HTML Programming
Basic Skills (3)

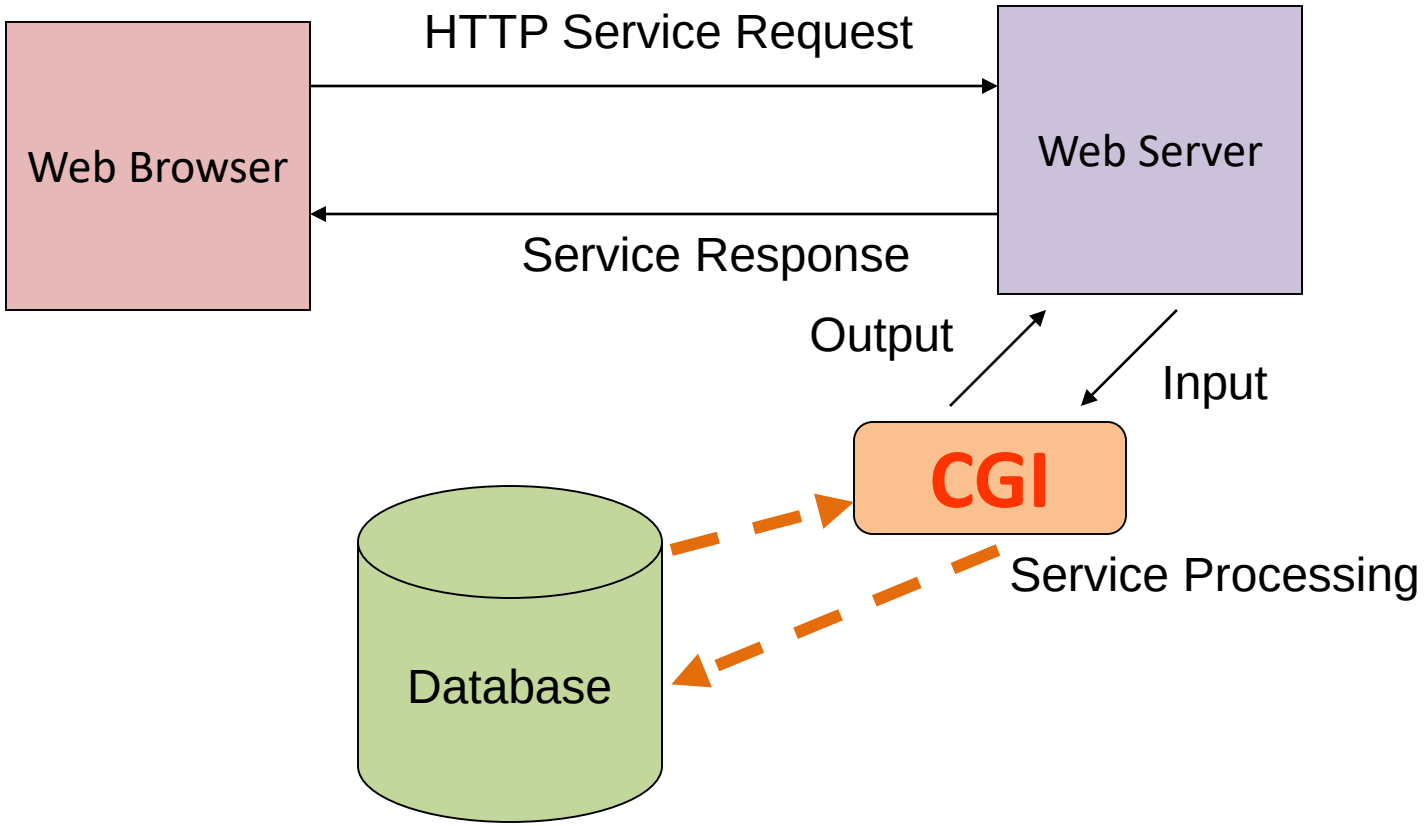
Wen-Hsiang Lu (盧文祥)
Department of Computer Science and Information Engineering,
National Cheng Kung University
2015/10/23

HTML Form and Inputs

- HTML forms are used to pass data to a server.
- A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.
  - <input type="text" name="YY" value="ZZ" size="WW">
  - <input type="password" name="YY" value="ZZ" size="WW">
- Syntax：<**form** name="XX" action="YY" method="ZZ">
  various input elements …
  </form>
  - name: form name
  - action: a form processing program (CGI script) is used to deal with the input data in the form
  - method: input data transmission methods
    - get: the CGI script receives the data in an environment variable
    - post: the CGI script receives the data in the standard input stream (stdin)
- [Ex1] part11-1.htm
- [Ex2] php1-3-form-password.html
- [Ex3] php1-4-class.php

CGI (Common Gateway Interface)

http://www.google.com.tw/search?source=ig&hl=zh-TW&q=%22travel+agenda%22&



<input> Elements

- <input> tag
- Syntax: <**input** type =“XX” name=“YY” value=“ZZ”>
  - type: eight different types of attributes
  - name: the name of the input element
  - value: default value for the name of the type attribute

Get.cgi

```
#include<stdio.h>

int main() {
    printf("Content-type: text/html; charset=utf-8\n\n");
    char *s=getenv("QUERY_STRING");//for get

    printf("GET:%s",s);
    return 0;

}
```

<input type="text">

- Function: a one-line input field that a user can enter text into
- Syntax<input **type="text"** name="YY" value="ZZ" size="WW" maxlength="TT">
  - type: text (one-line **text** input)
  - name: the name of the text
  - value: default value
  - size: the text field width (default 20 characters)
  - maxlength: the maximum length of input characters
- Example: part11-1.htm

<input type="pasword">

- Function: input password
- Syntax: <input **type="password"** name="YY" value="ZZ" size="WW" maxlength="TT">
  - type: password
  - name: the name of the password
  - value: default value
  - size: the text field width (default 20 characters)
  - maxlength: the maximum length of input characters
- Example: part11-1.htm

<input type="radio">

- Function: Radio buttons let a user select ONLY ONE of a limited number of choices
- Syntax: <input **type="radio"** name="YY" value="ZZ" checked="checked">
  - type: radio (only one choice)
  - name: the name of the radio
  - value: default value
  - checked: predefined selected item (an item per group)
- Example: part11-1.htm

<input type="checkbox">

- Function: Checkboxes let a user select ONE or MORE options of a limited number of choices.
- Syntax: <input **type="checkbox"** name="YY" value="ZZ" checked="checked">
  - type: checkbox (multiple choices)
  - name: the name of the checkbox
  - value: default value
  - checked: predefined selected items (several items per group)
- Example: part11-1.htm

<input type="reset">

- Function: remove the input data
- Syntax: <input **type="reset"** name="YY" value="ZZ">
  - type: reset (data elimination button)
  - name: the name of the reset
  - value: default value
- Example: part11-1.htm

<input type="submit">

- Function: A submit button is used to send form data to a server. The data is sent to the page (CGI script) specified in the form's action attribute.
- Syntax: <input **type="submit"** name="YY" value="ZZ">
  - type: submit (data submission button)
  - name: the name of the submit
  - value: default value
- Example: part11-1.htm

<input type="hidden">

- Function: a few hidden data which are not displayed, but are sent to the server CGI script
- Syntax: <input **type="hidden"** name="YY" value="ZZ">
  - type: hidden
  - name: the name of the hidden
  - value: default value
- Example: part11-1.htm

<input type="button">

- Function: unlike the reset and submit buttons, the buttons can trigger embedded scripts (programs)
- Syntax: <input **type="button"** name="YY" value="ZZ" onclick="FF">
  - type: button
  - name: the name of the botten
  - value: default value
  - onclick: execute the "FF" program while clicking the button
- Example: part11-2.htm

<input type=“file”>

- Function: Upload files in a client to a server
- Syntax: <input **type="file"** name="upload" value=“” size="50">
  - type: upload files
  - name: the name of the “file”
  - value: default value
  - size: the length of the filename field
- Example: part11-3.htm

<select> List

- Function: a select (pop-up) list
- Syntax: <**select** name="XX" size="N" multiple>
  - name: the name of the select element
  - size: the number of data record
  - multiple: multiple choice indicator
- <option> sub-tag: set an option item
  - Syntax: <**option** value="YY" selected="selected">
    - value: option value
    - selected: default selected option
- Example: part11-4.htm

<textarea> Tag

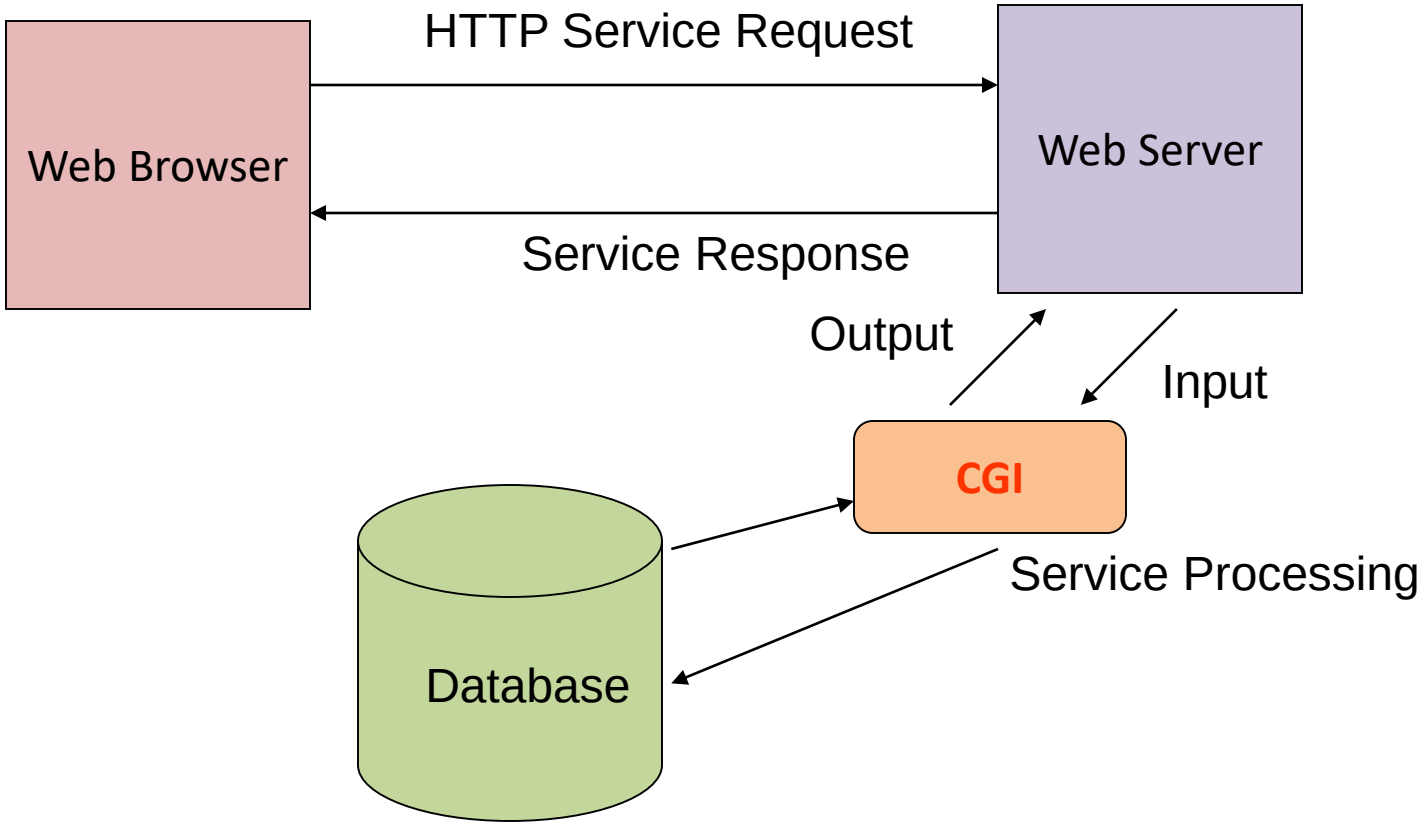
- Function: a multi-line text input control
- Syntax: <**textarea** name="XX" rows="N1" cols="N2">
  - name: the name of the textarea element
  - rows: the row number of text area
  - cols: the column number of text area
- Example: part11-4.htm

<Form>: send data to a server

- Format：<form action="YY" **method**="ZZ">
  - Action: a form processing program (CGI script) is used to deal with the input data in the form
  - Method: input data transmission methods
    - **get**: the CGI script receives the data in an environment variable
      - [CGI Example] get.c
    - **post**: the CGI script receives the data in the standard input stream (stdin)
      - [CGI Example] post.c
- Example: part11-4.htm

CGI (Common Gateway Interface)

http://www.google.com.tw/search?source=ig&hl=zh-TW&q=%22travel+agenda%22&



HTTP Request Processing

[範例] Perl language

```
sub read_query{
  %Input=();

  if ($ENV{'REQUEST_METHOD'} eq "POST"){
    read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
  }
  elsif ($ENV{'REQUEST_METHOD'} eq "GET"){
    $buffer= $ENV{'QUERY_STRING'};
  }#end elsif
  print "Input string: $buffer<br>\n";
  @pairs=split(/&/,$buffer);
  foreach $chunk (@pairs){
    $chunk=~tr/+/ /;
    ($key,$value)=split(/=/,$chunk);
    $value=~s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/eg;
    $Input{$key}=$value;
    print "$key => $value\n";
  }#end foreach
}#end read_query
```