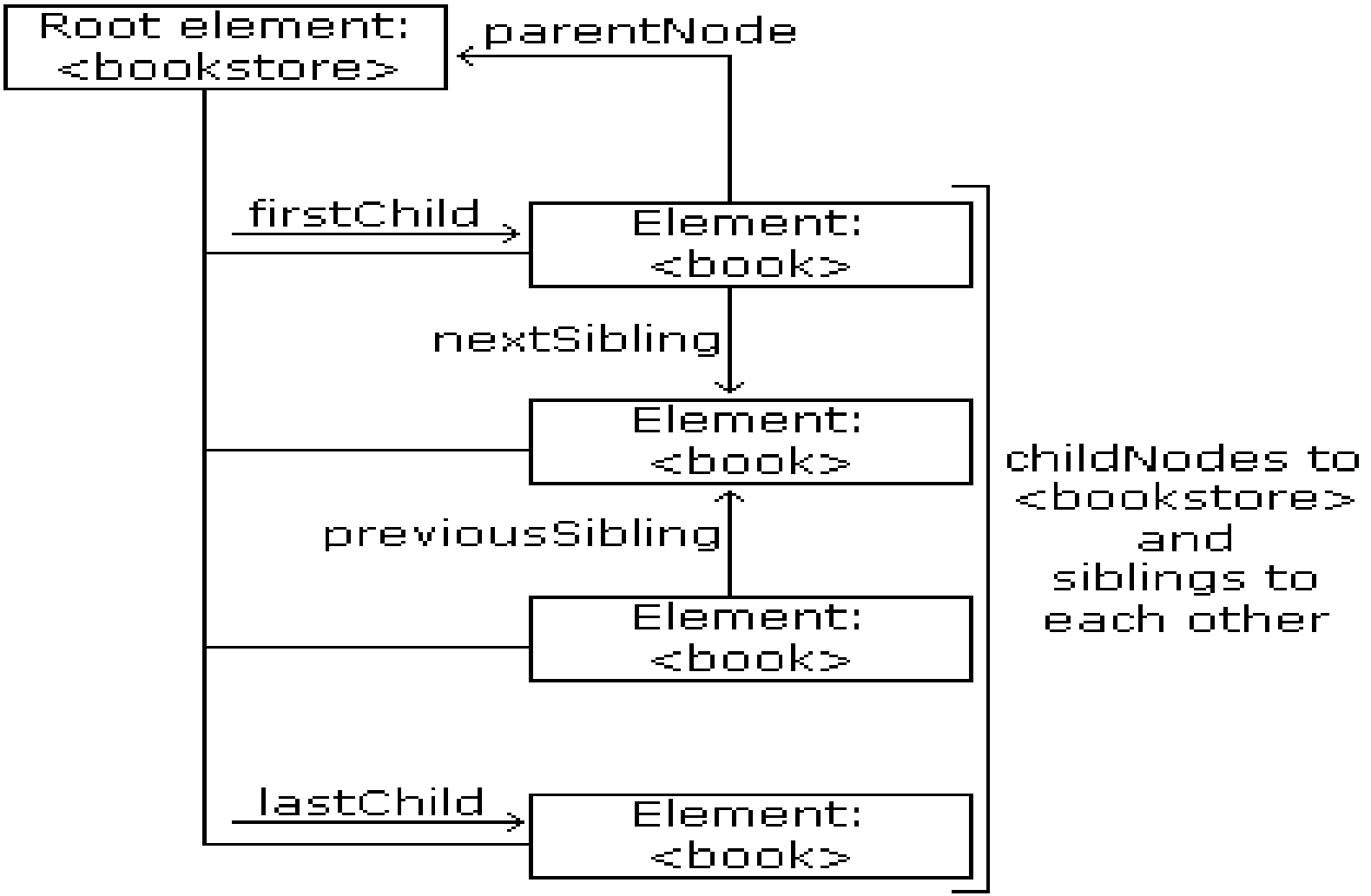
Example (again)

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

DOM Tree structure



DOM Tree Structure



DOM Tree structure

- Terminology:
  - The initial node of the XML document is the **root** node
  - Every node has a **parent** (except the root)
  - A **leaf** is a node without children
  - Two nodes with the same parent are **siblings**

Accessing nodes

- Basic node properties
  - x.nodeName - the name of x
  - x.nodeValue - the value of x
  - x.parentNode - the parent node of x
  - x.childNodes - the child nodes of x **(Array!)**
  - x.attributes - the attributes nodes of x

Accessing nodes

- Basic node methods
  - x.getElementsByTagName(name) - get all elements with a specified tag name (Array!)
  - x.appendChild(node) - insert a child node to x
  - x.removeChild(node) - remove a child node from x
  - x.getElementById(id) - recall how this works in HTML

Some examples

- Let's do some examples with the HTML DOM
  - This allows us to manipulate the web page the user is looking at, and practice our DOM skills!

```
newItem = document.createElement("li");
```

- This creates a new node. Its HTML tag is
  - Attention this does not (yet) appear anywhere in the document!

Some Examples

```
newItem = document.createElement("li");
newItem.innerHTML = “Some text”;
```

- This creates a new tag with type <li> which contains the text “Some text”
- Equivalent to writing Some text in HTML
- This still does not appear anywhere...

Some examples

- Consider the following HTML

  <ul id=”mylist”>

- We can add elements to this list

```
myList = document.getElementById("mylist");
newItem = document.createElement("li");
newItem.innerHTML = “Some text”;
myList.appendChild(newItem);
```

Some examples

- We can also set attributes of elements we add

```
newItem = document.createElement("a");
newItem.innerHTML = “Some link”;
newItem.setAttribute(“href”,”http://www.google.com”);
```

Some examples

- The previous method can be used to add a (unique) id to an item

  newItem.setAttribute(“id”,”someID”);

- We can the use this to find (and remove?) the item

  myItem = document.getElementById(“someID”);

  myItem.parentNode.removeChild(myItem);

Event listeners

- Accessing DOM elements of the HTML page is also useful because it allows us to manipulate the event-handling procedures associated with them.
- addEventListener()/removeEventListener()

Example

document.getElementById(“canvas”).addEventListener(“mousedown”,myfunction);

- (or “mouseup”, “mousemove”,...)

document.getElementById(“canvas”).removeEventListener(“mousedown”,myfunction);