# Chapter 11: The XML Document Model (DOM)



## Chapter 11 Objectives

- What is DOM?
- What is the purpose of the XML Document Object Model?
- How the DOM specification was developed at W3C
- About important XML DOM interfaces and objects, such as the Node and Document interfaces
- How to add and delete elements and attributes from a DOM and manipulate a DOM tree in other ways

# What is DOM

- A platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents (W3C)
- A tree-like structure
  - Hierarchy of *nodes*
- Indeed an API
  - Defining objects and properties of all XML elements, and methods (interface) to access them



# What Is XML DOM For?

- How an in-memory representation of an XML document is created?
  - Reading a file from disk. This is a common technique now when dealing with configuration files; user options held in an XML format are easier to deal with than the traditional techniques such as INI files or Registry settings.
  - Converting a string representation embedded in the source code. This is popular for examples but not often used in real-life scenarios because of the difficulty in changing the XML when needed.
  - Accepting a stream piped from another source such as a response from a web service or a database stored procedure.
- Parser reads XML document to memory and converts it into an internal representation
  - Needed to manipulate the XML document

# The Document Object Model at the W3C

**DOM Level 1**

Concentrates on the HTML and XML document models

www.w3.org/TR/DOM-Level-1-Core/

**DOM Level 2**

Defines a set of objects and interfaces for accessing and manipulating document object, including a style sheet object model

www.w3.org/TR/DOM-Level-2-Core/

**DOM Level 3**

Address document loading and saving, and content models (such as DTDs and schemas) with document validation support

www.w3.org/TR/DOM-Level-3-Core/



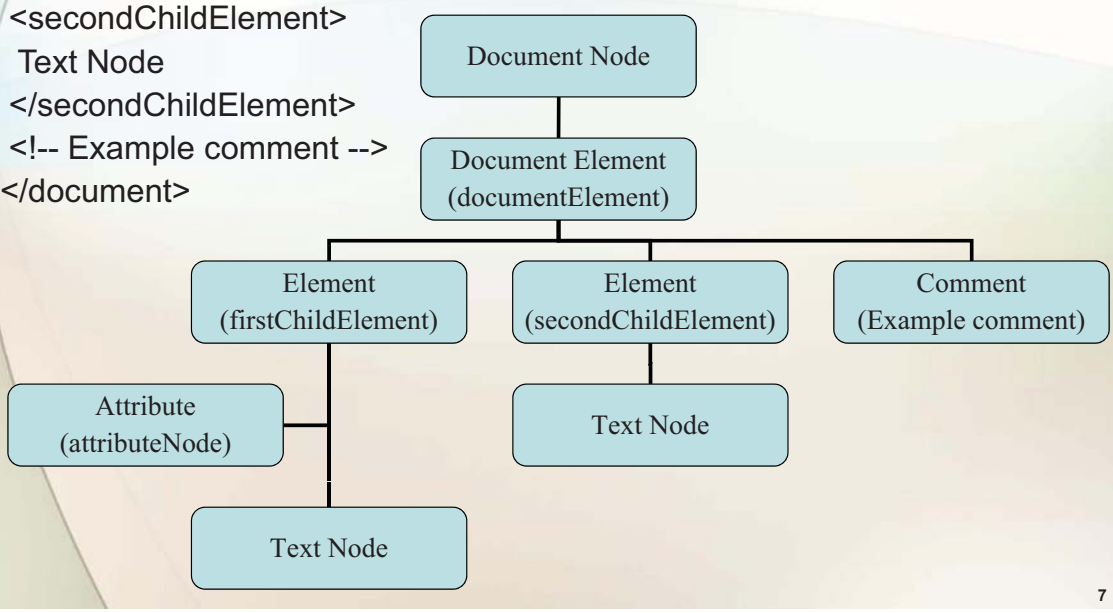
# Overview of the XML DOM Nodes

- **Root of the DOM hierarchy is the Document node**
- **Child nodes are:**
  - **DocumentType nodes**
  - **Element nodes**
  - **ProcessingInstruction nodes**
  - **Comment nodes**

*Only the element node can have child nodes.*

# DOM Example

```
<document>
 <firstChildElement attributeNode="attributeValue">
  Text Node
 </firstChildElement>
 <secondChildElement>
  Text Node
 </secondChildElement>
 <!-- Example comment -->
</document>
```






# DOM Nodes

| Node type | Children |
|---|---|
| Document | Element (maximum of one), ProcessingInstruction, Comment, DocumentType (maximum of one) |
| DocumentFragment | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| DocumentType | no children |
| EntityReference | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| Element | Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference |
| Attr | Text, EntityReference |
| ProcessingInstruction | no children |
| Comment | no children |
| Text | no children |
| CDATASection | no children |
| Entity | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| Notation | no children |

# The Node Object Properties

**attributes**
**childNodes**
**firstChild**
**lastChild**
**localName**
**namespaceURI**
**nextSibling**
**nodeName**
**nodeType**
**nodeValue**
**ownerDocument**
**parentNode**
**prefix**
**previousSibling**



# Methods of the Node Object

appendChild(newChild)
cloneNode(deep)
hasAttributes()
hasChildNodes()
insertBefore(newChild, refChild)
isSupported(feature, version)
normalize()
removeChild(oldChild)
replaceChild(newChild, oldChild)

# The NamedNodeMap Object

getNamedItem(name)
getNamedItemNS(namespaceURI, localName)
item(index)
removeNamedItem(name)
removeNamedItemNS(namespaceURI, localName)
setNamedItem(node)
setNamedItemNS(node)

*and*

NodeList Object
DOMException Object



# Interface Example: The Document Interface

- **Three properties**
  - **documentElement**
  - **doctype**
  - **Implementation**
- **Fourteen methods**

*Run examples from the book and / or at*
*http://www.w3schools.com/dom/default.asp*

| | |
|---|---|
| createAttribute(name) | createEntityReference(name) |
| createAttributeNS(namespaceURI, qualifiedName) | createProcessingInstruction(target, data) |
| createCDATASection(data) | createTextNode(data) |
| createComment(data) | getElementById(elementId) |
| createDocumentFragment( ) | getElementsByTagName(tagname) |
| createElement(tagName) | getElementsByTagNameNS(namespaceURI, localName) |
| createElementNS(namespaceURI, qualifiedName) | importNode(importedNode, deep) |