INTRODUCTION

TO

PHP

Introduction to PHP

"PHP is a server-side scripting language designed specifically for the Web. Within an HTML page, you can embed PHP code that will be executed each time the page is visited. Your PHP code is interpreted at the Web server and generates HTML or other output that the visitor will see" ("PHP and MySQL Web Development", Luke Welling and Laura Thomson, SAMS)

URLs and web servers

```
http://server/path/file
```

- usually when you type a URL in your browser:
  - your computer looks up the server's IP address using DNS
  - your browser connects to that IP address and requests the given file
  - the web server software (e.g. Apache) grabs that file from the server's local file system, and sends back its contents to you

- some URLs actually specify *programs* that the web server should run, and then send their output back to you as the result:
  `https://csjmu.ac.in/quote.php`
  - the above URL tells the server `csjmu.ac.in` to run the program `quote.php` and send back its output

Server-Side web programming

- server-side pages are programs written using one of many web programming languages/frameworks
  - examples: PHP, Java/JSP, Ruby on Rails, ASP.NET, Python, Perl
- the web server contains software that allows it to run those programs and send back their output
- each language/framework has its pros and cons
  - we will use PHP for server-side programming

PHP History

- 1994: Created by Rasmis Lesdorf, software engineer (part of Apache Team)
- 1995: Called Personal Home Page Tool, then released as version 2 with name PHP/FI (Form Interpreter, to analyze SQL queries)
- Half 1997: used by 50,000 web sites
- October 1998: used by 100,000 websites
- End 1999: used by 1,000,000 websites

Good about PHP

- Open-source
- Easy to use (C-like and Perl-like syntax)
- Stable and fast
- Multiplatform
- Many databases support
- Many common built-in libraries
- Pre-installed in Linux distributions

Why PHP?

There are many other options for server-side languages: Ruby on Rails, JSP, ASP.NET, etc.

Why choose PHP?

- free and open source: anyone can run a PHP-enabled server free of charge
- **compatible:** supported by most popular web servers
- **simple:** lots of built-in functionality; familiar syntax
- **available:** installed on servers and most commercial web hosts
- **well-documented:** type `php.net/`*functionName* in browser Address bar to get docs for any function

What is PHP?

- **PHP** stands for "PHP Hypertext Preprocessor"
- a server-side scripting language
- used to make web pages dynamic:
  - provide different content depending on context
  - interface with other services: database, e-mail, etc
  - authenticate users
  - process form information
- PHP code can be embedded in HTML code

What is PHP (cont’d)

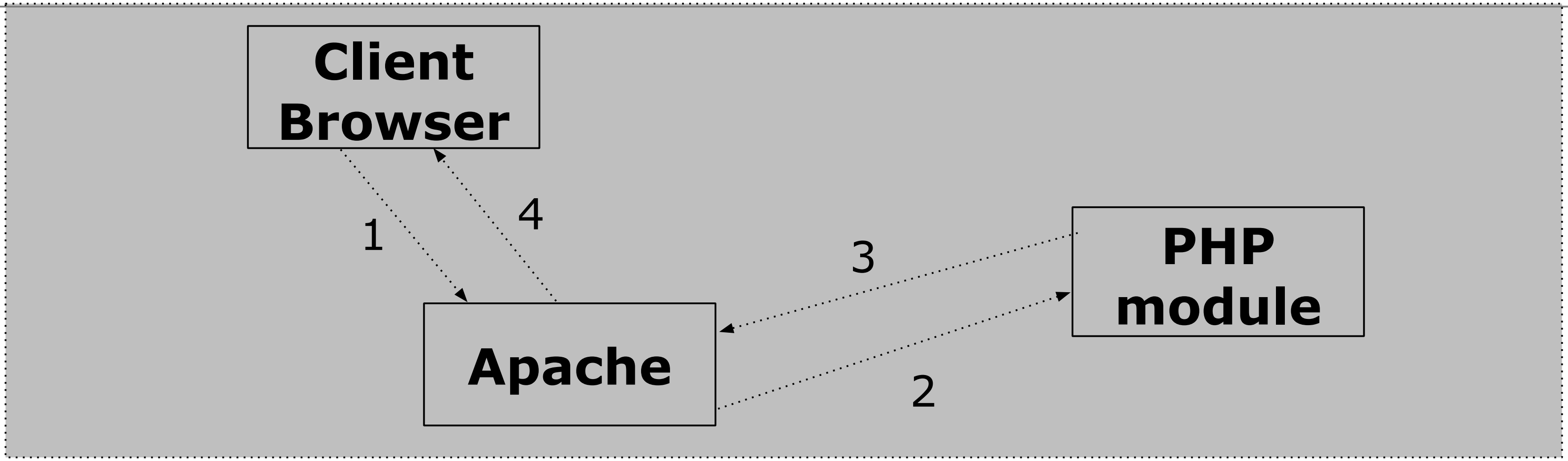
- Interpreted language, scripts are parsed at run-time rather than compiled beforehand
- Executed on the server-side
- Source-code not visible by client
    - ‘View Source’ in browsers does not display the PHP code
- Various built-in functions allow for fast development
- Compatible with many popular databases

What does PHP code look like?

- Structurally similar to C/C++
- Supports procedural and object-oriented paradigm (to some degree)
- All PHP statements end with a semi-colon
- Each PHP script must be enclosed in the reserved PHP tag

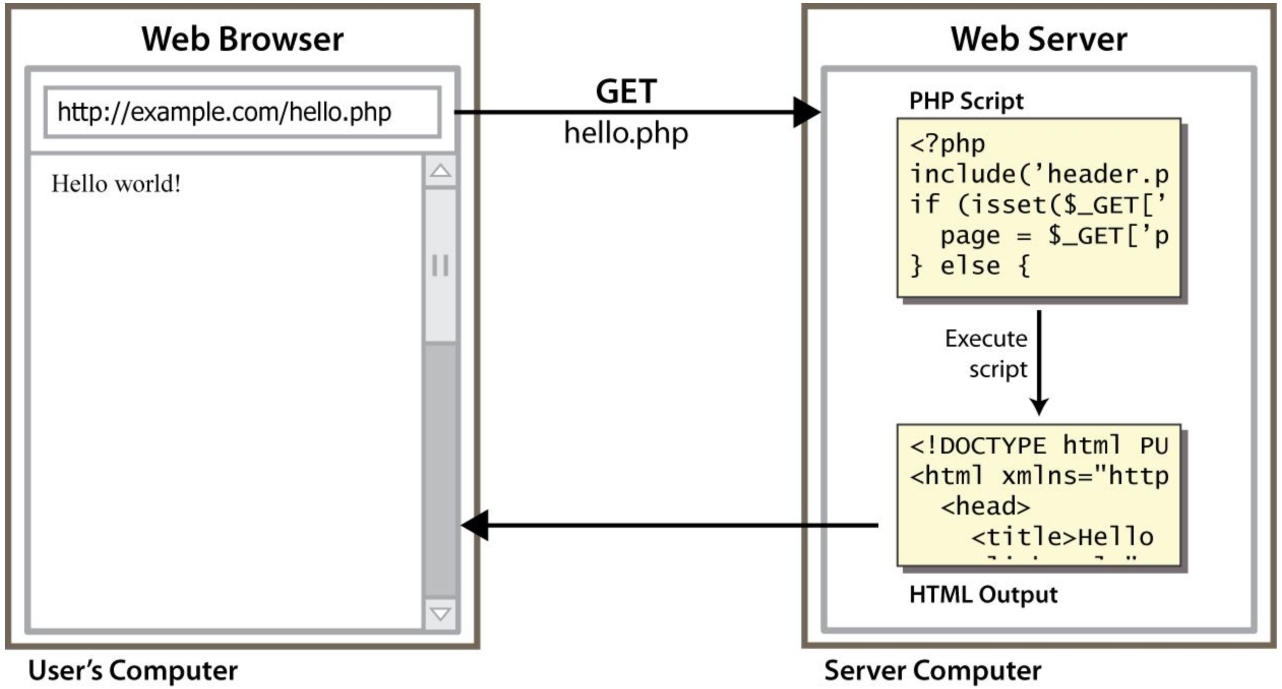
```
<?php
     …
?>
```

How PHP generates HTML/JS Web pages



1: Client from browser send HTTP request (with POST/GET variables)
2: Apache recognizes that a PHP script is requested and sends the request to PHP module
3: PHP interpreter executes PHP script, collects script output and sends it back
4: Apache replies to client using the PHP script output as HTML output

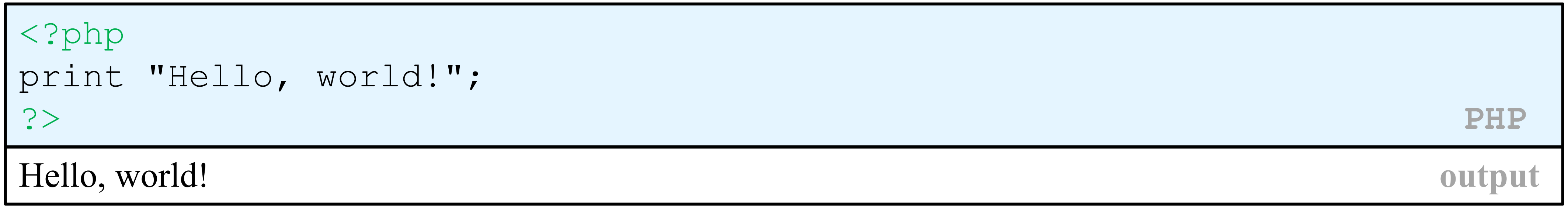
Lifecycle of a PHP web request



- browser requests a `.html` file (**static content**): server just sends that file
- browser requests a `.php` file (**dynamic content**): server reads it, runs any script code inside it, then

Hello, World!

The following contents could go into a file hello.php:

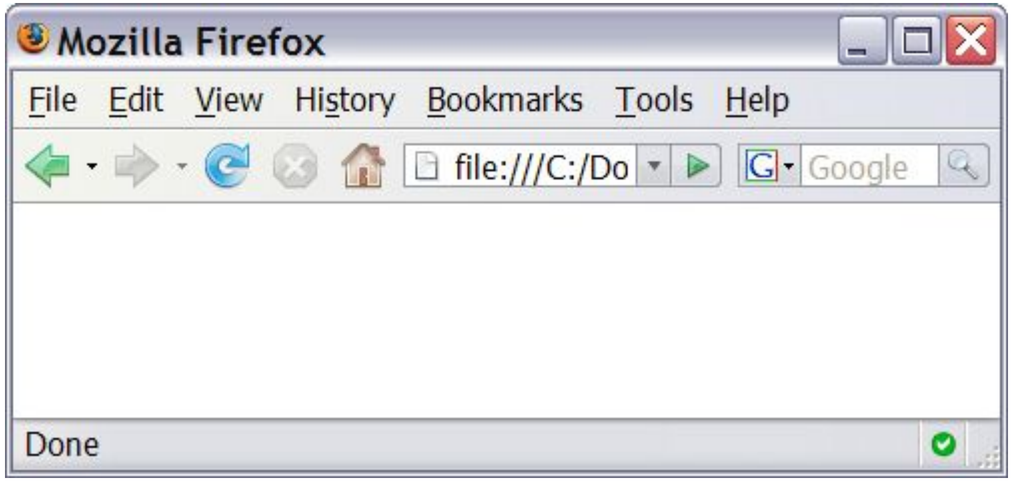


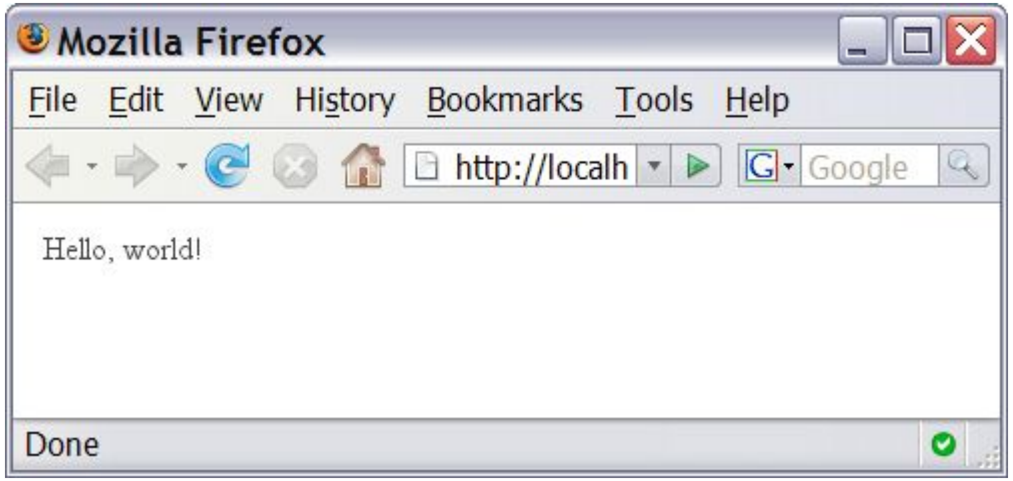
```php
<?php
print "Hello, world!";
?>
```

```
Hello, world!
```

- a block or file of PHP code begins with `<?php` and ends with `?>`
- PHP statements, function declarations, etc. appear between these endpoints

Viewing PHP output





```
Hello, world!
```

- you can't view your `.php` page on your local hard drive; you'll either see nothing or see the PHP source code
- if you upload the file to a PHP-enabled web server, requesting the `.php` file will run the program and send you back its output

Console output: print

```
print "text";                                                              PHP
```

```
print "Hello, World!\n";
print "Escape \"chars\" are the SAME as in Java!\n";

print "You can have
line breaks in a string.";

print 'A string can use "single-quotes".  It\'s cool!';                    PHP
```

```
Hello, World! Escape "chars" are the SAME as in Java! You can have line breaks in a
string. A string can use "single-quotes". It's cool!                    output
```

- some PHP programmers use the equivalent `echo` instead of `print`