CS3157: Advanced Programming
Lecture #14
Dec 12
Shlomo Hershkop
shlomo@cs.columbia.edu

Outline

• Advanced topics
  – More html stuff
  – Advanced examples
  – Other issues with multiple languages/environments
  – Thread models
• Review for final

Announcements

• Extra credit project
• Review material
• Review exam
• Extra office hours
  – Please drop email.
• Please fill out evaluations
  – Courseworks.columbia.edu

Basic HTML

• Specific set of tags (depends on version)
• Up to user to set things correctly
• Most browsers will attempt to figure out what you want
  – Example not putting end body end html, will still work
Advanced HTML

- CSS
  - Cascading style sheets
    - Define format of the WebPages
    - Central location of style
    - With a few clicks can completely change thousands of WebPages.
- DOM
  - Document object model
    - Formal data structure to represent web based documents/information
  - Client side scripting

DOM problems

- Different browsers supported things differently
  
  ```javascript
  if (document.getElementById &&
      document.getElementsByTagName) {
    // as the key methods getElementByld and getElementsByTagName
    // are available it is relatively safe to assume W3DOM support.
    obj = document.getElementById("navigation")
    // other code which uses the W3DOM.
    // ....
  }
  ```

Examples


Javascript

- Client side
  - PHP & CGI were both server side
  - Developed under Netscape as LiveScript
    - Currently 1.5
  - Developed under IE as Jscript
- Object oriented approach
- Syntax like c
  - No input/output support native
  - Keywords
  - DOM for interfacing between server and client
- Can evaluate reg expressions (eval)
Javascript

• Heavy use of defined functions
  – Example: MouseOver
• Need to adopt to specific browser if doing anything fancy
• Adobe
  – Support javascript in pdf
• MAC
  – Dashboard widgets

Programming

• You need to learn on your own
• Many good books/websites
• Most of the time .js file if not in html
• Powerful example:
  – Thunderbird/firefox
• Get good debugger

How to do research?

• Practical research
  – Know a programming language
  – Have an inquisitive mind
  – Keep an open mind to new ideas
  – Try to solve an open research problem 😊
• Theory research
  – Learn some math
  – Learn some theory
  – Relearn the math
  – Solve something 😊

Where to start?

1. Need an idea
2. See if anyone’s done it or tried it in your way
   1. Citeseer (citeseer.ist.psu.edu)
   2. Google
   3. Appropriate Faculty/Researcher
   4. Google groups
Sketch out the idea on small scale

- Design a small experiment which can validate your idea
- Data, data, data, and Data
  - Make or break you
  - Will help your research
  - Make sure it isn’t a circular relationship
- Evaluate results
  - Don’t fake them
  - Even bad results are results
  - Can learn of what not to do
- Write up results

Write up

- Word vs Latex
- gnuplot
- cvs
- Element of Style

In the real world

1. Keep it simple
   1. Don’t re-invent the wheel
   2. Design first
   3. Even with fancy blinking lights, a bad idea is still a bad idea (but with bad taste)
2. Incremental testing
   1. Recognize when the bug is your fault
   2. See if others have faced it too
   3. Make sure version 1 works on most popular browsers

Question

- What is this designed with?
- Can you do a better job?

- Theyrule.net
Multi threaded programming

- Thread ideas
- Support of threads
  - Programming language
  - Hardware support
- Issues
  - Synchronization
  - Shutdowns
  - Share resources

Bottom line

- We’ve covered a lot this semester
  - Some of it was fun
  - Some of it was hard work (ok most)
  - Some of it was frustrating.
- BUT
  - You have lots of tools
  - Have an idea of where to start when dealing with programming projects

Review time

- Focus on C
- Shell programming stuff
- CPP
- PHP
- Perl
- webthumb

C

- Basic constructs
- Basic type
- Advanced types
- (review labs and class examples)
- Memory stuff
- Arrays
- Functions
- Pointers
- Debuggers
C
- Working with CGI
- Working on different platforms
- Makefiles

C++
- Basic language
- Difference to c
- Classes
- Permissions
- new/free memory allocations
- Keywords

php
- Basic stuff we did in lab
- Design and language limitations

• Thanks