Fill-out Forms & CGI

Fill-out forms
Fields and buttons
CGI programming

http://www.comp.leeds.ac.uk/Perl/Cgi/start.html
http://www.jmarshall.com/easy/cgi/

Interactive Web Pages

- Web pages are not just for reading
- Search engine page
  http://www.google.com/
- On-line shopping page
  http://www.buy.com/default_2.asp
- Map request page
  http://maps.yahoo.com/maps
- Driving direction page
  http://www.mapquest.com/directions/main.adp
- Many other types of pages

Fill-out Forms

- Fill-out forms are used in Web pages
  - For users to input data
  - For underlying programs to read the user-provided data and then carry out operations

- Syntax skeleton of a FORM
  `<FORM ACTION="URL" METHOD="method">
  Normal page content INPUT and other tags
  </FORM>`
  - The URL is of the associated script
  - The method can be either GET or POST
  - When GET is used, the query string is attached at the end of the URL
  - When POST is used, the query string is sent directly to the script through a communications channel in a server-based format

- There may be multiple FORMS in a page
- No nesting of FORMS is allowed
- No overlapping of FORMS is allowed

Fill-out FORMS in an HTML page

- An HTML page can have one or more fill-out FORMS
- Each FORM is associated with an executable script (program), which typically reads some input, does something, generates some output
- Each FORM has one or more INPUT and some other tags
  - An INPUT tag is for a field or a button, for accepting user input
Fields and buttons in a FORM

• A one-line text entry field
  
  `<INPUT TYPE="text" NAME="user" VALUE="Donald Smith" SIZE=30>`

  – The TYPE is text
  – The NAME `user` is the name of the parameter, associated with this field
  – The optional VALUE `Donald Smith` is the default input value.
  – The SIZE 30 is the field width in characters, scrolling when needed.

  MAXLENGTH can be specified to limit the number of characters the field will accept.
  Different fields must have different NAMEs.

• A one-line password entry field

  `<INPUT TYPE="password" NAME="passwd" VALUE="xyzzy" SIZE=10>`

  – TYPE is password
  – All characters in the password field are shown as asterisks or dots.
  – Otherwise, this is the same as the previous one-line text entry field

  The value of the password field is passed as ordinary text, not encrypted or hidden.
  Using method GET will have the password value displayed in the URL field. Should use POST.

• TEXTAREA tag, not an INPUT tag

  For defining a large input text area, not just a field of a single line, use TEXTAREA tag.

  `<TEXTAREA NAME="LONGTEXT" ROWS=5 COLS=60>`

  It will appear just like a one-line text entry field, however much larger. Scroll bar(s) will show up when appropriate.

• A file upload field

  `<INPUT TYPE="file" NAME="upload" SIZE=40>`

  – TYPE is file
  – An input field and a Browse button will appear
  – When the Browse button is clicked, a File Upload window will show up for selecting a file
• A group of checkboxes
  They are used for a user to select multiple items.

  ```html
  <INPUT TYPE="checkbox" NAME="items" VALUE="engine" > Engine
  <INPUT TYPE="checkbox" NAME="items" VALUE="tire" CHECKED> Tire
  <INPUT TYPE="checkbox" NAME="items" VALUE="seat" > Seat
  ```

  - The TYPE is checkbox
  - Three checkboxes are declared above

  - They are grouped together with common parameter NAME items
  - CHECKED tire is a default selection, will be initially shown to user as checked

    ![Checkbox selection](image)

  - The text strings Engine, Tire and Seat are for user to read, the VALUES engine, tire and seat are to be passed to the program if selected
  - Assume that the user selected Engine and Tire, then the corresponding part of the query string will be
    
    `items=engine&items=tire`

• A group of radio buttons
  Similar to checkboxes, but the user can select only one out of a group of radio buttons.

  ```html
  <INPUT TYPE="radio" NAME="size" VALUE="small" CHECKED> Small
  <INPUT TYPE="radio" NAME="size" VALUE="medium"> Medium
  <INPUT TYPE="radio" NAME="size" VALUE="large">Large
  ```

  - TYPE is radio

  - The first radio button, CHECKED, is the default
  - The radio buttons in the same group must use the common parameter NAME, in here: size
  - Only the last selected one is effective
  - Assume that the user chose Large, then the corresponding part of the query string is

    `size=large`

    ![Radio button selection](image)

• A scrolling list, SELECT tag
  A list of options for user to pick. May allow only one or multiple ones to be picked.

  ```html
  <SELECT NAME="favorites" SIZE=4 MULTIPLE>
  <OPTION SELECTED>swimming
  <OPTION SELECTED>running
  <OPTION>jogging
  <OPTION>skiing
  <OPTION>basket ball
  <OPTION>foot ball
  </SELECT>
  ```

  - Window size 4 items
  - Scrolling if needed
  - Multiple choices allowed
  - Two default selections

• A reset button
  It allows a user to reset all the field values and selections in a FORM back to their original default values.

  ```html
  <INPUT TYPE="reset" VALUE="Clear">
  ```

  - TYPE is reset
  - The VALUE, Clear, is shown as the label of the button
  - When VALUE is not given, then the default label is Reset

    ![Reset button](image)
• A submit button
  When this button is clicked, the browser packages the contents, which may come from many INPUT tags of the form, and submits the resulting query string to the script designated in the FORM tag.
  
  ```html
  <INPUT TYPE="submit" VALUE="Send Order">
  ```
  
  – The TYPE is submit
  – The button is labeled with VALUE Send Order. Default for VALUE is Submit.

• Multiple submit buttons in a FORM
  Multiple submit buttons may be used for a single FORM.
  
  ```html
  <INPUT TYPE="submit" NAME="action" VALUE="Send Order">
  <INPUT TYPE="submit" NAME="action" VALUE="Order Later">
  ```
  
  – When multiple submit buttons are used in a single FORM, they should be labeled differently.
  – In here, there are two submit buttons, labeled Send Order and Order Later, respectively.
  – Only one submit button can be clicked/effective.
  – If the user clicked on the button labeled Send Order, then the corresponding part of the query string will be action=Send+Order

• Using an image as a submit button
  An image can also be used as a submit button, typically the image is some icon.
  
  ```html
  <input type="image" name="lion" src="../odulion.gif">
  ```
  
  – The TYPE is image
  – Must specify the source URL, src
  – When the image is clicked, the corresponding part of the query string will be: lion.x=xvalue&lion.y=yvalue

Common Gateway Interface (CGI)

• Common Gateway Interface (CGI)
  – A technology that enables a client web browser to request data from a program (also referred to as a script) executed on the web server
  – A protocol for passing data between the client and the program
  – Platform independent
    The Web server does not need to know anything about the language in question.

An example page, including all the items discussed previously:

http://www.cs.odu.edu/~shen/cs312/htmldir/InputTags2.html
• How CGI works
  – The client sends a cgi request
    • If the method is GET then the query string of the arguments is tacked onto the end of the URL and sends the URL to the www server.
    • If the method is POST then the client sends the query string directly to the server, separately from the URL.

  – When a web server receives a cgi request
    • It creates a set of environment variables containing information about
      – the server itself
      – the remote browser
      – the current request including QUERY_STRING
    • It calls the corresponding script with any arguments in the environment variable QUERY_STRING.

• The script picks up any information it wants from the environment variables and executes its own instructions particularly the arguments from QUERY_STRING

• The output by the script, an HTML page, is sent back to the client by the server

• Communicating with scripts through hyperlinks
  – Scripts may or may not require arguments from users.
  – The arguments are called a query string and may be appended at the end of a URL with the question mark “?” leading it.
  – Examples
    • No argument
      http://www.cs.odu.edu/~shen/cgi-bin/printEnv.cgi
    • Single argument value
      http://www.cs.odu.edu/cgi-bin/calendar?2005
    • One argument with parameter and value
      http://www.google.com/search?q=titanic
    • Argument value has blank space
      http://www.google.com/search?q=john+smith
      An alternative to “+” is “%20”.
    • Two or more parameters, using & to link pairs
      http://finance.yahoo.com/q/bc?s=AAPL&t=2y
    • An alternative to “&” is “%26”.

• www.cs.odu.edu CGI implementation
  – cgi scripts must be stored under public_html, which is directly under your root directory
  – cgi scripts are best, but not required, stored under public_html/cgi-bin/
  – The program should have access mode 705 so that it can be executed by the web server.
  – All directory above the program must have access mode at least 701
Some simple examples of perl cgi scripts

- Generating a plain text page

```perl
#!/usr/local/bin/perl
print<<PLAIN
Content-type: text/plain
Just plain text. Nothing fancy. Got it?
PLAIN
```

You need the two blank lines as given in the code.

http://www.cs.odu.edu/~s hen/cgi-bin/plainText.cgi

- The perl script mapCoordinatesPerl.cgi

```perl
#!/usr/local/bin/perl

$queryString = $ENV{'QUERY_STRING'};
print <<END;
Content-type: text/html
<html>
<p><b>The coordinates of your clicking were:</b><p>
$queryString
</html>
END
```

This is the program that we used earlier in giving us the x, y coordinates of the points where we clicked in images used for imagemaps.