

Programming on the Web (CSC309F)  
Tutorial

**Software Tools and CDF facilities**

*Jim Cai*

**[www.cs.toronto.edu/~jcai](http://www.cs.toronto.edu/~jcai)**

**Email: [jcai@cs](mailto:jcai@cs)**

## **Agenda**

### **Software and technology for CSC309**

- Web Browser
- Web Server
- Java Servlets Engine
- Java Database Connectivity (JDBC) Bridge
  
- XML
- JavaScript/ JavaScript Interpreter
- DOM
- Java Applets / Java Applets Container
- SQL / Relational Database

### **How to work at home**

- Cygwin/Putty/VMWare/Eclipse
- Apache
- Tomcat

### **Marking policies**

## Web Browser:

Internet Explorer

Mozilla

Opera

Netscape

...

## XML and DTD

Extensible Markup Language (XML)

- Notice that XHTML is an instance of XML

Example:

- `currency.xml`

```
<?xml version="1.0"?>
<!DOCTYPE currencies SYSTEM "currency.dtd">

<currencies>

  <currency>
    <name>US</name>
    <rate>0.76</rate>
    <date>01/10</date>
  </currency>

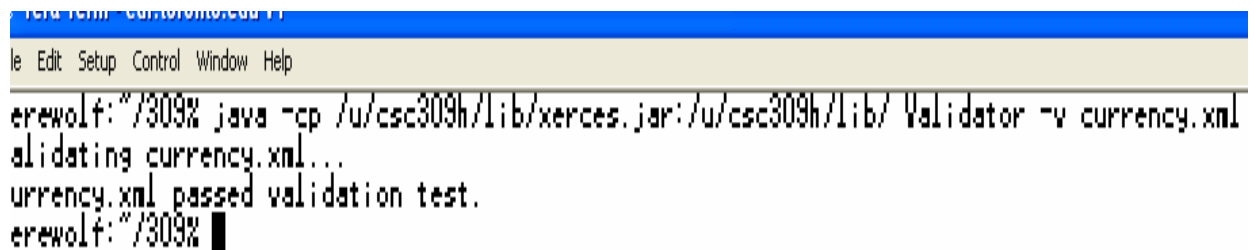
  <currency>
    <name>US</name>
    <rate>0.75</rate>
    <date>02/10</date>
  </currency>
  ...
```
- `currency.dtd`

```
<!ELEMENT currencies (currency+)>
<!ELEMENT currency (name,rate,date)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT rate (#PCDATA)>
```

# How to validate xml?

## CDF:

```
Java -cp /u/csc309h/lib/xerces.jar:/u/csc309h/lib/ Validator -v <doc.xml>
```



A terminal window with a blue title bar and a menu bar containing 'File Edit Setup Control Window Help'. The terminal text shows a user at a shell prompt running a Java command to validate an XML file. The output indicates that the file passed the validation test.

```
erewolf:~/309% java -cp /u/csc309h/lib/xerces.jar:/u/csc309h/lib/ Validator -v currency.xml
Validating currency.xml...
currency.xml passed validation test.
erewolf:~/309% █
```

## At home:

1. Download two files:

- [xerces.jar](#) and [Validator.java](#) from <http://www.cs.toronto.edu/~delara/courses/csc309/resources/resources/XML/validation/>

2. Simply put both files in the same directory as your xml files and type:

```
java -cp xerces.jar: Vaidator -v file.xml
```

# XHTML and CSS

By using a Cascading Style Sheet, one can separate the content and the look of the (HTML) page

Example: My website <http://www.cs.toronto.edu/~jcai>

# JavaScript and DOM

Scripting language vs. Programming Languages (JavaScript vs. Java)

- Variable declaration
- Interpret vs. compile
- Do you remember any other scripting language?

Document Object Model

- Programming Interface for manipulating XML
  - **What did you learn in 207?**
  - **DOM for javascript**

All examples can be obtained from:

<http://www.cdf.toronto.edu/~gljimccm/oldteaching.html>

Try them and get a sense of what DOM can do to your browser?

## Java Applets Container

- Will not be covered in detail

```
import java.applet.Applet;
import java.awt.Graphics;

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```

Run an Applet:

```
<HTML>
<HEAD>
<TITLE> A Simple Program </TITLE>
</HEAD>
<BODY>

<APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```

- How can I debug my Java Applets?  
Mozilla:  
Tools->web development->java console

## Web Server: Apache

- What is a web server?
  - What is a server anyways?
- What can we do with a web server?

### Run Apache Server on CDF:

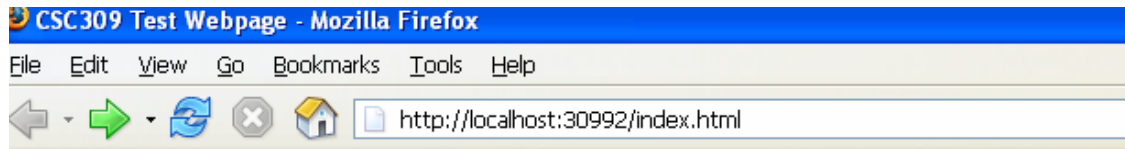
1. Copy all the files from <http://www.cs.toronto.edu/~delara/courses/csc309/resources/resources/apache.tar.gz>
2. untar and run bin/start.sh to start the server

```
werewolf:~/309% tar -xvf apache.tar
apache/
apache/bin/
apache/bin/start.sh
apache/bin/stop.sh
apache/cgi-bin/
apache/cgi-bin/printenv
apache/cgi-bin/test-cgi
apache/conf/
apache/htdocs/
apache/htdocs/helloworld.php
apache/htdocs/index.html
apache/logs/

werewolf:~/309% cd apache
werewolf:~/309/apache% ls
bin  cgi-bin  conf  htdocs  logs

werewolf:~/309/apache% cd bin
werewolf:~/309/apache/bin% start.sh
"Enter your assigned port number: "30992
creating httpd.309.conf using port 30992...
starting the apache web server...
werewolf:~/309/apache/bin% █
```

3. Run your web browser with the URL:  
<http://localhost:<port>/index.html> to get the welcome page.



## CSC309 Apache Server Installed Correctly

replace this page with something useful.

### Some interesting info

- ♦ [CSC Tutorial Manual](#)
- ♦ [CGI program to printenv](#)
- ♦ [CGI program to print stuff](#)
- ♦ [PHP hello world](#)

4. run stop.sh to terminate the running server. (Important: Always do this before logging off)

### Run apache at Home:

1. Download source from [www.apache.org](http://www.apache.org). We prefer you to run apache in Linux so it will better match up with the configuration in CDF environment.
2. Unzip the tar file.
3. Make sure you have root privilege, and do the following:

```
./configure
./make
./make install
fireup the server at default port 80 by running
/usr/local/apache/bin/apachectl start
```

# Java Servlets Engine: Tomcat5.0

- What is Tomcat?
- Where does Tomcat reside?

## Run Tomcat on CDF:

- 1 Copy all the files from <http://www.cs.toronto.edu/~delara/courses/csc309/resources/resources/tomcat/tomcat.tar.gz>
- 2 Set environment variables: (Why? What are we doing here?)  
sh-2.05b\$ JAVA\_HOME=/usr/local/packages/jdk1.3.1  
sh-2.05b\$ export JAVA\_HOME
- 3 run bin/start.sh to start the server
- 4 Run your web browser with the URL: <http://localhost:<port>> to get the welcome page.



- 5 Do not forget to run stop.sh to stop the server afterwards. Do not occupy the port number if you are not using it.

## Run Tomcat At home

1. Download the binary from tomcat website (search google)
2. Unzip the tar file
3. Specify JAVA\_HOME
4. Go to bin directory and run ./startup at default port 8080.

## TomCat Directory Structure:

- **Bin** – Startup/shutdown scripts and some other useful files.
- **Conf** – Configuration files. Server.xml is the file that holds port number information.
- **Logs** – event logs
- **Webapps** – Place that holds servlet applications.
  - Look at Webapps/ROOT
  - Look at Webapps/servlets-samples/WEB-INF/classes/
  - All the class files are held here
  - Now look at Webapps/servlets-samples/WEB-INF/web.xml

```
<servlet>
  <servlet-name>HelloWorldExample</servlet-name>
  <servlet-class>HelloWorldExample</servlet-class>
</servlet>
...
<servlet-mapping>
  <servlet-name>HelloWorldExample</servlet-name>
  <url-pattern>/servlet/HelloWorldExample</url-pattern>
</servlet-mapping>
```
  - To run HelloWorldExample, you need to point your browser at:

<http://localhost:<port>/servlets-samples/servlet/HelloWorldExample/>

More references:

<http://www.cdf.utoronto.ca/~csc309h/summer/pdf/tomcat-tut.pdf>

## **DBMS: PointBase**

- Why PointBase?
  - Portable: 100% Java™
  - Relational Database Management System
- Where is PointBase installed?  
`/h/u2/csc309h/lib/pointbase`
- Do I need to know SQL to work with PointBase?

### **Other Topics:**

- Web services
- Proxies
- Web security
- HTTP

**In one word: A lot of Fun!**

## Submitting your assignments

- 1) Validate all your XML files before submitting them
- 2) Don't submit any compiled code, shared files. (e.g., xerces.jar)
- 3) Compile and test with the same tools that the markers will be using
- 4) For every assignment you **MUST** submit a **README** file that tells your TA how to test your program
- 5) Also document all your bugs/missing features in your assignments in the **README** file
- 6) If the make fails the TA isn't required to try to fix it and may give you a failing mark
- 7) Marking is typically based on Functionality, correctness, usability and documentation.
- 8) Don't use any third-party software other than what is provided to you. (Introduce to Cgywin, Eclipse, Putty, VMWare etc)
- 9) Use “submit” command (man submit if you did not use it before)
- 10) Use CVS to backup and do versioning control. Adapt good software development skills.

## What will your TA do to mark an assignment?

- 1) type **make**
- 2) type **mozilla** <http://localhost:<yourport>/index.html>
- 3) **Kill the running apache/tomcat server**

Check the Guide

<http://www.cs.toronto.edu/~delara/courses/csc309/resources/csc309guide.html>

and

Post your questions to  
**ut.cdf.csc309h**