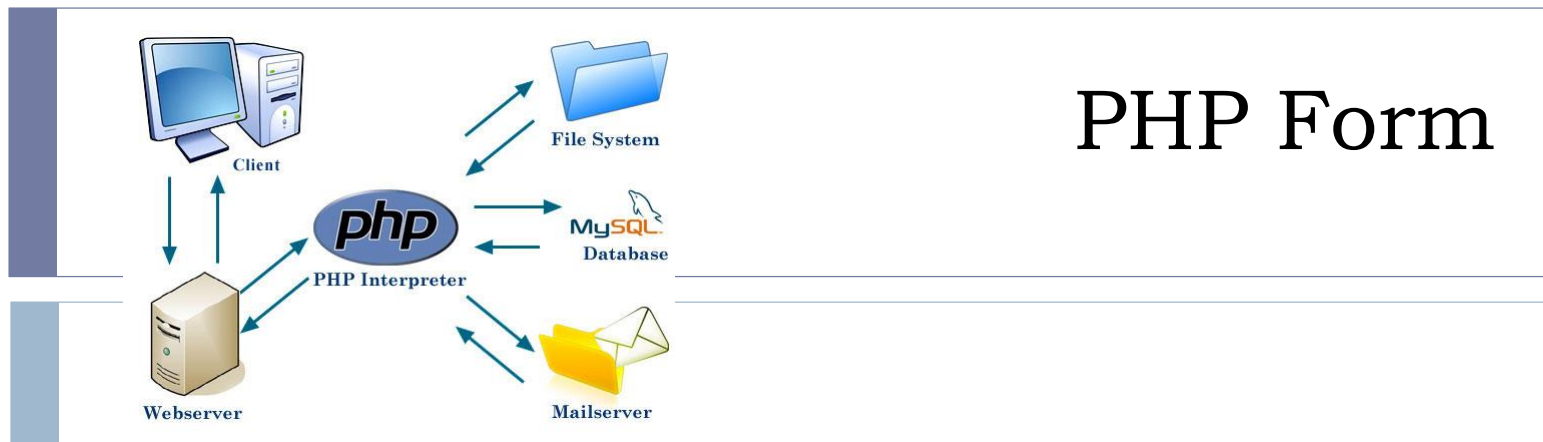


COMP 2021

Unix and Script Programming





Form Basics



Web Data



- ▶ Most interesting web pages revolve around data
 - ▶ examples: Google, IMDB, Facebook, YouTube, Amazon, Twitter
 - ▶ Can take many formats: text, HTML, XML, multimedia
- ▶ Many of them allow us to access their data
- ▶ Some even allow us to *submit our own new data*
- ▶ Most server-side web programs accept parameters that guide their execution



Passing Web Data to Server

```
URL?name=value&name=value...
```

```
http://example.com/student_login.php?username=alice&  
sid=1234567
```

- ▶ **Query string:** a set of parameters passed from a browser to a web server
 - ▶ often passed by placing **name/value pairs** at the end of a URL
- ▶ PHP code on the server can examine and utilize the value of parameters



HTML Forms

- ▶ **Form:** a group of UI controls that accepts information from the user and sends the information to a web server
- ▶ The information is sent to the server as a query string

STUDENT REGISTRATION FORM

FIRST NAME	<input type="text"/>	(max 30 characters a-z and A-Z)																									
LAST NAME	<input type="text"/>	(max 30 characters a-z and A-Z)																									
DATE OF BIRTH	Day: <input type="text"/> Month: <input type="text"/> Year: <input type="text"/>																										
EMAIL ID	<input type="text"/>																										
MOBILE NUMBER	<input type="text"/>	(10 digit number)																									
GENDER	Male <input type="radio"/> Female <input checked="" type="radio"/>																										
ADDRESS	<input type="text"/>																										
CITY	<input type="text"/>	(max 30 characters a-z and A-Z)																									
PIN CODE	<input type="text"/>	(6 digit number)																									
STATE	<input type="text"/>	(max 30 characters a-z and A-Z)																									
COUNTRY	<input type="text" value="India"/>																										
HOBBIES	Drawing <input type="checkbox"/> Singing <input type="checkbox"/> Dancing <input type="checkbox"/> Sketching <input type="checkbox"/> Others <input type="checkbox"/> <input type="text"/>																										
QUALIFICATION	<table><thead><tr><th>Sl.No.</th><th>Examination</th><th>Board</th><th>Percentage</th><th>Year of Passing</th></tr></thead><tbody><tr><td>1</td><td>Class X</td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td>2</td><td>Class XII</td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td>3</td><td>Graduation</td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td></tr><tr><td>4</td><td>Masters</td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td></tr></tbody></table>	Sl.No.	Examination	Board	Percentage	Year of Passing	1	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>	2	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>	3	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>	4	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>	(10 char max) (upto 2 decimal)
Sl.No.	Examination	Board	Percentage	Year of Passing																							
1	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>																							
2	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>																							
3	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>																							
4	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>																							
COURSES APPLIED FOR	BCA <input checked="" type="radio"/> B.Com <input type="radio"/> B.Sc <input type="radio"/> B.A <input type="radio"/>																										
<input type="button" value="Submit"/> <input type="button" value="Reset"/>																											

HTML form: <form>

```
<form action="destination URL">  
  form controls  
</form>
```

HTML

- ▶ Required action attribute gives the URL of the page that will process this form's data
- ▶ When form has been filled out and **submitted**, its data will be sent to the action's URL



Form Example

- ▶ Wrap the form's controls in a block element such as **div**

```
<h1>A Form Example </h1>
<form action="http://www.google.com/search">
  <div>
    Let's search Google:
    <input name="q" />
    <input type="submit" />
  </div>
</form>
```

HTML



The screenshot shows a web browser window with the address bar set to `http://localhost:63342/COMP2`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The page content displays the rendered HTML form: a heading "A Form Example" followed by the text "Let's search Google:" and a search form. The search form consists of a text input field and a "Submit Query" button.

Form controls: <input>

- ▶ **input element** is used to create many UI controls
 - ▶ an inline element **MUST** be self-closed
- ▶ **name** attribute specifies name of query parameter to pass to server
- ▶ **type** can be button, checkbox, file, hidden, password, radio, reset, submit, text, ...
- ▶ **value** attribute specifies control's initial text

```
<form action="http://www.google.com/search">
  <div>
    Let's search Google:
    <input type="text" name="q" value="HKUST" />
    <input type="submit" value="Go!" />
  </div>
</form>
```

Let's search Google:

Text fields: <input>

- ▶ **input attributes:** disabled, maxlength, readonly, size, value
- ▶ **size attribute** controls onscreen width of text field
- ▶ **maxlength** limits how many characters user is able to type into field

```
<form name="login">
  <div>
    ITSC Account:<br>
    <input type="text" name="itsc" size="15"
maxlength="15" /> <br><br>
    Password: <br>
    <input type="password" name="pwd" size="15"
maxlength="15" /> <br><br>
    <input type="submit" value="Log In" />
  </div>
</form>
```

Text Fields

ITSC Account:

Password:

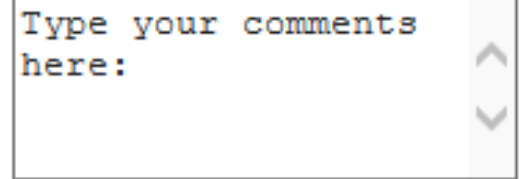
Log In

Text boxes: <textarea>

- ▶ Initial text is placed inside textarea tag (optional)
- ▶ required rows and cols attributes specify height/width in characters
- ▶ optional read only attribute means text cannot be modified

```
<textarea rows="4" cols="20">  
Type your comments here:  
</textarea>
```

HTML



Type your comments
here:

Check boxes: <input>

- ▶ None, 1, or many checkboxes can be checked at same time

```
<h2>Your major</h2>
<input type="checkbox" name="comp" checked="checked"/>
COMP
<input type="checkbox" name="cpeg" /> CPEG
<input type="checkbox" name="ece" /> ECE
```

HTML

Your major

☒ COMP ☐ CPEG ☐ ECE



Radio buttons: <input>

- ▶ Grouped by name attribute (only one can be checked at a time)
- ▶ Must specify a value for each one or else it will be sent as value on

```
<input type="radio" name="cc" value="visa"
checked="checked" /> Visa
<input type="radio" name="cc" value="mastercard" />
MasterCard
<input type="radio" name="cc" value="amex" /> American
Express
```

HTML



Text labels: <label>

- ▶ Associates nearby text with control, so you can click text to activate control
- ▶ Can be used with checkboxes or radio buttons
- ▶ *Label* element can be targeted by CSS style rules

```
<label><input type="radio" name="cc" value="visa"  
checked="checked" /> Visa</label>  
<label><input type="radio" name="cc" value="mastercard" />  
MasterCard</label>  
<label><input type="radio" name="cc" value="amex" />  
American Express</label>
```

HTML

Drop down lists: `<select>`, `<option>`

- ▶ Option element represents each choice
- ▶ Select optional attributes: disabled, multiple, size
- ▶ Optional selected attribute sets which one is initially chosen

```
<select name="favoritecharacter">
  <option>Frodo</option>
  <option>Bilbo</option>
  <option selected="selected">Gandalf</option>
  <option>Galandriel</option>
</select>
```

HTML



Using <select> for lists

```
<select name="favoritecharacter[]" size="3"
multiple="multiple">
  <option>Frodo</option>
  <option>Bilbo</option>
  <option>Gandalf</option>
  <option>Galandriel</option>
  <option selected="selected">Aragorn</option>
</select>
```

HTML

- ▶ optional multiple attribute allows selecting multiple items with shift- or ctrl-click
- ▶ must declare parameter's *name with []* if you allow multiple selections
- ▶ option tags can be set to be initially selected

Option groups: <optgroup>

```
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Frodo</option>
    <option>Sam</option>
    <option>Gandalf</option>
    <option>Aragorn</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Galandriel</option>
    <option>Bilbo</option>
  </optgroup>
</select>
```

HTML

- ▶ What should we do if we don't like the bold italic?



Reset Buttons

```
Name: <input type="text" name="name" /> <br />
Food: <input type="text" name="meal" value="pizza" /> <br />
<label>Meat? <input type="checkbox" name="meat" /></label>
<br />
<input type="reset" />
```

HTML

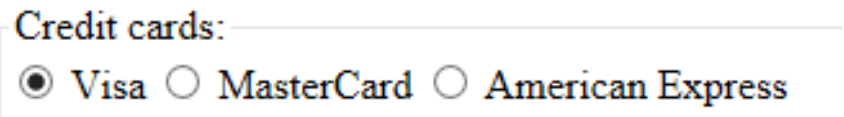
- ▶ Specify custom text on the button by setting its value attribute



Grouping input: <fieldset>, <legend>

```
<fieldset>
  <legend>Credit cards:</legend>
  <input type="radio" name="cc" value="visa"
checked="checked" /> Visa
  <input type="radio" name="cc" value="mastercard" />
MasterCard
  <input type="radio" name="cc" value="amex" />
American Express
</fieldset>
```

HTML



Credit cards:

☒ Visa ☐ MasterCard ☐ American Express

- ▶ fieldset groups related input fields, adds a border; legend supplies a caption
-



Common UI Control Errors

- ▶ “I changed the form's HTML code ... but when I refresh, the page doesn't update!”
- ▶ By default, when you refresh a page, it leaves the previous values in all form controls
 - ▶ It does this in case you were filling out a long form and needed to refresh/return to it
 - ▶ If you want it to clear out all UI controls' state and values, you must do a full refresh
 - ▶ Firefox: Shift-Ctrl-R
 - ▶ Mac: Shift-Command-R



Styling Form Controls

```
input[type="text"] {  
    background-color: yellow;  
    font-weight: bold;  
}
```

CSS

- ▶ **Attribute selector:** matches only elements that have a particular attribute value
- ▶ Useful for controls because many share the same element (input)



Hidden input parameters

```
<input type="text" name="username" /> Name <br />  
<input type="text" name="sid" /> SID <br />  
<input type="hidden" name="school" value="HKUST" />  
<input type="hidden" name="year" value="2048" />
```

HTML

- ▶ An invisible parameter that is still passed to the server when form is submitted
- ▶ Useful for passing on additional state that isn't modified by the user





Submitting Data



The value attribute

```
<label><input type="radio" name="cc" value="visa" />
Visa</label>
<label><input type="radio" name="cc" value="mastercard" />
MasterCard</label> <br />
Favorite Star Trek captain:
<select name="startrek">
    <option value="kirk">James T. Kirk</option>
    <option value="picard">Jean-Luc Picard</option>
<input type="submit" value="submit" />
</select> <br />
```

HTML

- ▶ Value attribute sets what will be submitted if a control is selected
- ▶ [cc] => visa, [startrek] => picard



URL-encoding

- ▶ Certain characters are not allowed in URL query parameters:
 - ▶ examples: " ", "/", "=", "&"
- ▶ when passing a parameter, it is URL-encoded
 - ▶ “COMP2021's cool!?” → “COMP2021%27s+cool%3F%21”
- ▶ You don't usually need to worry about this:
 - ▶ the browser automatically encodes parameters before sending them
 - ▶ The PHP `$_REQUEST` array automatically decodes them
 - ▶ ... but occasionally the encoded version does pop up (e.g. in Firebug)



Submitting data to a web server

- ▶ Though browsers mostly retrieve data, sometimes you want to submit data to a server
 - ▶ Hotmail: Send a message
 - ▶ Flickr: Upload a photo
 - ▶ Google Calendar: Create an appointment
- ▶ The data is sent in HTTP requests to the server
 - ▶ with HTML forms
 - ▶ with **Ajax** (not covered in this course)
- ▶ The data is placed into the request as parameters



HTTP GET vs. POST requests

- ▶ **GET** : asks a server for a page or data
 - ▶ If the request has parameters, they are sent in the URL as a query string
- ▶ **POST** : submits data to a web server and retrieves the server's response
 - ▶ if the request has parameters, they are embedded in the request's HTTP packet, not the URL



HTTP GET vs. POST requests (cont.)

- ▶ **For submitting data, a POST request is more appropriate than a GET**
 - ▶ GET requests embed their parameters in their URLs
 - ▶ URLs are limited in length (~ 1024 characters)
 - ▶ URLs cannot contain special characters without encoding
 - ▶ Private data in a URL can be seen or modified by users



Form POST example

```
<form action="http://localhost/app.php"
method="post">
<div>
    Name: <input type="text" name="name" /> <br />
    Food: <input type="text" name="meal" /> <br />
    <label>Meat? <input type="checkbox"
name="meat" /></label> <br />
    <input type="submit" />
</div>
</form>
```

HTML



GET or POST?

```
if ($_SERVER["REQUEST_METHOD"] == "GET") {  
    # process a GET request  
    ...  
} elseif ($_SERVER["REQUEST_METHOD"] == "POST") {  
    # process a POST request  
    ...  
}
```

PHP

- ▶ Some PHP pages process both GET and POST requests
- ▶ To find out which kind of request we are currently processing, look at the **global `$_SERVER` array's `"REQUEST_METHOD"` element**



Uploading files

```
<form  
action="http://webster.cs.washington.edu/params.php"  
method="post" enctype="multipart/form-data">  
    Upload an image as your avatar:  
    <input type="file" name="avatar" />  
    <input type="submit" value="upload"/>  
</form>
```

HTML

- ▶ Add a file upload to your form as an input tag with type of file
- ▶ Must also set the `enctype` attribute of the form



Processing Form Data in PHP

"Superglobal" Arrays

Array	Description
<u>\$_REQUEST</u>	parameters passed to any type of request
<u>\$_GET</u> , <u>\$_POST</u>	parameters passed to GET and POST requests
<u>\$_SERVER</u> , <u>\$_ENV</u>	information about the web server
<u>\$_FILES</u>	files uploaded with the web request
<u>\$_SESSION</u> , <u>\$_COOKIE</u>	"cookies" used to identify the user (seen later)

- ▶ PHP **superglobal arrays** contain information about the current request, server, etc.
- ▶ These are special kinds of arrays called associative arrays.



Recap: Associative Arrays

```
$addrbook = array();  
$addrbook["cindy"] = "3535";  
$addrbook["csoffice"] = "3528";  
...  
print "cindy's office is " . $addrbook["cindy"] .  
" .\n";
```

PHP

- ▶ associative array (a.k.a. map, dictionary, hash table) : uses non-integer indexes
- ▶ associates a particular index "key" with a value
 - ▶ key "cindy" maps to value "3535"



Example: exponents

```
<?php
    $base = $_REQUEST["base"];
    $exp = $_REQUEST["exponent"];
    $result = pow($base, $exp);
?>
<?= $base ?> ^ <?= $exp ?> = <?= $result ?>
```

PHP

- ▶ What should we do to run this with xampp?

