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://webster.cs.washington.edu/cse190m/quote.php`
  - the above URL tells the server `webster.cs.washington.edu` to run the program `quote2.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

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

Requirements & Installation

To start using PHP, you need:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL (Wamp or Xampp)
- Any Text Editor (Notepad++) , Web Editor (Adobe Dreamweaver)
- Web Browser (Edge , Chrome , Firefox, Opera …)

Saving Your PHP Files

If You have XAMPP :

Place your PHP **files** in the "HTDocs" folder located under the "XAMPP" folder on your C: drive.
The **file** path is "C:**xampp**\htdocs" for your Web server.
Make sure your PHP **files** are **saved** as such; they must have the ". php" **file** extension.

If You have WAMP :

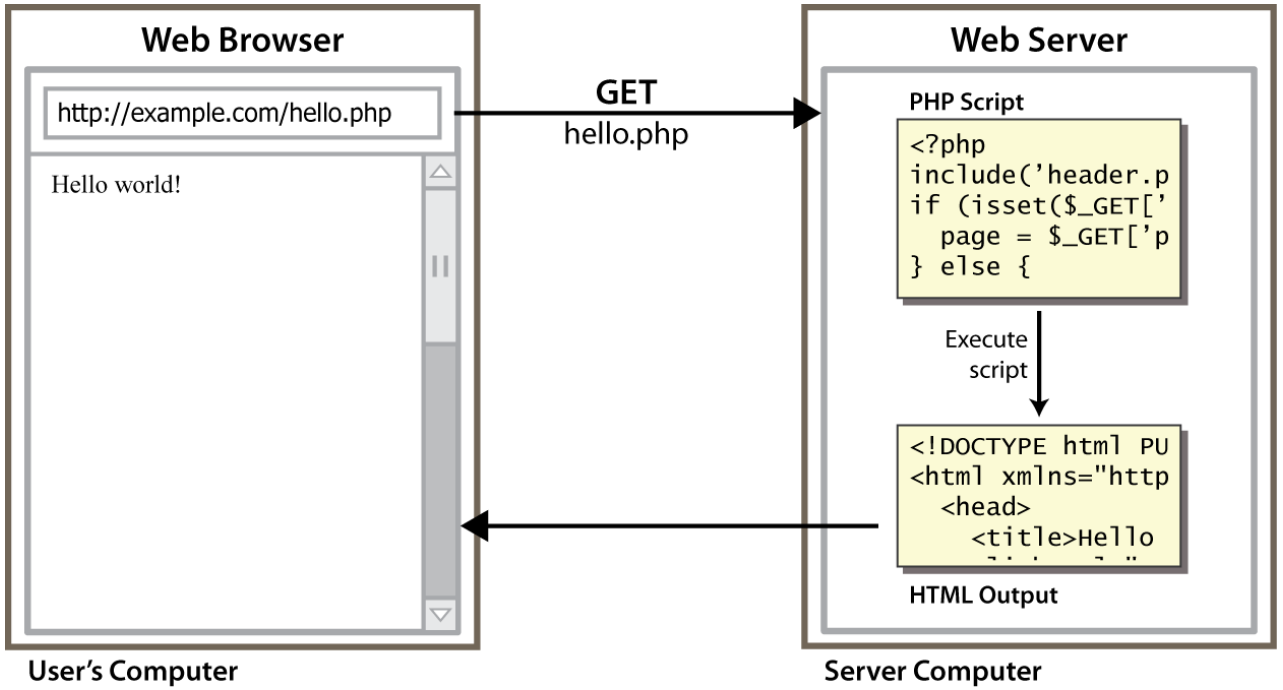
Assume you installed **WAMP** in C Drive. Go to: C:**wamp**\www

Make sure your PHP **files** are **saved** as such; they must have the ". php" **file** extension.

View Your PHP File in Web Browser

1. - Make sure that you saved the file in the root directly of your server (Xampp or Wamp)
2. - Make sure that your server is running
3. - Open any Web Browser
4. - In the address bar type the following : **http://localhost/your_file_Name.php**

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!";
?>
```

PHP

```
Hello, world!
```

output

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

Hello, World!

Example

```php
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

PHP is Fun!

Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.

Variables

```php
<?php
$txt1 = "Learn PHP";
$txt2 = “IUL.edu.lb";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

Output:

Learn PHP

Study PHP at IUL.edu.lb
9

Data Types

Variables can store data of different types, and different data types can do different things.

PHP supports the following data types:

String

Integer

Float (floating point numbers - also called double)

Boolean

Array

Object

NULL

Variables

```
$name = expression;
```
PHP

```
$user_name = "PinkHeartLuvr78";
$age = 16;
$driving_age = $age + 2;
$this_class_rocks = TRUE;
```
PHP

- names are case sensitive; separate multiple words with _
- names always begin with $, on both declaration and usage
- implicitly declared by assignment (type is not written; a "loosely typed" language)

for loop

```
for (initialization; condition; update) {
    statements;
}
```
PHP

```
for ($i = 0; $i < 10; $i++) {
  print "$i squared is " . $i * $i . ".\n";
}
```
PHP

if/else statement

```php
if (condition) {
  statements;
} else if (condition) {
  statements;
} else {
  statements;
}
```

- can also say `elseif` instead of `else if`

while loop (same as Java)

```
while (condition) {
  statements;
}
```
PHP

```
do {
  statements;
} while (condition);
```
PHP

- break and continue keywords also behave as in Java

Comments

```php
# single-line comment

// single-line comment

/*
multi-line comment
*/
```

PHP

- like Java, but # is also allowed
  - a lot of PHP code uses # comments instead of //
  - we recommend # and will use it in our examples