XML Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">...</xsd:schema>

<xsd:complexType name="name">...
</xsd:complexType>

<xsd:complexType>
  <xsd:simpleContent>
    <xsd:extension base="SomeSimpleType">
    ...
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="myType">
  <xsd:restriction base="baseType">
    <maxExclusive value="100"/>
    <minInclusive value="1"/>
    <pattern value="pattern"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="name" type="type" />

<xsd:element ref="reference" minOccurs="0" maxOccurs="unbounded"/>

<xsd:element name="name">
   <xsd:complexType>...</xsd:complexType>
</xsd:element>

<xsd:sequence>...</xsd:sequence>
<xsd:choice>...</xsd:choice>
<xsd:all>...</xsd:all>

<xsd:attribute name="name" type="type" use="use" default/fixed="value" />
(e.g. <xsd:attribute name="temp" type="xsd:decimal" use="required"/>)

Built-in types: e.g. string, integer, positiveInteger, decimal, float,
boolean, time, date, ...

Enumeration facet:

<xsd:simpleType name="USState">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="AK"/>
    <xsd:enumeration value="AL"/>
    ...
  </xsd:restriction>
</xsd:simpleType>

List and union types:

<xsd:simpleType name="listOfMyItemType">
   <xsd:list itemtype="myItem"/>
</xsd:simpleType>

<xsd:simpleType name="MyUnionType">
   <xsd:union memberType="type1 type2..."/>
</xsd:simpleType>
```

XPath

```
Syntax of an location step:

axis::node-test[expr][expr]...

Axes: child, descendant, parent, ancestor, following-sibling,
preceding-sibling, following, preceding

node tests:

'name-of-element' (e.g. para)
child::*  selects all element children of the context node
attribute::* selects all attributes of the context node
node() true for any node
text() true for any text node
comment()  true for any comment node
processing-instruction()  true for any processing instruction node

core functions:

number last()
number position()
number count(node-set)
node-set id(object)  (e.g. id("foo"))
number sum(node-set)

abbreviated syntax:

child:: -> can be omitted from a location step (default)
attribute:: -> @
/descendant-or-self::node()/ -> //
self::node() -> .
parent::node() -> ..
```

XQuery

```
Below, square brackets ([ and ]) are not part of the syntax.

Element constructors: e.g. <x a="y">...</x>

for $a in distinct-values(document("x.xml")[path expression])
let $b := avg($a/price)
where [conditions]
order by [path expression]
return [element constructors, FLWOR expressions...]

if [condition]
then [expression]
else [expression]

where some $p in [expression] satisfies
where every $p in [expression] satisfies

contains($p/text(), "somestring")

Functions: avg, sum, count, max, min, empty
```