FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

dbai

Semi-structured Data

7 - Document Object Model (DOM)

Methods Overview

Andreas Pieris and Wolfgang Fischl, Summer Term 2016

Node Methods

public String getNodeName()

- The name of the node, depending on its type
  - Document - “#document”
  - Element - Element.tagName
  - Attr - Attr.name
  - Text - “#text”

Node Methods

public String getNodeValue() throws DOMException

- The value of the node, depending on its type
  - Document - null
  - Element - null
  - Attr - Attr.value
  - Text - the content of the text node

Node Methods

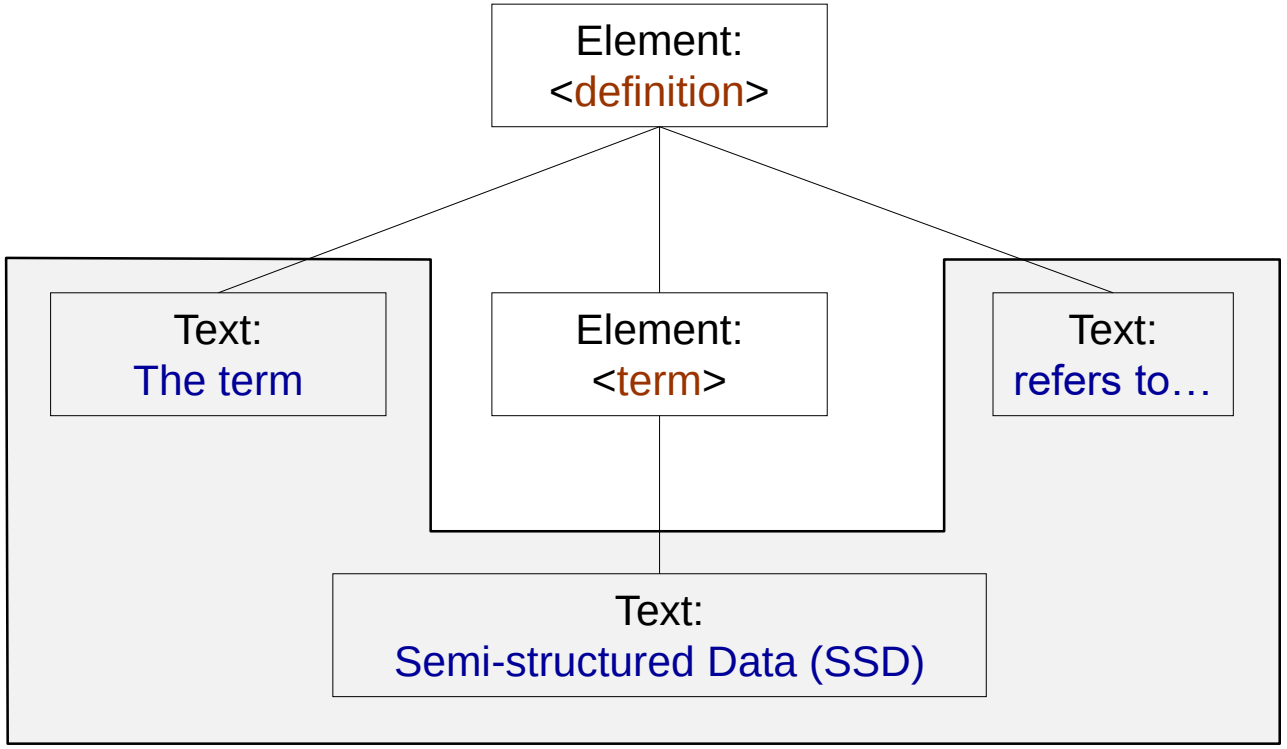
public String getTextContent() throws DOMException

- The text content of the node, depending on its type
  - Document - null
  - Element
  - Attr

  } - concatenation of the text content of every child node
  - Text - the content of the text node

Node Methods

public String getTextContent() throws DOMException



The term Semi-structured Data (SSD) refers to…

Node Methods

public short getNodeType()

- A code representing the type of the underlying object
  - Document - 9 (DOCUMENT_NODE)
  - Element - 1 (ELEMENT_NODE)
  - Attr - 2 (ATTRIBUTE_NODE)
  - Text - 3 (TEXT_NODE)

Node Methods

public String getNamespaceURI()

the namespace URI of the node, or null if it is undefined

public String getPrefix()

the namespace prefix of the node, or null if it is undefined

public String getLocalName()

the local part of the qualified name of the node

Node Methods

- public Node getParentNode()
- public boolean hasChildNodes()
- public NodeList getChildNodes()
- public Node getFirstChild()
- public Node getLastChild()
- public Node getPreviousSibling()
- public Node getNextSibling()
- public boolean hasAttributes()
- public NamedNodeMap getAttributes()

abstraction of an ordered collection of nodes

- int getLength() - number of nodes in the list
- Node item(int i) - i-th node in the list; null if i is not a valid index

- If a node does not exists, then we get null
- A NodeList may be empty (no child nodes)
- getAttributes() from elements; otherwise, null

collection of nodes that can be accessed by name

- int getLenght() - number of nodes in the map
- Node getNameditem(String name) - retrieves a node by name; null if it does not identify any node in the map
- Node item(int i) - i-th node in the map; null if i is not a valid index

Node Methods

```
public Node insertBefore(Node newChild, Node refChild)
                throws DOMException
```

- Inserts the node newChild before the existing node refChild, and returns the inserted node
- If refChild = null, then newChild is inserted at the end of the list of children
- If newChild is already in the tree, it is first removed

Node Methods

```
public Node replaceChild(Node newChild, Node oldChild)
                throws DOMException
```

- Replaces the child node oldChild with newChild in the list of children, and returns the old child
- If newChild is already in the tree, it is first removed

Node Methods

public Node removeChild(Node oldChild) throws DOMException

- Removes the child node oldChild from the list of children, and returns it

public Node appendChild(Node newChild) throws DOMException

- Adds the node newChild to the end of the list of children, and returns it
- If newChild is already in the tree, it is first removed

Node Methods

public Node cloneNode(boolean deep)

- Returns a duplicate of the node - a generic copy constructor for nodes
- If deep = true, recursively clones the subtree under the specified name
- If deep = false, clones only the node itself (and its attributes, in case of an element)

see http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/Node.html

Document Interface

- It provides methods to create new nodes:
  - Attr createAttribute(String name) throws DOMException
    creates an attribute of the given name; its value is the empty string
  - Element createElement(String tagName) throws DOMException
    creates an element of the given name
  - Text createTextNode(String data)
    creates a text node given the specified string

see http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/Document.html

Element Interface

- NodeList getElementsByTagName(String name)

  returns a node list of all descendant elements with the specified tag name, in document order

- boolean hasAttribute(String name)

  returns true if an attribute with the given name is specified on this element; otherwise, it returns false

- String getAttribute(String name)

  returns the name of the given attribute as a string, or the empty string if that attribute does not have a specified value

- void setAttribute(String name, String value) throws DOMException

  adds a new attribute; if an attribute with the given name is already present in the element, its value is simply changed

Element Interface

- void removeAttribute(String name) throws DOMException

  removes the attribute with the given name

- Attr getAttributeNode(String name)

  returns an attribute node with the specified name, or null if such an attribute does not exist

- Attr setAttributeNode(Attr newAttr) throws DOMException

  adds a new attribute node; if the specified attribute exists, then the replaced attribute node is returned

- Attr removeAttributeNode(Attr oldAttr) throws DOMException

  removes the specified attribute node, and returns it

see http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/Element.html

Attribute Interface

- String getName()

  returns the name of the attribute

- String getValue()

  returns the value of the attribute as a string

- Element getOwnerElement()

  the element this attribute is attached to

see http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/Attr.html