Perl/CGI
CGI Programming in Perl

- CGI: the standard programming interface between Web servers and external programs
  - Common: interacts with many different OSs
  - Gateway: provides users with a way to gain access to different programs
  - Interface: uses a well-defined method to interact with a Web server
- CGI standard lets Web browsers pass information to programs written in any language
Writing and Running CGI Scripts

- CGI scripts can be compiled programs or batch files or any executable entity
- Typically CGI scripts are written in:
  - Perl scripts
  - C/C++ programs
  - Unix Scripts

- The 2 most common ways of running a CGI script are:
  - From an HTML form – the “action” attribute of the form specifies the CGI script to be run
  - Direct URL reference – A CGI script can be run directly by giving the URL explicitly in HTML
Why Use Perl for CGI?

Perl is the standard for CGI programming:
- **Socket support** – create programs that interface seamlessly with Internet protocols
- **Pattern matching** – ideal for handling form data and searching text
- **Flexible text handling** – no details to worry
- **Advantage of an interpreted language** – simplicity in development, debugging, and revision
How Does CGI Work?

• CGI programs are always placed on a disk that the Web server has access to.

• Web servers are generally configured so that all CGI applications are placed into a \texttt{cgi-bin} directory.
Invoke CGI Programs

- There are many ways to invoke CGI programs besides using a web browser to visit the URL.
- We can start CGI programs from:
  - a hypertext link
    - `<a href="cgi-user/ftang/www/cgi-bin/hello.cgi">Click here to run the hello CGI program</a>`
  - a button on a HTML form
  - a server-side include
- We can use a question mark to pass information to a CGI program.
  - For example, passing keywords that will be used in a search
    - `<a href="cgi-user/ftang/www/cgi-bin/search.pl?Wine+1993">Search for 1993 Wine</a>`
    - The information follows the question mark will be available to your CGI program through the `QUERY_STRING` environment variables.
- Generally speaking, visitors to your web site should never have to type in the URL for a CGI program.
- A hypertext link should always be provided to start the program.
The Hello World Example

- When the web server executes your CGI program, it automatically opens the STDIN, STDOUT, STDERR file handles for you.
  - **STDIN**: the standard input of your CGI program might contain information that was generated by an HTML form. Otherwise, you shouldn’t use STDIN.
  - **STDOUT**: the standard output of your CGI program is linked to the STDIN of the web browser.
  - **STDERR**: The standard output of your CGI program is linked to the web server’s log file.
CGI Script Output

- A CGI script must send information back in the following format:
  - The output header
  - A blank line
  - The output data

- For examples:
  - `print("Content Type: text/html\n\n");`

- Note: Between the Header and Data there MUST be a blank line
Output Header: Content-Type

- The output header for a CGI script must specify an output type
- 3 forms of Header Type:
  - Content-Type:
    - text/html, text/plain, image/gif, image/jpeg, application/, postscript, video/mpeg
  - Location
  - Status
- Content-Type is the most popular type

- See our hello world example:
  - [http://www.csupomona.edu/cgi-user/ftang/www/cgi-bin/hello.pl](http://www.csupomona.edu/cgi-user/ftang/www/cgi-bin/hello.pl)
Output Header: Location

- The **Location** header is used to redirect the client Web browser to another page

- For example:
  - `print "Location: http://www.google.com/\n\n";`
CGI Script Input

• A CGI script will often require some form of input in order to operate

• We’ll study:
  – What form of input a CGI can receive
  – How a CGI receives input
  – How to process the input in a CGI Perl script
  – How a useful Perl library makes this easy
Accepting Input from the Browser

- A CGI script can receive data in many ways:
  - Environment variables
  - Standard input
    - Data can be passed as standard input through the POST method of an HTML form
  - Arguments of CGI script
    - If you call a CGI script directly or use the GET method of a form, arguments are following the “?” after the script URL and multiple arguments are separated by &
Receiving/Processing Information in CGI

• Two basic ways:
  – Do it yourself: write Perl code to process the input
  – Use pre-written Perl libraries
    • http://stein.cshl.org/WWW/CGI/ a Perl5 CGI Library
Forms and CGI: What Can They Do?

- A CGI program can be used to:
  - accept the data which the user inputs
  - do something with it
    - email the data to someone
    - add an entry to a database
    - write out a text file
    - create a customized display
    - anything that you can program
  - send a reply back to user
Form Processing

• The two most important options with the <form> tag:
  – **method**: specifies the manner in which form information is passed to the CGI scripts:
    • **POST**: contacts the CGI program and once connection established sends data
    • **GET**: Contacts the CGI program and sends data in a single transaction
  – **action**: specifies the URL of the CGI script that will be invoked when the submit button is clicked
    • `<form method = “post” action="/cgi-bin/hello.pl">`
Handling Form Information

• The GET method
  – `<form method="get" action="/cgi-bin/guestbook.pl">`
  – The GET method appends all of the form data to the end of the URL used to invoke the CGI script.
  – A question mark is used to separate the original URL and the form information.
    • `http://somserver/cgi-bin/foo.pl?fname=Craig&lname=Kelley`
  – The GET method can’t be used for larger forms
Handling Form Information

- The POST method
  - `<form method="post" action="cgi-bin/guestbook.pl">
  - The POST method sends all the form information to the CGI program using the STDIN file handle.

- Which to use?
  - It is good practice to use the GET method whenever you are able to because the POST method is more difficult for a user to manage, and it doesn’t function well with a browser’s back or history button.

  - It is good to use the POST method when something secure or private is being sent such as password or a credit card information.
CGI.pm: a Perl 5 CGI Library

• This Perl 5 library uses objects to do many things. It
  – Uses objects to create Web fill-out forms on the fly and to parse their contents
  – Provides a simple interface for parsing and interpreting query strings passed to CGI scripts
  – Offers a rich set of functions for creating fill-out forms

• Everything is done through a “CGI” object
• The most basic use of CGI.pm is to get the query parameters submitted to your script. To do so, put the following at the top of your Perl/CGI programs:
  – Use CGI;
  – $query = new CGI;
  – Or Use CGI qw(:standard);

http://stein.cshl.org/WWW/CGI/
Function Oriented vs. Object Oriented

```perl
#!/usr/local/bin/perl
# imports standard set of functions
use CGI qw/:standard/;

print header(),
    start_html(-title=>'Wow!'),
    h1('Wow!'),
    'Look Ma, no hands!',
    end_html();
```

http://stein.cshl.org/WWW/CGI/#functionvsoo
What Can You Do with the Query Object?

• Fetch the names of all the parameters passed to your script
  – @$names = $query->param;

• Fetch the value(s) of a named parameter
  – @$values = $query->param('foo'); # or
  – $value = $query->param('foo');

• And many other methods ...

• Example:
  – http://www.csupomona.edu/~ftang/www/courses/CS299-S09/examples/simple.html
Printing HTML

• Instead of printing the quote marks and write in new line characters

• An easier way is to use a special print command in Perl:

  print <<AnyWord
  ...
  AnyWord

  ENDHTML
Save the Current State of a Form

• Save the state to a file
  – $query->save(FILEHANDLE);
  – The contents are written out as TAG=VALUE pairs

• Save the state in a self-referencing URL
  – $my_url = $query->self_url;
  – Useful when you want to jump around within a script-generated document using internal anchors, but don’t want to disrupt the current contents.
Three Different Methods

- Print out the information after form submission
  - www.csupomona.edu/~ftang/www/cgi-bin/procform1.cgi
- Submit the information to an email address
  - www.csupomona.edu/~ftang/www/cgi-bin/procform2.cgi
- Save the information to a file (will not use this one)
  - www.csupomona.edu/~ftang/www/cgi-bin/procform3.cgi

- To run the above programs:
  - http://www.csupomona.edu/~ftang/www/courses/CS299-S09/examples/runcgi.html
Creating the HTTP Header

• Creating the standard header
  – print $query->header()
    • default to ‘text/html’

• Creating the header for a redirection request
  – print $query->redirect('http://www.google.com');
HTML Shortcuts

- Create an HTML header
  - `print $query->start_html(many options);`
- End an HTML document
  - `print $query->end_html;`
- Shortcut methods for many HTML tags
- Importing CGI methods
  - Single query object
  - `import CGI module methods`
  - `use CGI qw(:standard);`
  - `$dinner = param('entrée');`
Other Supports

• Support for CSS
  – http://www.csupomona.edu/cgi-user/ftang/www/cgi-bin/style.pl
  – http://www.csupomona.edu/cgi-user/ftang/www/cgi-bin/bio.pl

• Support for JavaScript
  – Similar as CSS
  – check http://stein.cshl.org/WWW/CGI/ for details
    – http://www.csupomona.edu/cgi-user/ftang/www/cgi-bin/contact.pl
Many Perl/CGI Examples

- http://stein.cshl.org/WWW/CGI/examples/