An XML document is a tree. XPath allows you to concisely describe sets of XML elements and attributes in this tree.

Analogy to file system paths. A file system hierarchy is a tree of directories and files – a filesystem path describes how to reach a specific file, starting either from the root, or the current working directory.

XPath exploits this intuitive notion of a path, and extends it to XML documents.

XPath expressions are evaluated against XML after it has been read by an XML processor, and hence character entity references and CDATA sections have already been parsed, and hence are unavailable to the XPath processor.

In the XML tree available for XPath expressions, there are 7 node types:

- The root node (one per document)
- Element nodes
- Attribute nodes
- Text nodes
- Comment nodes
- Processing instruction nodes
- Namespace nodes

Notes: root node contains only one element node, and possibly many PI nodes and/or comment nodes.
Element nodes correspond to element tags (but not to the enclosed content)
Attribute node correspond to attributes defined on elements
Text nodes are the textual content of a node, and its children

Notes: Although an element node is the parent of its attribute nodes, those attribute nodes are not children of their parent. The children of an element are the text, element, comment, and processing instruction nodes contained in the original element. If you want a document's attributes, you must ask for them specifically.

Processing Instruction Nodes:
A processing instruction node has two parts, a name (returned by the name() function) and a string value. The string value is everything after the name, including whitespace, but not including the ?> that closes the processing instruction.
Namespace Nodes
Namespace nodes are almost never used in XSLT stylesheets; they exist primarily for the XSLT processor’s benefit. Remember that the declaration of a namespace (such as xmlns:auth="http://www.authors.net"), even though it is technically an attribute in the XML source, becomes a namespace node, not an attribute node.

Notion of context

Context is similar to the notion of a current working directory in filesystems. Path expressions are evaluated with respect to the context.

There are two forms of path expression: abbreviated, and unabbreviated

Consider the following XML document:

```xml
<A>
  <B>
    <C/>
  </B>
  <B>
    <D>
      <C id="123"/>
      <C id="456"/>
    </D>
    <E/>
  </B>
</A>
```

The following are abbreviated path expressions (all assume the root is the context)

/ A/B/C – the C element that is a child of a B element that is a child of an A element
C – all three C elements
B – both B elements
D/C – the two C elements that are a child of D
/D/C – the empty set (no D element under root)
D/E – the empty set (no matching path expressions)

Notice:
A path expression is a query whose result is a set of nodes. The result set can be empty, hold one node, or hold multiple nodes.
Unabbreviated path expressions:

/child::A/child::B/child::C – same as A/B/C
child::C – same as C
child::D/child::C – same as D/C

Accesses surrounding a context node are divided into axes:

Parents: “parent::” or “..” selects the parent of the context node
Attributes: attributes of the current node
   “attribute::type” or “@type” are equivalent
Self: “self::” or “.”
Preceding sibling – get previous sibling, if any
Following sibling – get next sibling, if any

XPath wildcards:

- * - all element nodes in the current context – only selects element nodes, not attributes, text nodes, comments, processing instructions
- @* - all attribute nodes in the current context
- node() – all nodes in the current context, regardless of type
- // - zero or more elements can occur between the slashes

Predicates:

Can additionally constrain the number of nodes returned.

number – select only a given element number, as in /D/C[2] – the second C that is a child of D.
attributes – select only elements with a given attribute, as in C[@id="123"] – the C element that has the id “123”.
Example from XSLT, Tidwell, with DTD omitted:

<!-- Default sonnet type is Shakespearean, the other allowable  -->
<!-- type is "Petrarchan."                                      -->
<sonnet type="Shakespearean">
<auth:author xmlns:auth="http://www.authors.com/"
    <last-name>Shakespeare</last-name>
    <first-name>William</first-name>
    <nationality>British</nationality>
    <year-of-birth>1564</year-of-birth>
    <year-of-death>1616</year-of-death>
    </auth:author>
<!-- Is there an official title for this sonnet? They're   -->
<!-- sometimes named after the first line.         -->
<title>Sonnet 130</title>
<lines>
    <line>My mistress' eyes are nothing like the sun,</line>
    <line>Coral is far more red than her lips red.</line>
    <line>If snow be white, why then her breasts are dun,</line>
    <line>If hairs be wires, black wires grow on her head.</line>
    <line>I have seen roses damasked, red and white,</line>
    <line>But no such roses see I in her cheeks.</line>
    <line>And in some perfumes is there more delight</line>
    <line>Than in the breath that from my mistress reeks.</line>
    <line>I love to hear her speak, yet well I know</line>
    <line>That music hath a far more pleasing sound.</line>
    <line>I grant I never saw a goddess go,</line>
    <line>My mistress when she walks, treads on the ground.</line>
    <line>And yet, by Heaven, I think my love as rare</line>
    <line>As any she belied with false compare.</line>
</lines>
</sonnet>
<!-- The title of Sting's 1987 album "Nothing like the sun" is  -->
<!-- from line 1 of this sonnet. -->
Class exercise:

What is selected by the following path expressions (assuming the context node is root):

/sonnet/lines/line
line
/sonnet/@type
/sonnet/auth:author/text()
/sonnet/auth:author/last-name/text()
/processing-instruction()
/auth:*
/@auth:*
/sonnet//line
/sonnet[@type="Shakespearean"]