IT350 Web and Internet Programming

Fall 2007

SlideSet #12: Perl

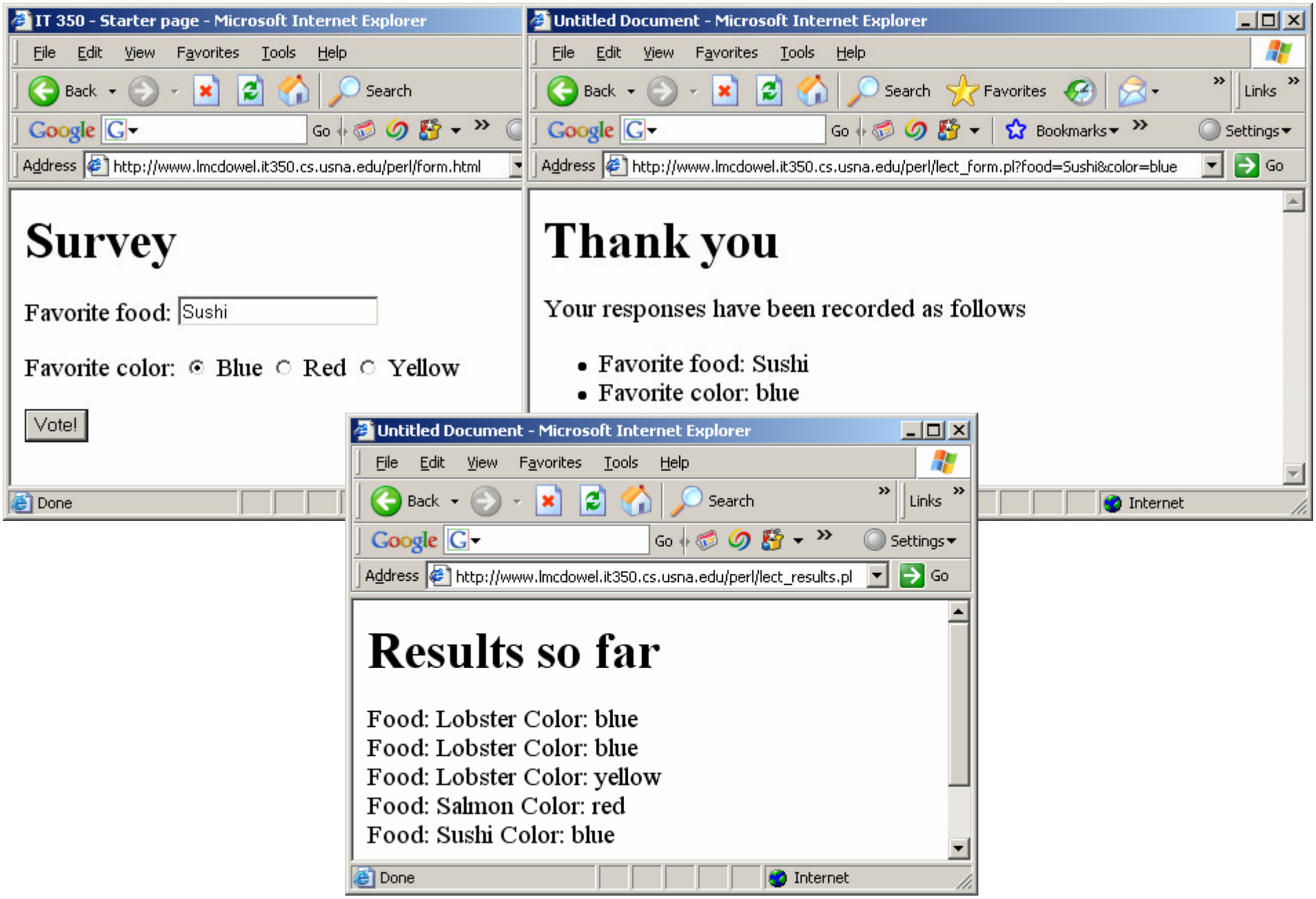
(from Chapter 25 of the text)

FLASHBACK

Things we'll learn and do

- XHTML – basics, tables, forms, frames
- Cascading Style Sheets
- JavaScript
- Dynamic HTML
- CGI

CGI – What does it all look like?



CGI Script Basics

- Common Gateway Interface (CGI)
  - "Common": Not specific to any operating system or language

- Output file generated at runtime:
  1. When a program executed as a CGI script, "standard output" is redirected to client Web server
  2. Web server then redirects output to client's browser

How can CGI get data from user?

Technique #1: Forms

- User enters data via a form, submits
- Form directs results to a CGI program

- Script receives data in one of two ways:
  1. Method = "GET"
  2. Method = "POST"

  Use language-specific method to get these inside CGI program

Technique #2: URL with parameters

```
<a href=http://www.cs.usna.edu/calendar/view.pl?events=seminars>
     Seminars </a>
```

form.html

The Big Example Part 1 (the form)

```
(standard header stuff…)
<body>
  <h1> Survey  </h1>

  <form method="GET" action="lect_form.pl">

      <p> Favorite food: <input type="text" name="food" /> </p>
      <p> Favorite color:
          <input type="radio" name="color" value="blue" /> Blue
          <input type="radio" name="color" value="red " /> Red
          <input type="radio" name="color" value="yellow" /> Yellow
      </p>
      <input type="submit" value="Vote!" />

  </form>
</body>
</html>
```

lect_form.pl

The Big Example Part 2 (CGI to receive)

```
use CGI qw( :standard );
print( header() );
print( start_html() );

# Get inputs from browser user
$favFood    = param("food");
$favColor   = param("color");

# Save result in file.  Use colon as separator
open ( OUTFILE, ">>perl/favorites.txt" );
print OUTFILE "$favFood : $favColor" . "\n";
close ( OUTFILE );

# Thank user and display what was received.
print "<h1> Thank you </h1> \n";
print "<p> Your responses have been recorded as follows</p> \n";

print "<ul> \n";
print li("Favorite food: $favFood");
print li("Favorite color: $favColor");
print "</ul>\n";

print ( end_html() );
```

The Big Example Part 3 (CGI to process)

```
use CGI qw( :standard );
print( header() );
print( start_html() );
print h1("Results so far");
$redCount = 0;

open ( INFILE, "perl/favorites.txt" );
while ($aLine= <INFILE>) {
  chomp ($aVal);

  # Split lines wherever we see a colon
  @myArray = split (/:/, $aLine);

  # Print out the various parts
  print "Food: $myArray[0] Color: $myArray[1] <br/>";

  if ($myArray[1] =~ /red/i) {
    $redCount++;
  }
}
close ( INFILE );

print h2("Found $redCount matches for 'red'.");
print ( end_html() );
```

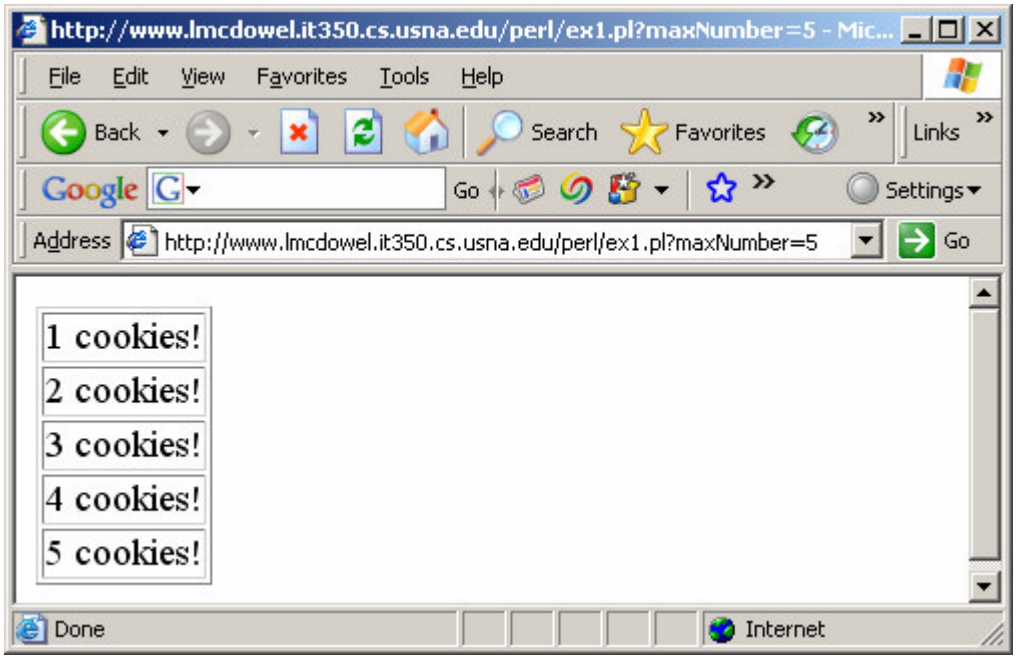
Perl Basics

```
use CGI qw( :standard );
print( header() );

$x = 2 + 3;
$y = $x * 4;

if ($x == 5.0) {
  print ("x is five");
}

for ($i = 0; $i < 3; $i++) {
   $squared = $i * $i;
   print ("<br> \$i = $i, squared is $squared");
}

$pet1 = "dog";
$pet2 = "ll" . "ama";

# Single quotes vs. double quotes
print ("<br/>I have a $pet1 and a $pet2.");
print ('<br/>I have a $pet1 and a $pet2.');

$comp1 = ($pet1 eq "dog");
print ("<br/> comp1: $comp1");
```

Exercise #1

- Write Perl code that will, given the URL provided below, generate HTML that looks like the screenshot

  http://www.lmcdowel.it350.cs.usna.edu/perl/ex1.pl?maxNumber=5



(extra space)

lect_io_array.pl

Exercise #2: What does this code do?

```
use CGI qw( :standard );
print( header() );
print( start_html() );

$index = 0;    $sum    = 0;

open ( MYFILE, "numbers.txt" );
while ($aNum = <MYFILE>) {
  chomp $aNum;
  if ($aNum > 0) {
    $myArray[$index] = $aNum;
    $sum             += $aNum;
    $index++;
  }
}
close ( MYFILE );

$myArray[$index] = $sum;
$index++;

$size = @myArray;
open ( MYFILE, ">numbers.txt");
for ($i = 0; $i < $size; $i++) {
  print br() . $myArray[$i];
  print MYFILE $myArray[$i] . "\n";
}
close (MYFILE);
print ( end_html() );
```

Exercise #3: Write Perl code that accepts two numbers from browser user, prints error if num2 is zero, otherwise outputs num1/num2.

Perl Stuff

"Scalar" variables:

```
$x = 3;
$y = "Hello";
```

"Array" variables:

```
@list = (3, 7, "dog", "cat");
@list2 = @list1;      # copies whole array!
```

A single element of an array is a "scalar:

```
print "Second item is: $list[1]";    # Don't use @
```

Get array length by treating whole array as scalar:

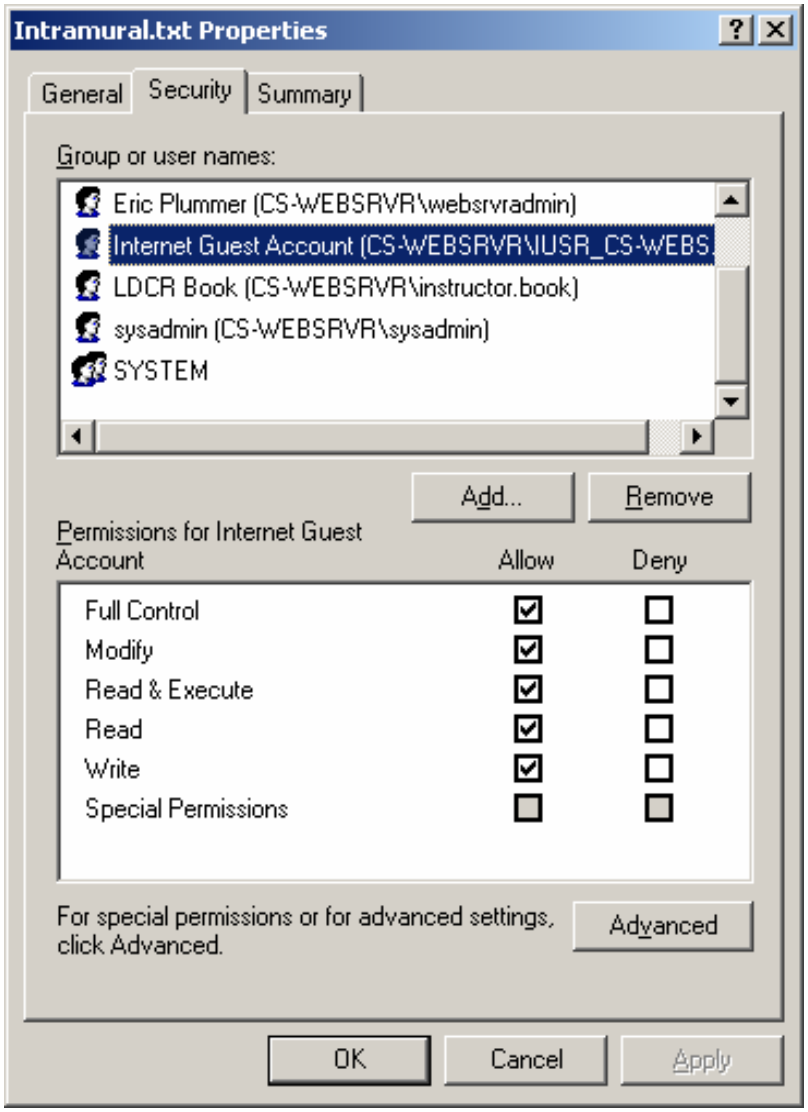
```
$lengthOfList2 = @list2;
```

File operations

```
open ( MYFILE, "input.txt" );
open ( MYFILE, ">output.txt" );
open ( MYFILE, ">>LOG.txt" );
```

File Access

- Ownership: Input/Output files usually **NOT** owned by “Web Server”.
  - Operating system may enforce read, write, and/or modify restrictions on I/O files
  - For file output/append, may need to create file prior to first use
  - File permissions need set for access by the “web server” account (Right-click on file, pick Properties, then set permissions like example on right)



File Access

- File Path/Naming
  - CGI Script may **NOT** run within script's location
    - May need to provide full or relative path information
      - All CGI processes on ***cs-websrvr*** are run from the top directory of the account and require path information to Input/Output file's location
      - E.g. "Lab10/LOG.txt" or "perl/results.txt"