

What is DOM?

- **Interfaces to represent and manipulate HTML/XML documents**
- **Language-independent**
- **Nodes are objects**
- **Logical tree (or rather, forest) to reflect object relationships**
- **3 levels of specification**

DOM

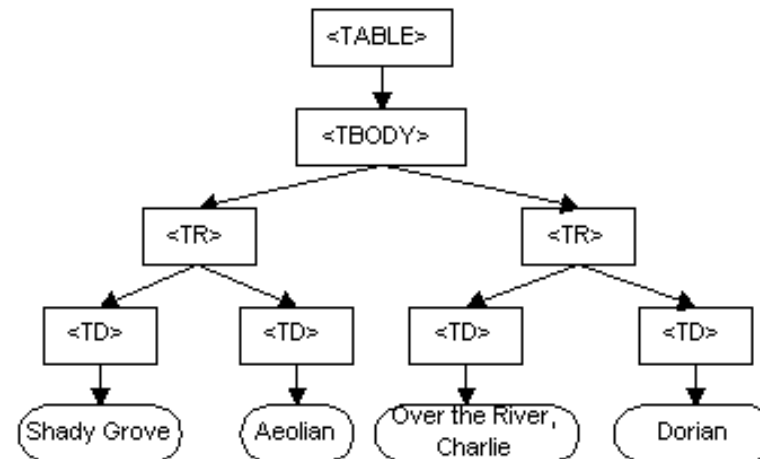
- **DOM is:**
 - **Interfaces and objects used to represent and manipulate documents**
 - **Semantics of these, including behavior and attributes**
 - **Relationships and collaborations between these**
- **DOM is NOT:**
 - **A set of datastructures**
 - **A way of persisting objects to HTML/XML**

DOM level 1

- **2 parts: Core and HTML/XML**
- **Implementations must implement Core and one of either HTML or XML interfaces**
- **DOM does not specify how objects or relationships are implemented, Ie. No requirement to use trees**

DOM example

- `<TABLE>`
- `<TBODY>`
- `<TR>`
- `<TD>Shady Grove</TD>`
- `<TD>Aeolian</TD>`
- `</TR>`
- `<TR>`
- `<TD>Over the River, Charlie</TD>`
- `<TD>Dorian</TD>`
- `</TR>`
- `</TBODY>`
- `</TABLE>`



DOM expansions

- **Level 2**
 - **Extra interfaces and types in Core**
 - **Views: alternate representations of source document**
 - **Events**
 - **Style: style sheets and CSS**
 - **Traversal + Range: tree traversal plus range of content selection**
- **Level 3**
 - **Load + save: importing/exporting XML content**
 - **Validation: queries “if I insert X, will document still be valid?”**
 - **Events expansion**

What is DVM?

- **Interface for accessing XML documents**
- **Value oriented: Nodes are values, not objects**
- **Logical tree representation**
- **Based on “functional” approach**

DVM outline

- **No specification of implementation details**
- **Simple specification, no support for advanced features inherent**
- **Allows sharing of nodes**
- **Values (ie. Nodes) referenced by unique Value Reference**
- **Document referenced by ValRef of root node**

DVM pros and cons

- **Pros:**
 - Values require no concurrency control
 - No need to load entire document into memory to manipulate (which is not true with DOM)
 - Sharing may be exploited when persisting documents
- **Cons:**
 - “functional” approach unusual
 - Few features
 - Frequent updates expensive, without relaxing value model
 - ValRef not human-friendly. Requires name-service.