NATTANAI WONGVATTANA
NATTANAI@MANYTECHS.COM

Coding Structure / Syntax / Variables / Types / Operators

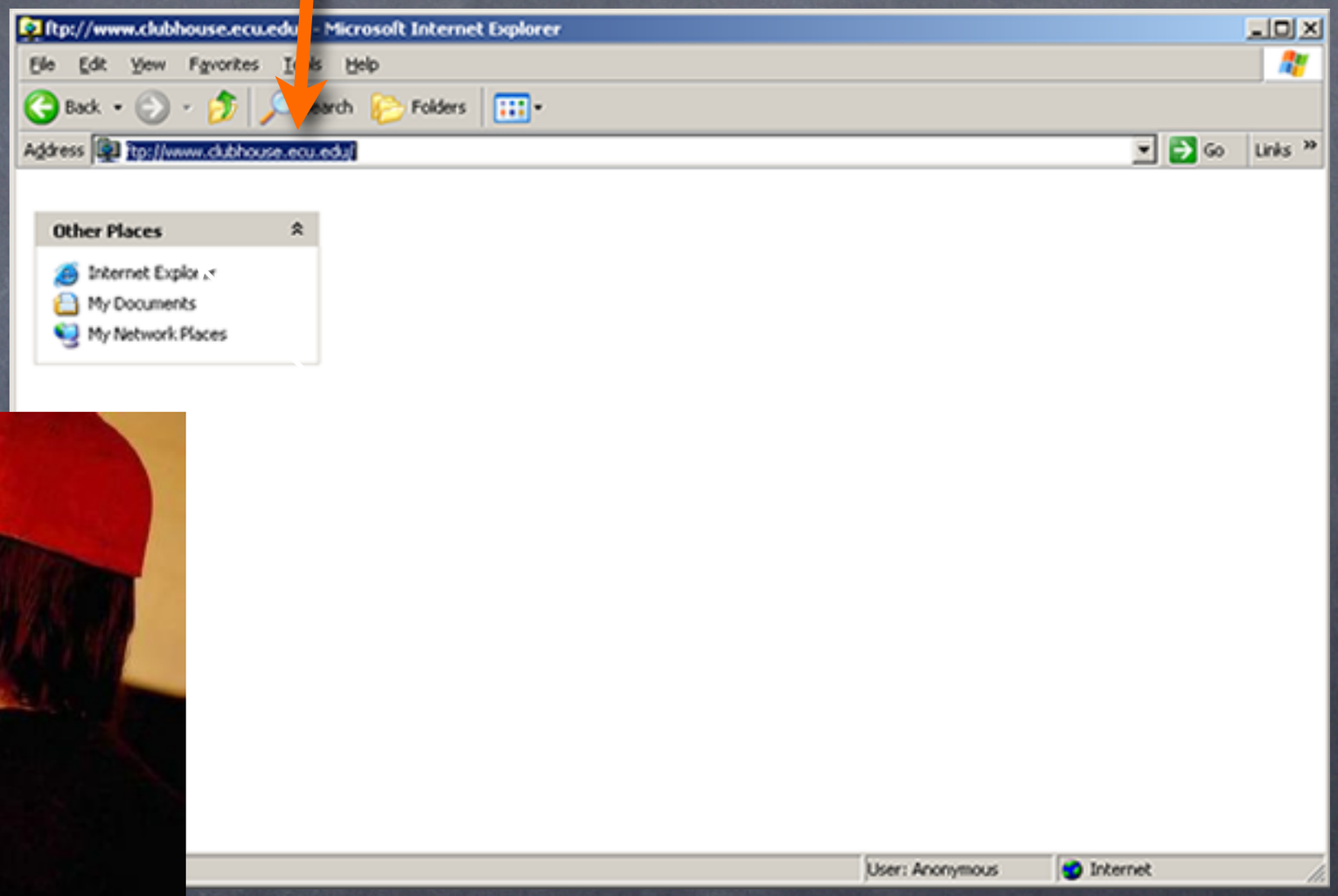
- PHP (PHP: Hypertext Preprocessor) is a server-side scripting language intended to help web developers build dynamic web sites quickly.

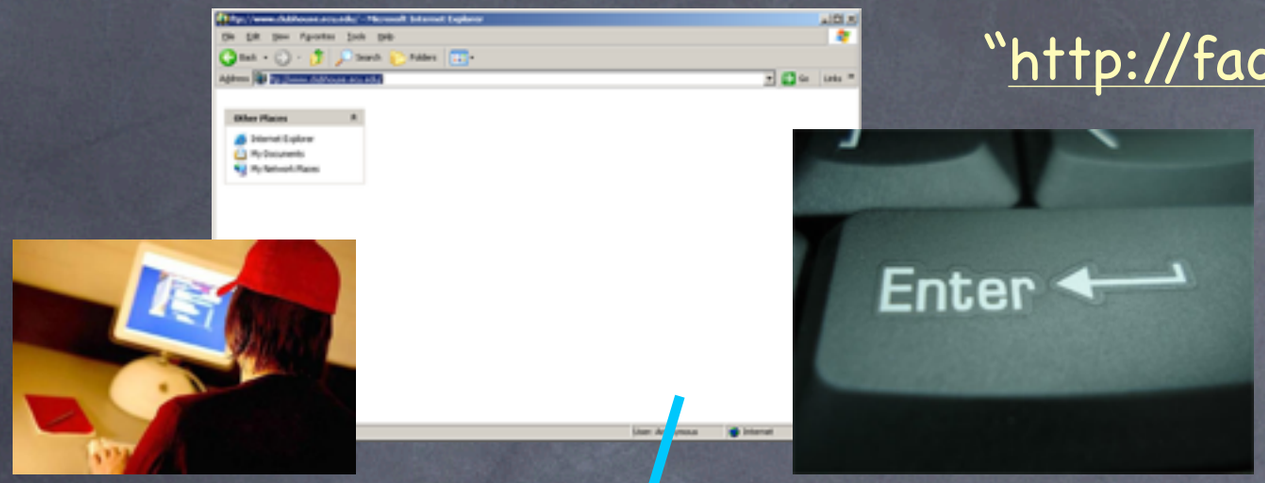
How does PHP work?

- PHP scripts are executed on the server, before the web page is displayed to the user (this is what we mean by "server-side"). The user only sees the end result, which consists of client-side markup and scripts (i.e. HTML, JavaScript, CSS etc). Therefore, the user/browser doesn't actually see any PHP code. If the user views the source code, all they would see is HTML, JavaScript, CSS etc - they wouldn't see any PHP code.
  This happens because, whenever the server processes a file with the .php extension, it knows to look for PHP code. When it encounters the PHP code, it processes it. Generally, the same .php file will also have client side code such as HTML. The server knows to process the PHP bits and output the client-side bits. You, as the programmer, determine which pieces of HTML will be displayed and when. You do this using PHP code.

PHP is scripts (lines of command)
which will be executed when web server has received REQUEST.
After all PHP scripts completely executed, web server will send the result (of executed) back to web browser as RESPONSE.

This guy enter url "http://facebook.com/profile.php?id=1234" at his web browser





"http://facebook.com/profile.php?id=1234"

When he presses enter,
The web browser connects to server which named 'facebook.com'.

Asking 'facebook.com' that
>>> 'I want to view this page - profile.php'
>>> 'And I give u additional data it is - id as 1234'

We call this as 'Request'

Request arrived

>>> 'I want to view this page - profile.php'
>>> 'And I give u additional data it is - id as 1234'

When 'Request' has arrived.

'facebook.com' server reads file named 'profile.php' which is inside its hard drive.
'facebook.com' server executes every php commands which is inside 'profile.php'.
some php commands requires variable named 'id' to be executed.
'facebook.com' server provides variable named 'id' as value '1234'.
php commands produces HTML text and stores HTML text into memory (RAM).

PHP has to be executed completely on web server before sending anything back to web browser.

'facebook.com' server sends HTML text (which is inside memory) back to that guy's web browser.
Web browser gets HTML text.
Web browser render information, images on the monitor screen.

We call this as 'Response'

Response arrived

(This slide is from first week lesson) HTTP is about Request / Response.

Web browser

1. typing http://google.com then press enter, web browser sends request to Web Server named google.com

3. send response back in HTML text or others

Web Server
(PHP inside)

2. Web server processes to response back by getting datas from database, calculate it and then write it back to Database server. In this case, google may write visiting statistic into their database.

PHP executed here, at step 2.

Database Server
(MySQL inside)

PHP Coding Structure

Creating PHP file

- Pain text file (No rich text file such as a file created with MS Word or Wordpad).
- HTML and PHP code inside.
- File has to be saved with .php extension.

(The .php extension tells the web server that it needs to process this as a PHP file. If you accidentally save it with a .html extension the server won't process your PHP code and the browser will just output it all to the screen.)

Scripting blocks

- Every block of PHP code must start with <?php and end with ?>
- <? and ?> can be used instead but some servers are not support this style so it is better to use <?php and ?>
- PHP blocks normally embed to HTML code to control how HTML text should be rendered.

Semi colon ;

- You need to place a semi-colon (;) at the end of each line of PHP code. This tells the server that a particular statement has finished.

```
<!-- example1.php -->
<html>
  <head>
     <title>PHP Syntax Example</title>
  </head>
  <body>
      <?php
        print “PHP is Pretty Hot Programming!”;
      ?>
  </body>
</html>
```

Commenting

- Use this `// Comment here` for single line comment.
- Use this `/* Comment multiple lines */` for multiple lines comment.
- Benefits to explain your idea, info, warning to other programmers who read your code.

```php
<?php
    /*
      This file name is example2.php
      Author: Nattanai Wongvattana
    */
?>
<html>
  <head>
      <title>PHP Syntax Example</title>
  </head>
  <body>
      <?php
        // show a cool word to visitor
        print “PHP is Pretty Hot Programming!”;
        print “ Isn’t it?”; // Ask visitor if they agree.
      ?>
  </body>
</html>
```

White space, Carriage return (new line), tab, etc.

- You can use tabbing, spaces, carriage returns(new line) etc to indent and format your code - this won't cause any issues for the PHP interpreter. As long as you don't forget to close each line with a semi-colon.

```
<?php
    print “Are you ”   ;   print   “ready?”;
    print
                    “ Not yet.”;
    print “ Why not?”
    ;
?>
```

Variable

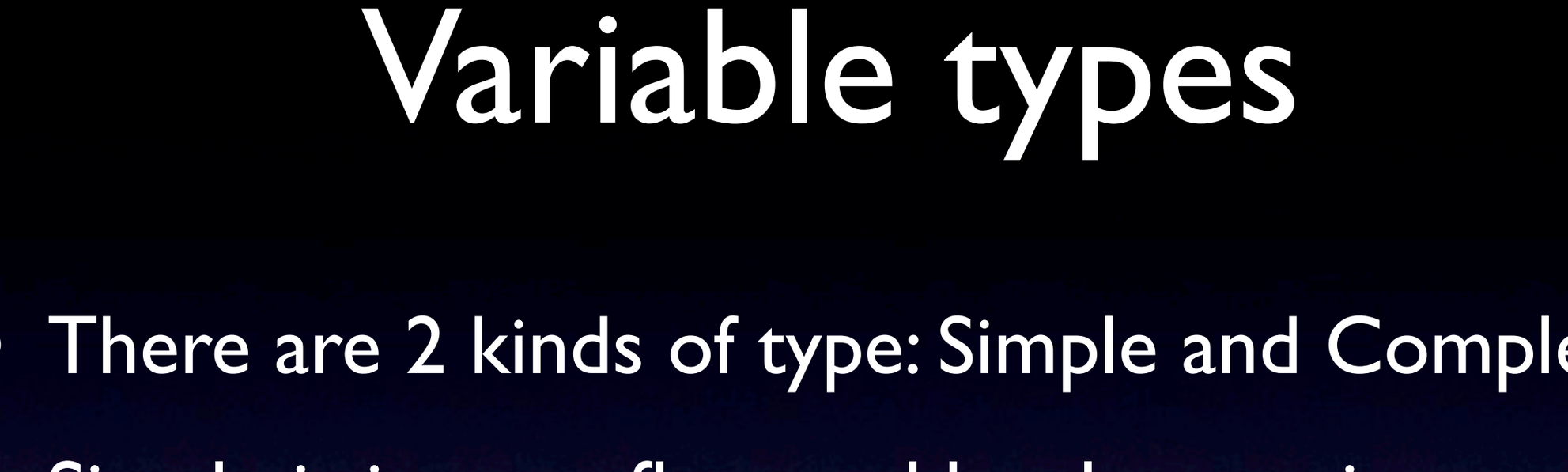
- Variables are named "containers" that allow you to store a value. This value can then be used by other parts of the application, simply by referring to the variable's name. For example, the application could display the contents of the variable to the user.

- In PHP, variable names must start with a dollar sign ($). For example:

```php
<?php
  $myVariable = "PHP is too easy for us";
  print $myVariable;
?>
```

Variable types

- There are 2 kinds of type: Simple and Complex
- Simple is integer, float and boolean, string.
- Complex is array type and object type.



Integer

- Integer variables are able to hold a whole number. Negative values can be assigned by placing the minus (-) sign after the assignment operator and before the number.

```php
<?php
    $variable1 = -2;
    $variable2 = 9;
    $variable3 = $variable1 + $variable2;
    print $variable3;
?>
```

Float

- Floating point variables contain numbers that require the use of decimal places.

```php
<?php
	$variable1 = -4234.2233;
	$variable2 = 0.0999;
	$variable3 = $variable1 * $variable2;
	print $variable3;
?>
```

Boolean

- PHP boolean type variables hold a value of true or false and are used to test conditions such as whether some part of a script was performed correctly.

```php
<?php
    $variable1 = true;
    $variable2 = false;
    $variable3 = 1;
    $variable4 = 0;
    $variable5 = $variable1 && $variable4;
    print $variable5;
?>
```

String

- The string variable type is used to hold strings of characters such as words and sentences.
- A string can be assigned to variable either by surrounding it in single quotes (') or double quotes ("). If your string itself contains either single or double quotes you should use the opposite to wrap the string:

```
<?php
    $variable1 = “Hello Tony”;
    $variable2 = ‘Hello Tony’;
    $variable3 = “Hello ‘Tony’ ”;
    $variable4 = ‘Hello “Tony” ’;
?>
```

Array

- PHP Arrays provide a way to group together many variables such that they can be referenced and manipulated using a single variable. An array is list of variables.

```php
<?php
  $ar1 = array(); // Creating empty array (nothing inside)
  // Create an array with various types inside
  $ar2 = array(1, 2.5, 'Dog', false);
  // Access and print value from array slot#2
  print $ar2[2];
  print '<br />'; // For visibility, add new line looks
  $ar3 = array('A'=>80, 'B'=>70, 'F'=>'Fail');
  // Access and print value from array slot named 'B'
  print $ar3['B'];
?>
```

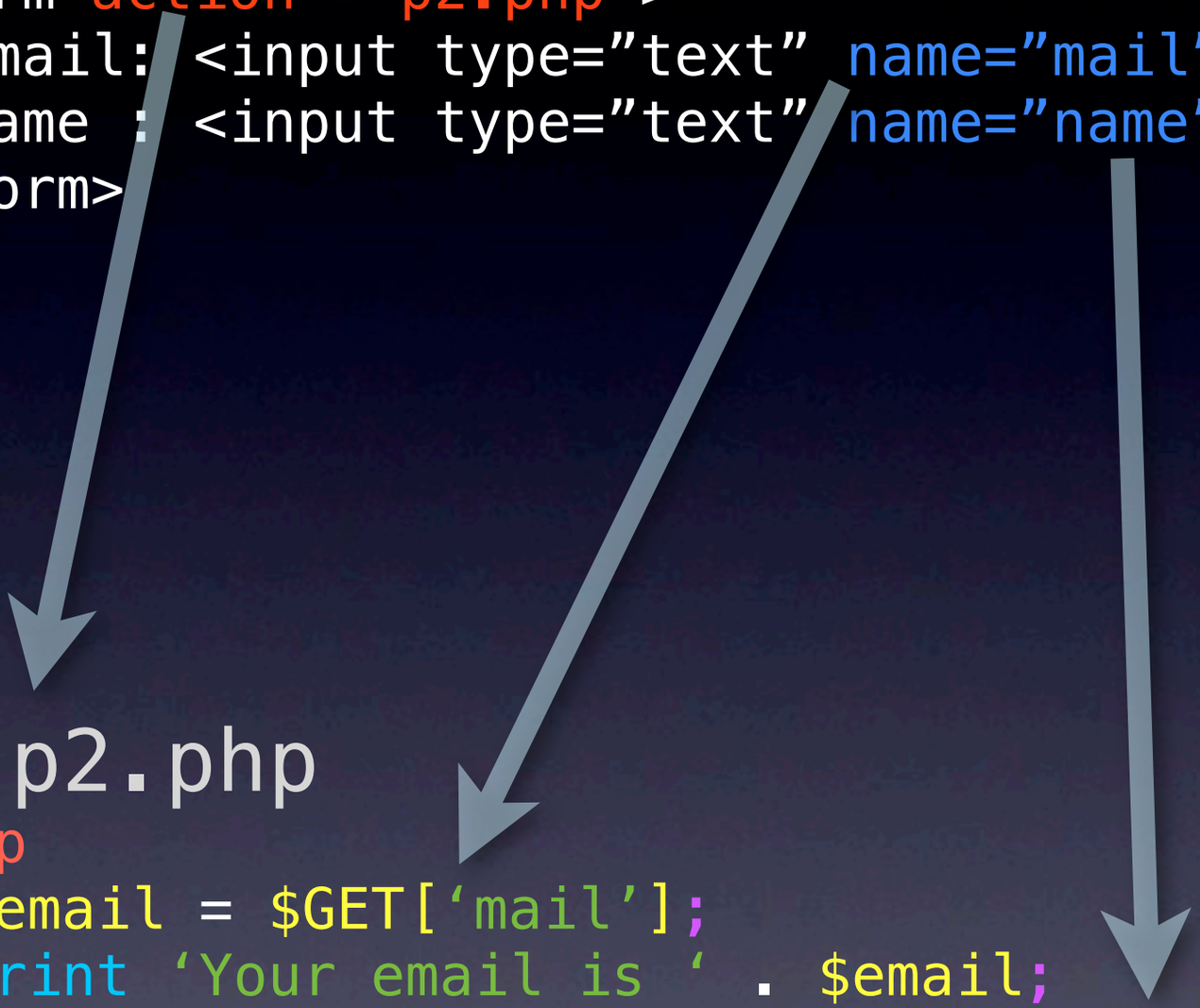
Operators

Operator	Description	Example	Result
+	Addition	23 + 7	30
-	Subtraction	90.34 - 1.11	89.23
*	Multiplication	34232.43434 * 0	0
/	Division	15 / 3	5
%	Modulus (Division remainder)	10 % 3	1
++	Increment	$a = 99; ++$a	100
--	Decrement	$b = 50; --$b;	49
==, !=	is equal to / is not equal to	80 == 6	FALSE
> , <	is greater than / is less than	7 > 3	TRUE
&&	and	(5==5 && 6==6)	TRUE
||	or	(5==5 || 6==8)	FALSE
!	not	!(0 > 1)	TRUE
.	string concatenation	PH' . 'P' . "!"	PHP!

Submitting FORM data to PHP page

- You have 2 php files.
- p1.php is a sending page
- p2.php is a receiving page
- p1.php has FORM tag with action=”p2.php” attribute
- p1.php has input tags with name=”anyname” attribute
- p2.php has php code to get datas which will be submitted from p1.php
- p2.php gets datas by referring to $_GET[‘anyname’]

```
// p1.php
<form action =”p2.php”>
  Email: <input type=”text” name=”mail” />
  Name : <input type=”text” name=”name” />
</form>
```



```
// p2.php
<?php
    $email = $GET[‘mail’];
    print ‘Your email is ‘ . $email;
    print ‘ and your name is ’ . $GET[‘name’];
?>
```

Assignment

- Create a simple trip planner which has ability to calculate cost of activities and budget
- There are 8 activities cost calculation support
- Main goal is to make sure budget is enough for all activities
- Use knowledge from first week lesson to this week lesson to create it.
- Finishing by next Tuesday will gain 2 points of bonus (Next Tuesday is deadline)

- Free to make it looks nicer, works better as you want. This may gain more points
- Copying a whole system from internet is not allowed but copying some small parts is allowed (such as code snippet). Instructor may ask you to explain about your program flow. So if you understand what you programed, it is OK to copy.
- If you want a presenter for your trip planner, recommend 'Super Kute'
- Make your code easy to understand by using comment.
- Zip all your files and send to codeenjoy@gmail.com by next Tuesday.

Trip Planner

Trip to South Africa

Your Budget = 400000 THB

To do	Cost
Air ticket	120000 THB
Hotel for 3 nights	80000 THB
World cup match	50000 THB
Big Lunch	20000 THB
Coffee shop	5000 THB
Massage	1800 THB
	THB
	THB

Check your budget

123200

Sample at http://istudy.tk/trip-planner