

2nd Half of Class
Start material after midsem

Oct. 26
2017
C53 010

PHP: Server side Language

- We are going to write responders to form actions
- Ch. 9 - PHP - object, dynamic typing

This can even be used in the middle of HTML

Must use extension php

- Types: Integer; Double = Floats; Strings; Boolean

- 2 Forms for String constants

① ' ' or ② " " ② Allows expression ^{Interpolation} ~~Interpolation~~

① What you see is what you get

#(expression) will be replaced by the expression value

For single var \$var example: "the sum is \$total \n"
↳ "the sum is #(\$total/\$count)"

- Variable are standard form except they always start w/ \$
implicitly declared on use

- Expression: number +, -, *, /, %, ++, --

• mixed arithmetic, cast to double

• if int division doesn't give exact value, the answer is double

Functions: abs, floor, ceil, sin, cos, min, max, round(V₁, V₂)
↳ round gives pseudo # between

- String - 1 operator concatenation. Example "2" + "3" = 5
"2", "3" = "23"

- We can reference individual character name \$per[7] (0-based position)

Function strlen(str) ~~XXX~~ strcmp(str1, str2) → returns an int

↳ strpos(str1, str2) 0 if same; ⊕ # if str1 > str2; ⊖ # if str1 < str2

start pos of str1 in str2

C53010 page 2

10/25/07

str to lower (str)
str to upper (str)

= function call

chomp
trim (str)
ltrim

=> remove white space from end
" " both ends
" " beginning

- Assignment: std c/c++ and ic

- Output: When used as a form responder, the output of pltr is sent to the client

zdgxc@mail.umsl.edu

10/31/17
C53010

PHP - Server side language; dynamic typing

- Standard variables; std arithmetic (C, C++ operators); std arguments
- `print string`; [free format]
- `printf("format", expression, expression)`

String

%#1#2f print wrapping expression as float w/ #1 total position #2 digit after decimal

%d int

%s string

Control structures: std boolean operators; std comparators

→ std boolean operators `&&`; `||`; `!`; plus and, or

• Decision std if/else if/else - the body are single statement

• std switch/case if body is bigger, `{ }`

• Loops - std while; do-while; for loops

• Arrays - keys → values; the keys can be strings, ~~array name~~ ^{into being}
array were into being by assigning to an element

`$A["Mo"] = 43;`

• Array constructor → `$B = array (key ⇒ value, ...)`

example: `$capitals = array ("co" ⇒ "denver", "SD" ⇒ "Pierre" ...);`

Access `$var[key]` ~~XXX~~ parallel access = `$`

`unset (array-ref);` remove element ~~XXX~~ `unset (array-name);` remove array

- `array_keys (arrayname)` → returns an array of keys; indexed to 0 →
- `array_values (arrayname)` → return array of values; indexed 0 →
- `array_key_exists (key)` → return true if valid key / false else

PHP

11/2/17

CS 3010

Arrays - Associative Arrays

keys can be strings & no space allowed for mixed keys

`$arr = array_keys($array)`

`$arr = array_values($array)`

`sizeof($array)`

`T > array_key_exists($key, $array)`
`F > array_key_exists($key, $array)`

a) Want to go through all elements in array?

• It is useful to step through all array elements. May not know all keys
2 schemes:

a) `current($array)` "points" to the "current" element; returns value
`next($array)` moves current to next item; returns the new current value

• using only `next()` never returns the first value

b) `each($array)` → return an array of (key, value) then move current
return false (not an array) @ end.

`is_array($var)` — T if array

F if not array

Functions → function names are NOT case sensitive.

Function definition: `function name(parameters) { stuff }`

• the `return [expression];`

• the number of call parameters need not match number of defined

• Variable Scope: All functions are local in extent (like C/C++) by default

• You can declare a variable as global: `global $var` // inherits that var from

• Normally the lifetime of var is the span of function execution (outside)
when function exists, the variable vanishes. Sometimes you want to keep a value
between invocations. Example: random number

`X = (X * 32794 + 7170) % 65536`

`return X`

BACK

11/2/17

CS3010 Page 2

- You can declare a variable static: `static $var`
- `var` is writeable in function only & its value is kept between invocations
- Call method: parameters are passed by value.
pass by value for arrays is passing address of the array, so array contents can be changed & be writeable outside.

↳ I need the ability to have changes to an argument be seen by caller

↳ Example: `swap(a, b)` // after call, a & b values interchange

- define formal parameter to be reference by prepending `&`
- call function w/ a reference `swap(&a, &b)`

function `swap(x, y)`

- PHP is often used to write the server program to handle the form data.

↳ In the form tag `<form action="url for server page">`

↳ For PHP this must end in `.php`

- This file can contain PHP code enclosed in a `<?php>` A `<?>`

② std html outside of that

- The doc returned by the server is the html code & the output from the PHP sections

Databases: a database lies behind many websites

Nov 14, 2017
CS3010

- A database is a storage block of information
- A database manager is a program to get data w/ arbitrary constraints efficiently.
- There are many database managers → SQL; mySQL; postgres SQL
- The language to extract data is pretty standard
- The language is SQL; this views a database as a set of relations. "Relational Model"
- table

→ The database on Hware is mySQL: db name: CS3010

username: CS3010 password: 30User10

from command line, no space

mysql -u CS3010 -p30User10 CS3010
 user pwd db name

Statements: SELECT field to print (, separated)
 FROM table (, separated)
 WHERE condition ; (1)

Example:

Databases - MySQL

SQL SELECT retrieval

UPDATE

INSERT

DELETE

PHP to access a database

CS3010

Nov. 28. 2017

$\$set = \text{html_specialchars}(\$str)$

convert $\<, >$ to ~~char~~ ^{char} entities

- problem w/ quote in input ($\$_GET$) ($\$_POST$)

- PHP is set up to put \backslash before all the quote

- When you form your query... the quote get in the way

Example - `SELECT... FROM... where name = \"George\";`

and back slash are not valid

- the ~~function~~ `$\$str = \text{strip_slashes}(\$str)$` remove back slashes.

- DB Access

① `$\$db = \text{mysqli_connect}(\$host, \$user, \$pass, \$dbname);$`

`"local host" "CS3010" "30User10" "CS3010"`

The return value is "mysqli" and will be used a lot

`$\text{mysqli_connect_error}()$` The problem; if it not

② Build your request PHP string, manipulation

③ `$\$result = \text{mysqli_query}(\$db, \$query);$`

④ `$\$numrows = \text{mysqli_num_rows}(\$result);$`

`$\text{mysqli_num_fields}(\$result);$`

`$\$row = \text{mysqli_fetch_assoc}(\$result);$`

successive calls get further rows, return false if no more

$\$row$ is an array indexed by field name

⑤ might look like

BACK

might look like:

```
$num_rows = my $qli - num_rows ($result)
$num_fields = my $qli - num_fields ($result)
$row = my $qli - fetch_assoc ($result)
// display And name $keys = array_keys ($row)
// Print the keys into HTML result
for ($i=0, $i < $num_rows; $i++) {
    $vals = array_values ($row)
    // print the array
    print "<br />"
    $row = my $qli - fetch_assoc ($result)
    my $qli - close ($db)
```

- Mark effort in step 2

example: using CS3010 DB

name, address with either or both or neither of

① CPU name =

② speed =

\$ GET cpu name =

speed =

relation = Ut, Get, Error

NEXT PAGE

Nov. 28, 2017
C53010

- the request

\$request = "SELECT name, addr FROM customer"

\$where = "WHERE TRUE"

if \$GET[cpu_name] != "." || \$GET[memory] != "."

\$request = \$request, " order;

\$where = "customer.cust_no = order.customer"

if \$GET(pname) =

\$where = \$where, "All.cpu_name = " \$GET(cpu_name)

DB + PHP → function ^{FOR} my SQL

Nov 30, 2017
CS 3010

Part 1 ① \$db = mysqli_connect

Part 2 ② Build the query → std PHP string manipulation

③ \$result = mysqli_query(\$db, \$query)

\$result = assoc array (indexed by SQL field name)

④ Display → mysqli_fetch_assoc // mysqli_num_rows(\$result)
mysqli_num_fields(\$result)

std PHP to extract data (row by row) from result

- Python ver. 2.7 (deliberate) * indentation is significant - it replaces need for braces
 - number = number □ float = double □ int = long int
 - (big)int = arb length □ std arithmetic function

- caution: if an expression is too long for a line, split after an operation of middle of ()

- assignment or assignment → operator = ; extended

vars = list

example: a, b, c = 3, 2.7, 92 + b

- print is different between 2 & 3
 ↖ python 2 → print _____
 ↗ python 3 → print() _____

Python: scripting language not compiled, run interactively

Use shebang line: `#!/bin/python`

CS 3010

December 5, 2017

- Std types (but no typing) - vars NOT declared
assignment = assignment ~~XXX~~ "=" operator

- if ~~if~~ condition:

→ then body

elif condition:

→ elif body

else

→ else body

All 3 bodies must be indented within any 1 body, indentation must be consistent

- loops → while

→ while condition:

→ body

else:

body

else is executed when the loop exits normally (without a break)

- For loop - iterate over a set of values

for ^{var}target in value

→ body

else:

body

- usually the value is a [expression]

→ example: for i in [1, 3, 5, 7, 9, 11]:

→ example: for name in ["honda", "Audi", "Kia", "Jaguar"]

- functions

def name(args):

→ body

return value.

call → name(actuals)

- Final next Tuesday - 6 - 7 questions
1 question from exclusively 1st half - essay - 15h

Dec. 7, 2017
CS 3010

- Ajax → Be able to update parts of web page w/o updating rest of page.
- Databases share a common access language → SQL
 - select - Lets you pull out of database
 - insert - Lets you change & delete from database
- MySQL → ~~PHP~~
- PHP forces you to do good formatting. Forced consistent indentation

10 AM TUESDAY