

# XML

BUSI 2710 – Class 18

# XML

- eXtensible Markup Language
- Define your own markup “tag”
  - ▣ E.g. <PurchaseOrder> ... </PurchaseOrder>
  - ▣ E.g. <Customer> ... </Customer>
- A start-tag, corresponding end-tag, and the content between them is called an XML Element
- Tags and nested element structure (“tags within tags”) are defined using XML Schemas
- XML Documents can use those tags to structure business information, e.g. purchase orders, invoices, etc.

# Example

- Elements can contain other elements
  - ▣ Possibly with the same name

```
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <comment>Hurry, my lawn is going
wild!</comment>
  <items>
    <item partNum="872-AA">

<productName>Lawnmower</productName>
      <quantity>1</quantity>
      <USPrice>148.95</USPrice>
      <comment>Confirm this is
electric</comment>
    </item>
  </items>
</purchaseOrder>
```

# Exercise

Describe your Drivers License using an XML document. Make up the tags as you need them.

# XML Schema

- Defines the tags/elements and their structure in an XML document
- Every XML element is of a type:
- Elements of complex Types
  - May have elements as content
  - May have attributes
- Elements of simple Types
  - No other elements as content
  - Cannot have attributes

ShipTo could be an instance  
of a Complex Type

```
<shipTo>  
  <name>Alice Smith</name>  
  <street>123 Maple Street</street>  
  <city>Mill Valley</city>  
  <state>CA</state>  
  <zip>90952</zip>  
</shipTo>
```

Street could be an instance  
of a Simple Type

# XML Pre-defined Simple Types

string	A string of character, i.e. some text
integer	A whole number
float	A number with decimal point
time	Time (hours, minutes, etc.)
date	Date (year, month, date, etc.)
dateTime	Date and Time together
language	Language codes (e.g. “en”, “de”, etc.)
boolean	True or False values

Only a selection!



# XQuery and XPath



# Example Document

```
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <comment>Hurry, my lawn is going wild!</comment>
  <items>
    <item partNum="872-AA">
      <productName>Lawnmower</productName>
      <quantity>1</quantity>
      <USPrice>148.95</USPrice>
      <comment>Confirm this is
electric</comment>
    </item>
  </items>
</purchaseOrder>
```

# XPath

- ❑ In an XML document with nested elements you can select elements as in this example:
  - ▣ `/PurchaseOrder/shipTo/street`
- ❑ This selects the `street` element that is within the `shipTo` element that is within the `PurchaseOrder` element
- ❑ To select the content only (text) of that element, use
  - ▣ `/PurchaseOrder/shipTo/street/text()`

# Advanced XPath

- To select all elements contained in another element, you can use the “\*”, as in this example:
  - ▣ `/PurchaseOrder/items/*`
- This selects all elements that are within the `items` element that is within the `PurchaseOrder` element
- To select only the quantity of the 3<sup>rd</sup> item, use
  - ▣ `/PurchaseOrder/items/item[3]/quantity/text()`

# XQuery

- The simplest XQuery expression is an XPath expression
- Example (note the extra curly brackets for XQuery):
  - `{ /PurchaseOrder/comment/text () }`

# Exercise

Write an XPath expression to select the expiry date of your drivers license

[www.xpathtester.com](http://www.xpathtester.com)

[xpath.online-toolz.com/tools/xpath-editor.php](http://xpath.online-toolz.com/tools/xpath-editor.php)

[chris.photobooks.com/xml](http://chris.photobooks.com/xml)