CS 5142
Scripting Languages

9/30/2013

Server-Side Scripting (PHP)

Administrative

Announcements

Prelim 2 on Friday, 11/01/2013 in class.

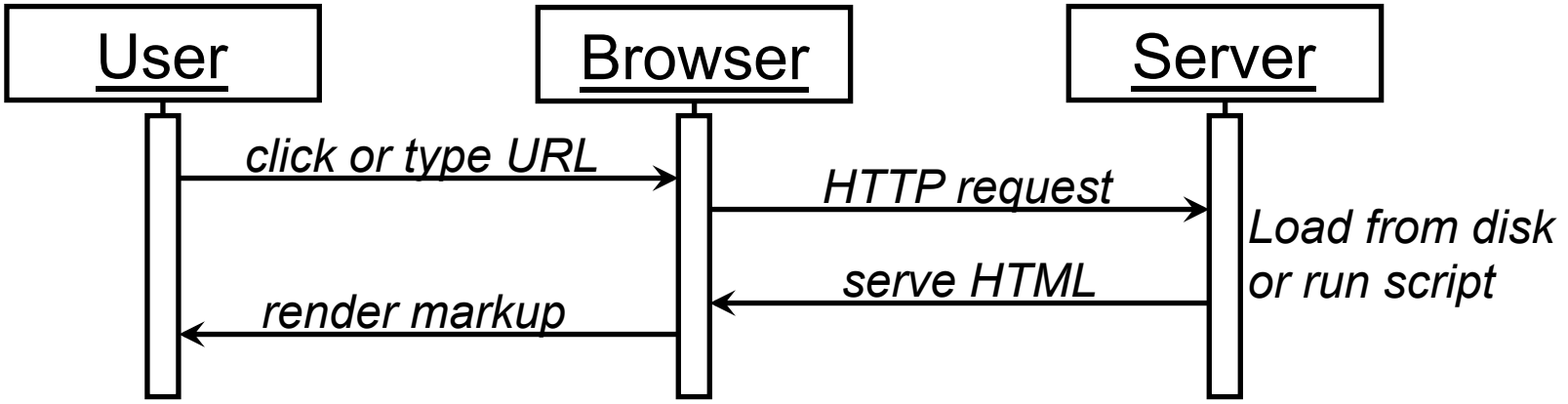
Final Exam on Tuesday, 12/17/2013 at 7pm.

Outline

- HTML Basics
- PHP Basics

About HTML

- HyperText Markup Language
  - HyperText = contains clickable links
  - Markup Language = structuring and rendering instructions
- Used for most of the internet
  - Static HTML = simple file retrieved from server's disk, or generated by server-side script, e.g., PHP
  - Dynamic HTML = HTML containing client-side script



Related Languages

- Markup languages: troff, LaTeX, texinfo
- 1991, Tim Berners-Lee published description of HTML as application of SGML
- HTTP = HyperText Transfer Protocol
- Standards vs. Browser implementations
- 1996, XML = eXtensible Markup Language
  - Simpler variant of SGML
  - Used to define markup languages, such as HTML
- 2000, XHTML = HTML that conforms to XML
- 2012, HTML5 = Subsume HTML4 and XHTML

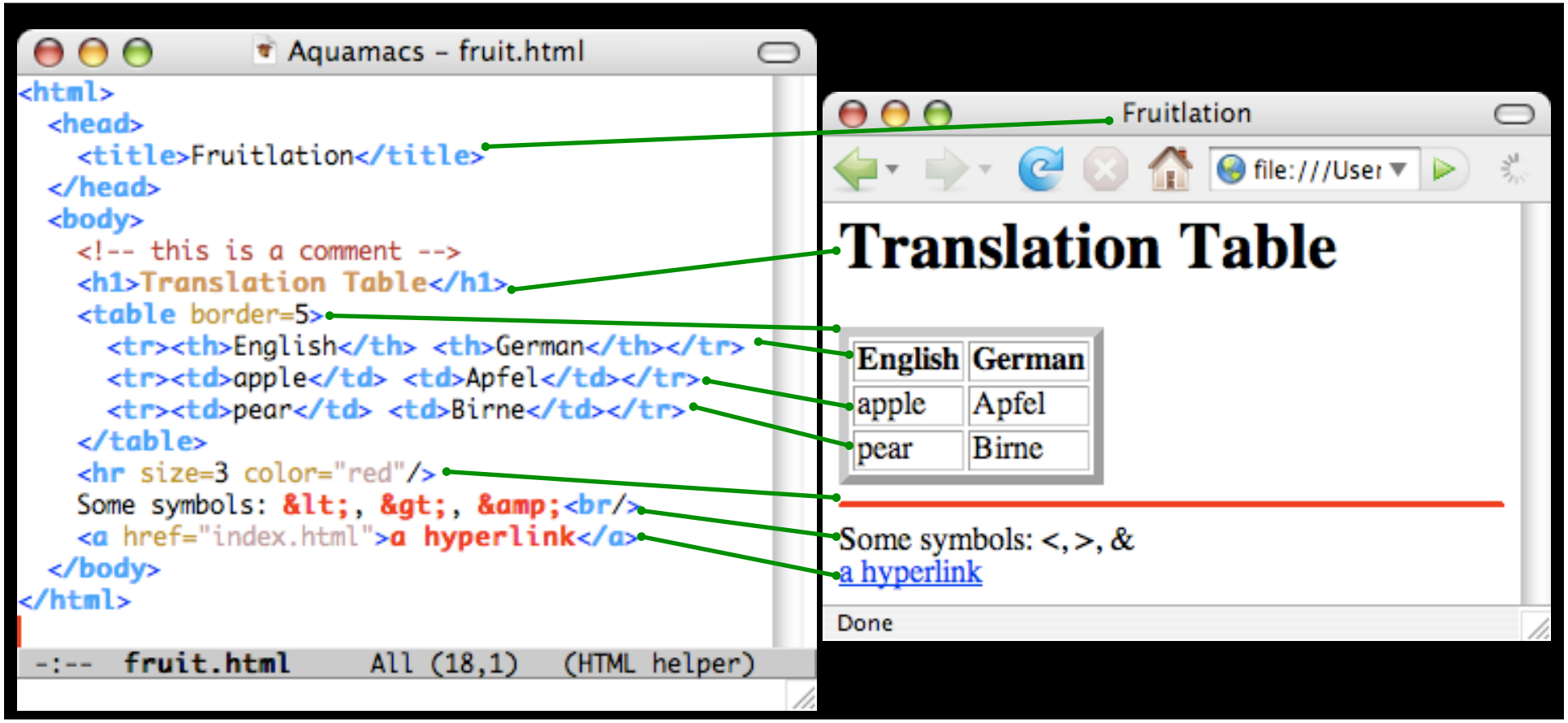
How to Write + Run Code

- Install VPN:
  http://www.it.cornell.edu/services/vpn/howto/install.cfm
- Create web site:

```
ssh en-cs-cs5142.coecis.cornell.edu
mkdir $HOME/public_html $HOME/public_html/php
chmod 701 $HOME $HOME/public_html $HOME/public_html/php
cd $HOME/public_html/php
echo '<h1>Welcome to ' `pwd` '</h1>' > index.html
chmod 604 index.html
```

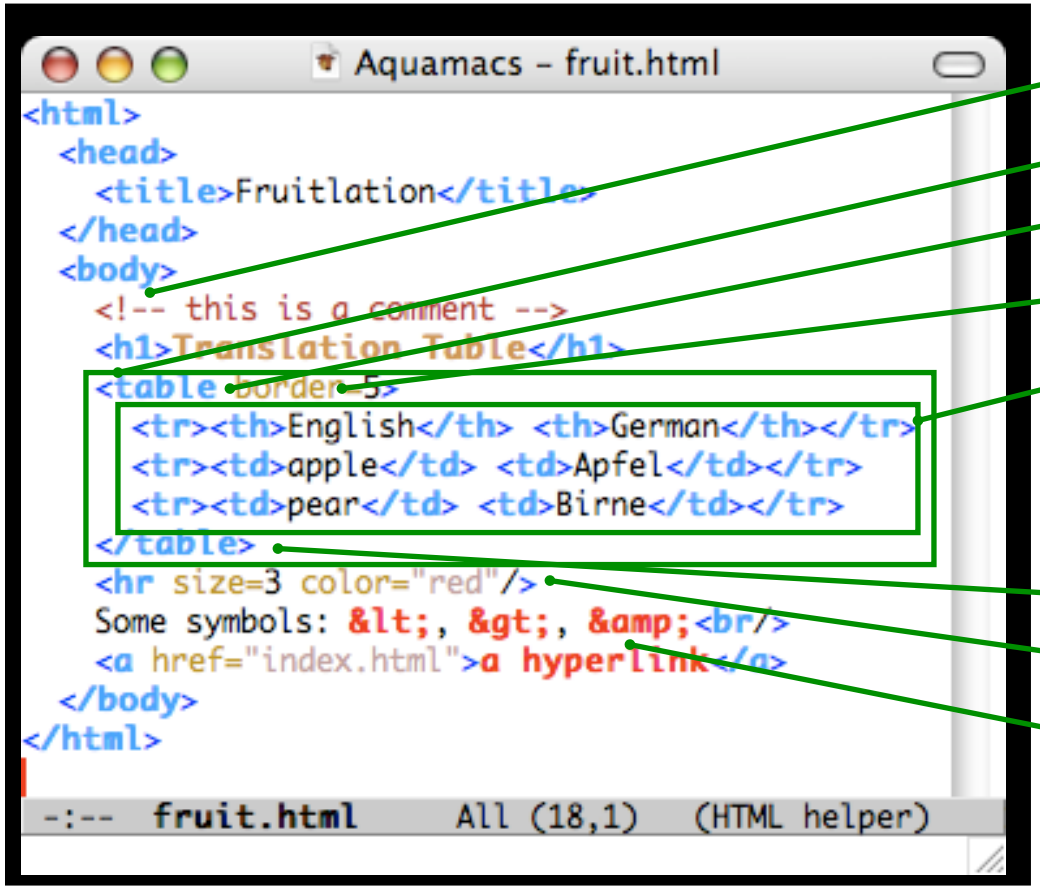
- Use a web browser to look at your web site:
  http://en-cs-cs5142.coecis.cornell.edu/~*<netid>*/php/index.html
  http://en-cs-cs5142.coecis.cornell.edu/~*<netid>*/php/

Structure of an HTML Document



Text Editor (e.g. Emacs) Web Browser (e.g. Firefox)

Lexical Peculiarities



Comment `<!-- … -->`

Element `<table>…</table>`

Start tag `<table>`

Attribute `border=5`

Contents (between matching tags)

End tag `</table>`

Empty element `<hr …/>`

Entity `&`

Reference

HTML Documentation

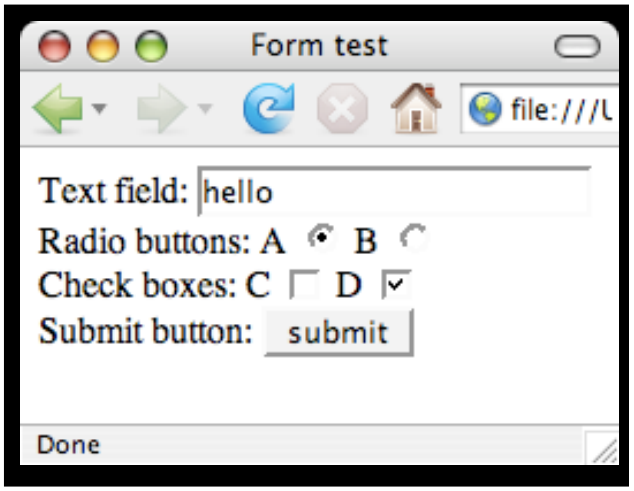
- http://www.w3schools.com/html/
-

html	document	i	italic
head, title	header	a href	hyperlink
body	main part	ul	• unordered list
h1, h2, h3	headings	ol	1. ordered list
p	paragraph	li	list entry
br	line break	table	table
hr	horizontal line	tr	table row
pre	preformatted	td	table data
b	bold	th	table head

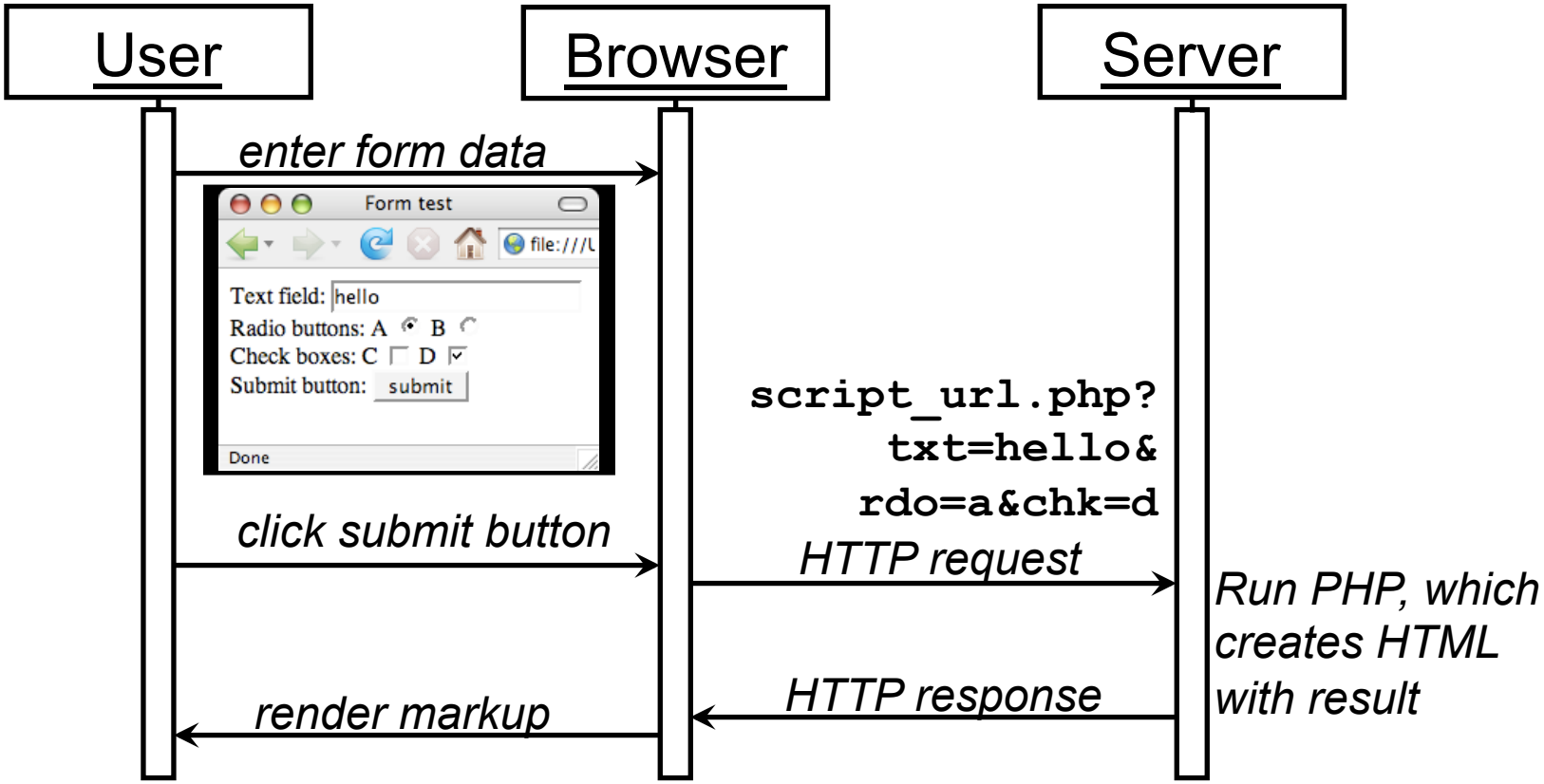
HTML

Input and Output

```
<html>
  <head>
    <title>Form test</title>
  </head>
  <body>
    <form name="frmTest" action="script_url.php"
        method="get"> <!-- or: method="post" -->
      Text field: <input type="text" name="txt">
      <br/>
      Radio buttons:
      A <input type="radio" name="rdo" value="a"/>
      B <input type="radio" name="rdo" value="b"/>
      <br/>
      Check boxes:
      C <input type="checkbox" name="chk" value="c"/>
      D <input type="checkbox" name="chk" value="d"/>
      <br/>
      Submit button:
      <input type="submit" value="submit"/>
    </form>
  </body>
</html>
```



HyperText Transfer Protocol



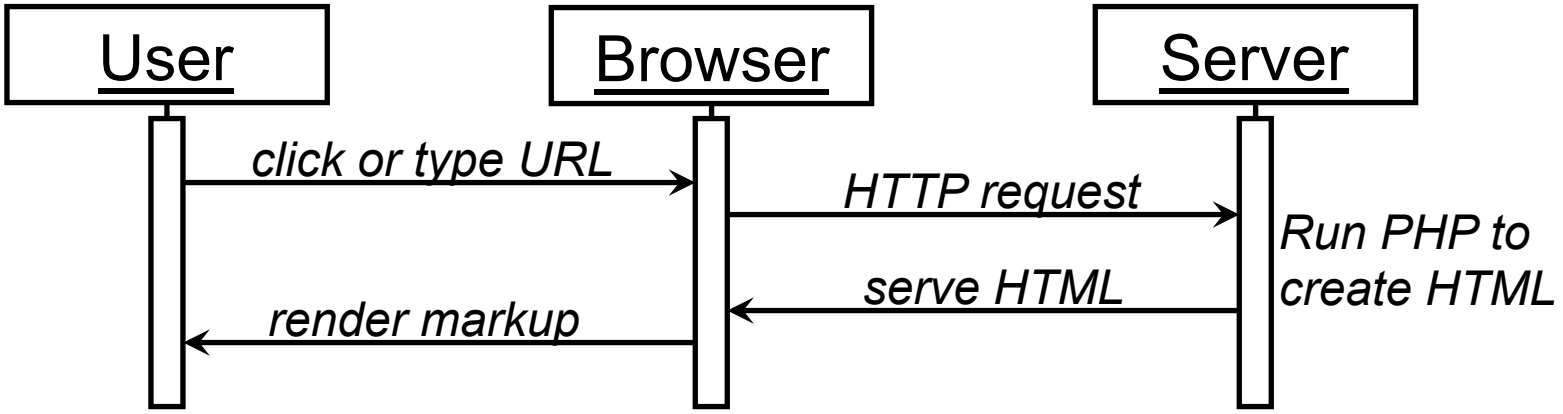
- HTTP GET: parameters encoded in URL
- HTTP POST: parameters in message header

Outline

- HTML Basics
- PHP Basics

About PHP

- PHP: Hypertext Processor
  - Recursive acronym
  - HTML = HyperText Markup Language
- Claim to fame: simplicity
  - Script embedded in HTML generates HTML
  - Libraries for MySQL, Oracle, PDF, XML



Related Languages

June 1995:	PHP = Personal Home Page Tools, Rasmus Lerdorf, CGI scripts written in C
April 1996:	PHP/FI scripting language, Rasmus Lerdorf
June 1998:	Engine rewritten by Zeev Suraski and Andy Gutsmans in Tel Aviv
May 2000:	Parser rewritten again: “Zend Engine”
May 2004:	PHP 5, current language version

- Evolution from script collection to scripting language
- Other server-side languages embedded in HTML: VisualBasic (ASP), Java (JSP)

Secure Your Website!

- Put the following in **$HOME/public_html/php/.htaccess**:

```
AuthType Basic
AuthUserFile /home/<netid>/.htpasswd
AuthName "Members ONLY"
require valid-user
```

- Create a user name and password:

```
/usr/bin/htpasswd -c $HOME/.htpasswd user_name
# make up a new password for user_name, don't forget it
chmod 604 $HOME/.htpasswd $HOME/public_html/php/.htaccess
```

- Use a web browser, that will request authorization:
  http://en-cs-cs5142.coecis.cornell.edu/~<netid>/php/index.html

How to Write + Run Code

- Put the following in **$HOME/public_html/php/hello.php**:

```
<html><body>
<?php
  if(!empty($_GET['who'])) { echo "Hi, {$_GET['who']}.";}
?>
<form action="<?php echo $_SERVER[ PHP_SELF]; ?>"
  method=get>
  Who shall be greeted: <input type="text" name="who" />
</form>
</body></html>
```

- Set the permissions:

```
chmod 604 $HOME/public_html/php/hello.php
```

- Use a web browser to look at your php script:
  http://en-cs-cs5142.coecis.cornell.edu/~*<netid>*/php/hello.php

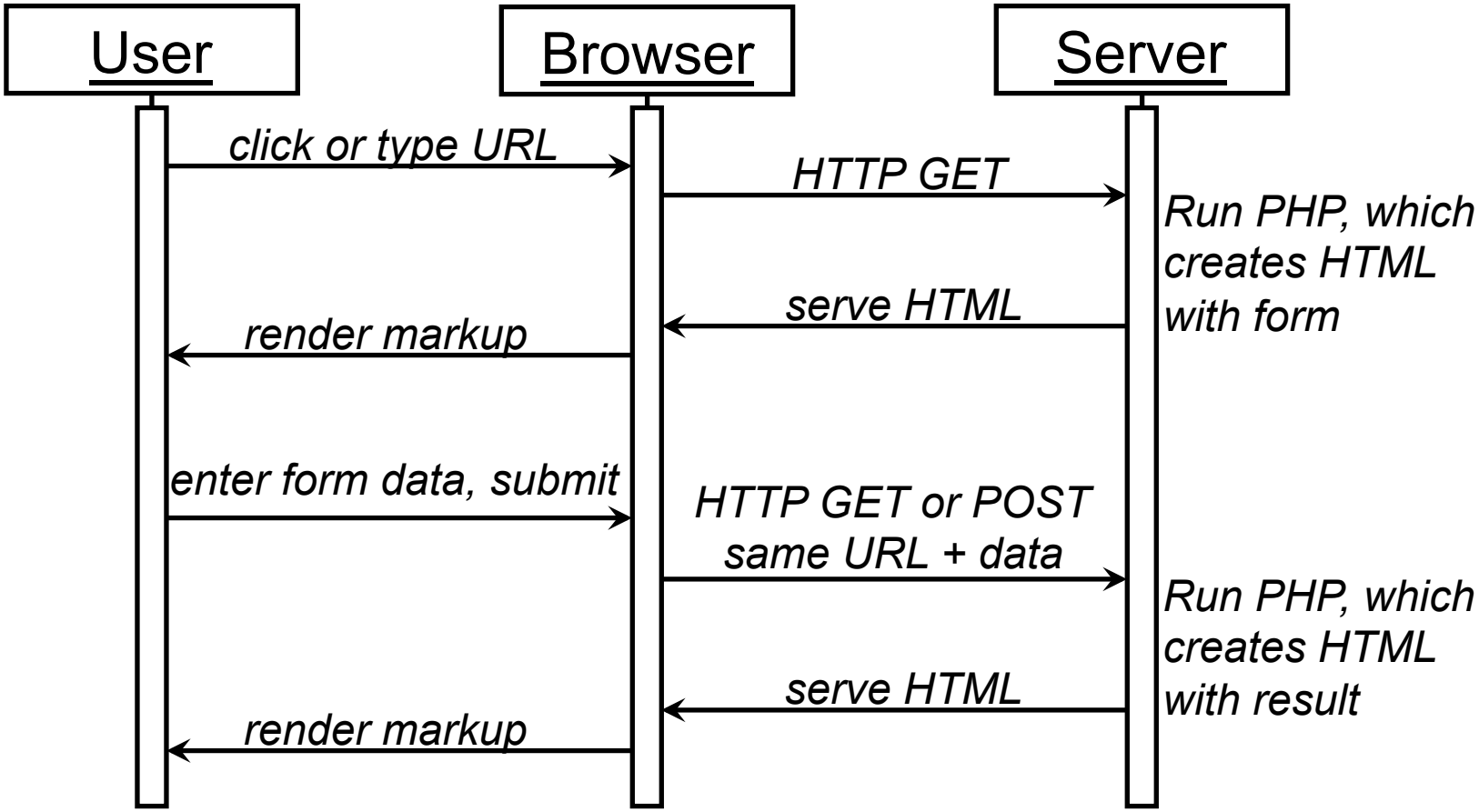
Embedding Code in Web Pages

- Browser doesn't see PHP script, only result of running script on server
- PHP code gets replaced by its own output (printed by "**`echo`**" and other functions)
- Four styles of embedding PHP:
  - XML style (preferred!) **`<?php … ?>`**
  - SGML style **`<? … ?>`**
  - ASP style **`<% … %>`**
  - Script style (deprecated)

```
<script language="php"> … </script>
```

- Variant of SGML style: **`<?=`** … **`?>`** is shorthand for **`<? echo … ?>`**

Self-Processing Pages



- Same PHP script, different HTML response

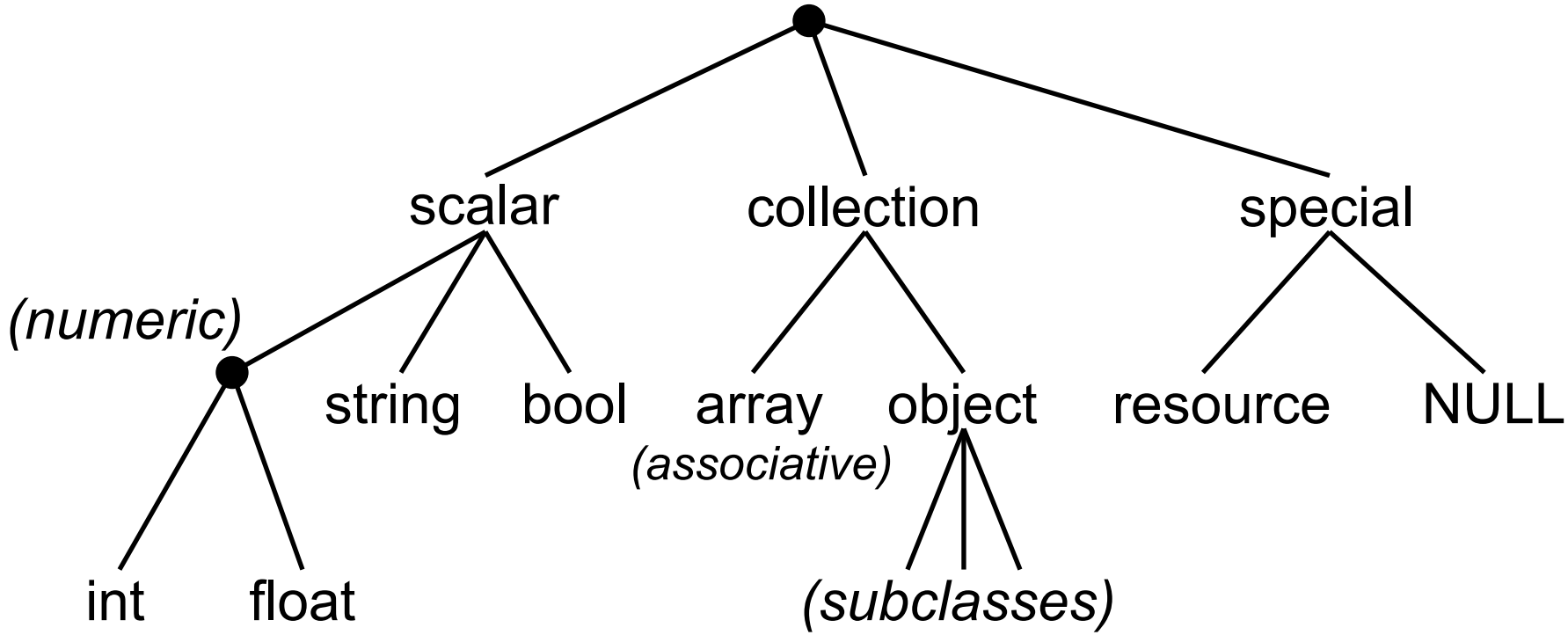
Input and Output

- Input: EGPCS superglobals
  - **`$_ENV`**, **`$_GET`**, **`$_POST`**, **`$_COOKIE`**, **`$_SERVER`**
  - **`$_REQUEST`**: union of G,P,C
  - **`$_FILES`**: contains uploaded files
  - **`$_SESSION`**: persistent state across loads
- Output: printed text in HTML document
  - **`echo`**, **`print()`**, **`printf()`**
  - **`var_dump()`**, **`print_r()`**: print human-readable form for debugging; warning: problems with cycles
  - **`phpinfo()`**: prints lots of diagnostic information

Lexical Peculiarities

- Embedded in HTML with **`<?php`** … **`?>`**
- Variables are case sensitive; classes, functions, and keywords are case insensitive
- All variables (including arrays) begin with dollar sign (**`$`**), can be interpolated in string
- Semicolon required even after last statement in block, optional only before **`?>`**
- Single-line comments, or up to **`?>`**: **`#`**, **`//`**
- Multi-line comments, even across **`?>`**: **`/*`**…**`*/`**
- Literals: **`"s"`**,**`'s'`**,multi-line string,**`true`**,**`null`**
- Heredocs: continues after closing tag

Types


scalar
collection
special
(numeric)
string
bool
array
object
resource
NULL
(associative)
int
float
(subclasses)

PHP

Type Conversions

Value		bool	number	string
bool	`false`	Identity	`0`	`""`
	`true`		`1`	"1"
number	`0`	`false`	Identity	Represent value as string
	`1`	`true`		
	Other	`true`		
string	`""`	`false`	`0`	Identity
	`"0"`	`false`	`0`	
	Other	`true`	Numeric prefix	
`null`		`false`	`0`	`""`
array	empty	`false`	Error	`"Array"`
	Other	`true`		
object	empty	`false`	Error	`"Object"`
	Other	`true`		

Variable Declarations

Implicit	`echo $a + 1;` `$b = 5;`	Read NULL if non-existent
Constant (global)	`define('PI', "3.14")` `echo PI`	Not a variable, don't need dollar sign (`$`)
Global, used locally	`global $g;` `$g = $g + 1;`	Otherwise, write creates new local `$g`
Local, unlimited lifetime	`static $s;` `$s++;`	Otherwise, value forgotten after return

PHP

Operators

new		N	Create object
[…**]**		L	Array subscript
++, **--**	1	N	Auto-increment / decrement
~, **@**, **(int)**, **(array)**, …	1	N	**~**: bitwise negation; **@**: inhibit errors; **(***type***)**: cast
instanceof	2	N	Class/interface membership
!	1	R	Logical negation
*****, **/**, **%**	2	L	Multiplicative
+, **-**, **.**	2	L	Additive; .: string concatenation
<<, **>>**	2	L	Bitwise shift
<, **<=**, **>**, **>=**	2	N	Comparison
==, **!=**, **<>**, **===**, **!==**	2	N	Identity; **===**, **!==**: equal value + type
&, **^**, **|**	2	L	Bitwise (not all same precedence)
&&, **||**	2	L	Logical (not all same precedence)
?:	3	L	Conditional
=, **+=**, **-=**, …	2	L	Assignment
and, **or**, **xor**	2	L	Logical (not all same precedence)
,	2	L	List separator

Administrative

Last Slide

- Homework 4 due Friday 10/4/2013

- Today's lecture
  - HTML, HTTP
  - Server-side scripting
- Next lecture
  - More PHP