Announcements

- Deadline extension for HW 5
- HW 6 out
  - Shopping cart functionality
  - Extensions unlikely…
- HW 4 graded

XPath 1.0

- http://www.w3.org/TR/xpath (11/99)
- Not really a programming language
  - Locate nodes in XML tree
  - Perform data operations, e.g., numeric operations, string operations, comparisons
  - XPath 1.0 focuses on locating nodes
  - Newer XPath 2.0 has more XQuery features for data processing, e.g., for loop
- Building block for other W3C standards:
  - XSL Transformations (XSLT)
  - XML Link (XLink)
  - XML Pointer (XPointer)
  - XML Query
XPath – Navigating XML

• When XML is stored in a tree, XPath allows you to navigate to different nodes:

```xml
<Class>
  <Student>Jeff</Student>
  <Student>Pat</Student>
</Class>
```

• XML is similar to a file structure, but you can select more than one node:

```xml
//Class/Student
```

```xml
<Class>
  <Student>Jeff</Student>
  <Student>Pat</Student>
</Class>
```

XPath – Navigating XML

• An XPath expression looks just like a file path
  – Elements are accessed as `/element/`
  – Attributes are accessed as `@attribute`

• Everything that satisfies the path is selected
  – You can add constraints in brackets `[]` to further refine your selection
XPath – Navigating XML

```
<class name='CS 330'>
  <location building='Phillips' name='2017'>
    <professor>Mirek Riedewald</professor>
    <ta>Biswanath Panda</ta>
    <ta>Rishit Shetty</ta>
    <student_list>
      <student id='999-991'>John Smith</student>
      <student id='999-992'>Jane Doe</student>
    </student_list>
  </class>
```

Starting Element Attribute Constraint

```
//class[@name='CS 330']/student_list/student/@id
```

Selection Result: The attribute nodes containing 999-991 and 999-992

XPath - Context

- **Context** – your current focus in an XML document
- Use:
  
  ```
  //<root>/...
  ```

  when you want to start from the beginning of the XML document

XPath - Context

XPath: List/Student
**XPath - Context**

XPath: `Student`

**XPath – Examples**

```xml
<Basket>
  <Cherry flavor='sweet'/>
  <Cherry flavor='bitter'/>
  <Cherry/>
  <Apple color='red'/>
  <Apple color='red'/>
  <Apple color='green'/>
  ...
</Basket>
```

Select all of the red apples:

```
//Basket/Apple[@color='red']
```

**XPath – Examples**

```xml
<Basket>
  <Cherry flavor='sweet'/>
  <Cherry flavor='bitter'/>
  <Cherry/>
  <Apple color='red'/>
  <Apple color='red'/>
  <Apple color='green'/>
  ...
</Basket>
```

Select the cherries that have some flavor:

```
//Basket/Cherry[@flavor]
```
**XPath – Examples**

```
<orchard>
  <tree>
    <apple color='red'/>  
    <apple color='red'/>  
  </tree>
  <basket>
    <apple color='green'/>  
    <orange/>  
  </basket>
</orchard>

Select all the apples in the orchard:
//orchard/descendant()[/apple]
```

---

**Example for XPath Queries**

```
<bib>
  <book><publisher> Addison-Wesley </publisher>
  <author> Serge Abiteboul </author>
  <author> <first-name> Rick </first-name><last-name> Hull </last-name>
  <author> Victor Vianu </author>
  <title> Foundations of Databases </title>
  <year> 1995 </year>
</book>
  <book price="55">  
  <publisher> Freeman </publisher>
  <author> Jeffrey D. Ullman </author>
  <title> Principles of Database and Knowledge Base Systems </title>
  <year> 1998 </year>
</book>
</bib>
```

---

**Data Model for XPath**

[Diagram showing the data model for XPath with nodes such as root, book, publisher, and author, and edges indicating relationships like processing instruction, comment, and the root element.]
XPath: Simple Expressions

/bib/book/year

Result:  

<year> 1995 </year>
<year> 1998 </year>

/bib/paper/year

Result:  empty  (there were no papers)

--

XPath: Restricted Kleene Closure

//author

Result:  

<author> Serge Abiteboul </author>
<author> <first-name> Rick </first-name> <last-name> Hull </last-name> </author>
<author> Victor Vianu </author>
<author> Jeffrey D. Ullman </author>

/bib/first-name

Result:  <first-name> Rick </first-name>

--

XPath: Functions

/bib/book/author/text()

Result:  Serge Abiteboul
Victor Vianu
Jeffrey D. Ullman

Rick Hull doesn't appear because he has firstname, lastname

Functions in XPath:

- text() = matches the text value
- node() = matches any node (= * or @* or text())
XPath: Wildcard

//author/*

Result: <first-name> Rick </first-name>
<last-name> Hull </last-name>

* Matches any element

XPath: Attribute Nodes

/bib/book/@price

Result: “55”

@price means that price has to be an attribute

XPath: Qualifiers

/bib/book/author[first-name]

Result:
<author> <first-name> Rick </first-name>
<last-name> Hull </last-name>
</author>
XPath: More Qualifiers

/bib/book/author[firstname][address[.//zip][city]]/lastname

Result: <lastname> … </lastname>

/bib/book[@price < “60”]

/bib/book[author[@age < “25”]]

/bib/book[author/text()]