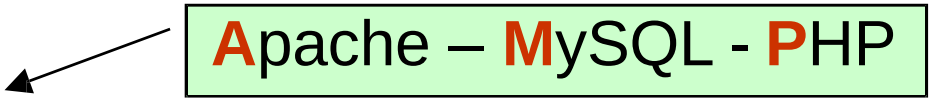
XAMPP Installation

Introducing XAMPP

- An *integration* package containing a number of useful packages that make it easy to host web sites on various platforms.
  - WAMP or LAMP



- Allow the ease of installation and set up
- Main Page: **http://www.apachefriends.org/en/xampp.html**

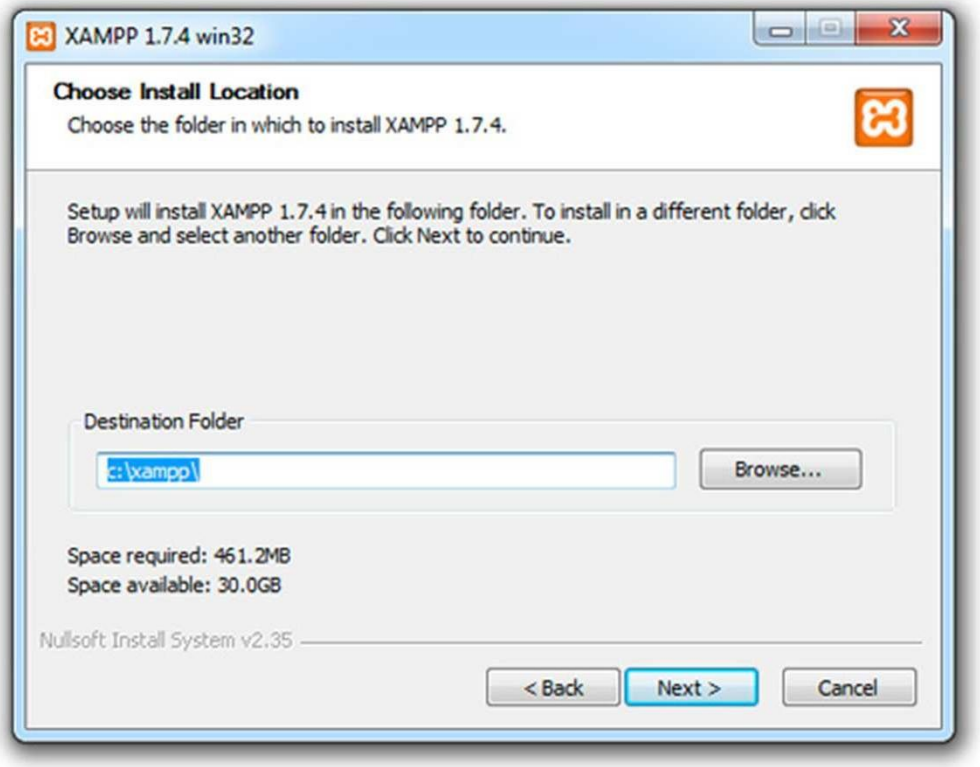
Introducing XAMPP (cont.)

Basic packages include system, programming & server software:

- Apache: the famous Web server
- MySQL: the widely-used, free, open source database
- PHP: the programming language
- Perl: the programming language
- ProFTPD: an FTP server
- OpenSSL: for secure sockets layer support
- PhpMyAdmin: for MySQL admin.

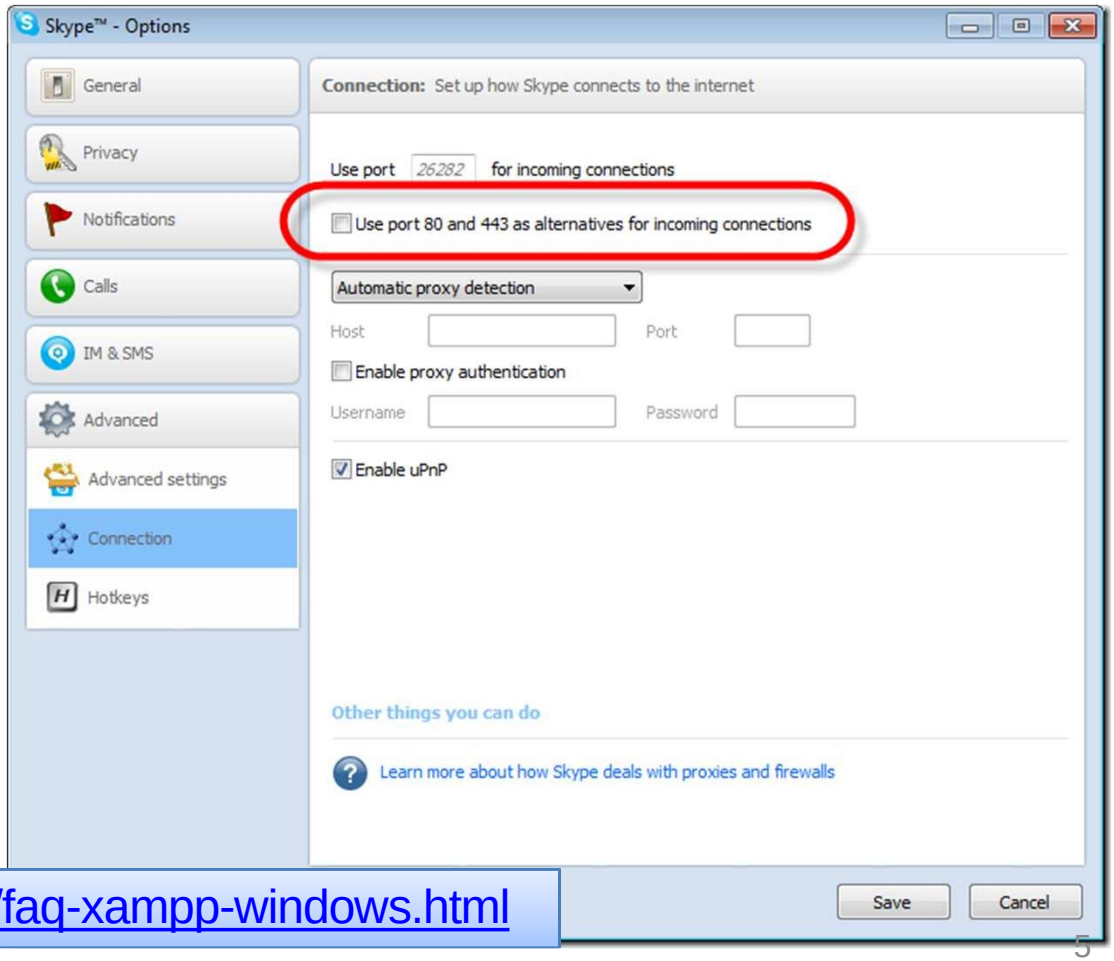
XAMPP Installation

- Download XAMPP installer and let the install begin:
  - Using the installer version is the easiest way to install XAMPP.
  - Use default directory for convenience



There can be some problems

Port 80 (Apache's default port) can be occupied by other programs €

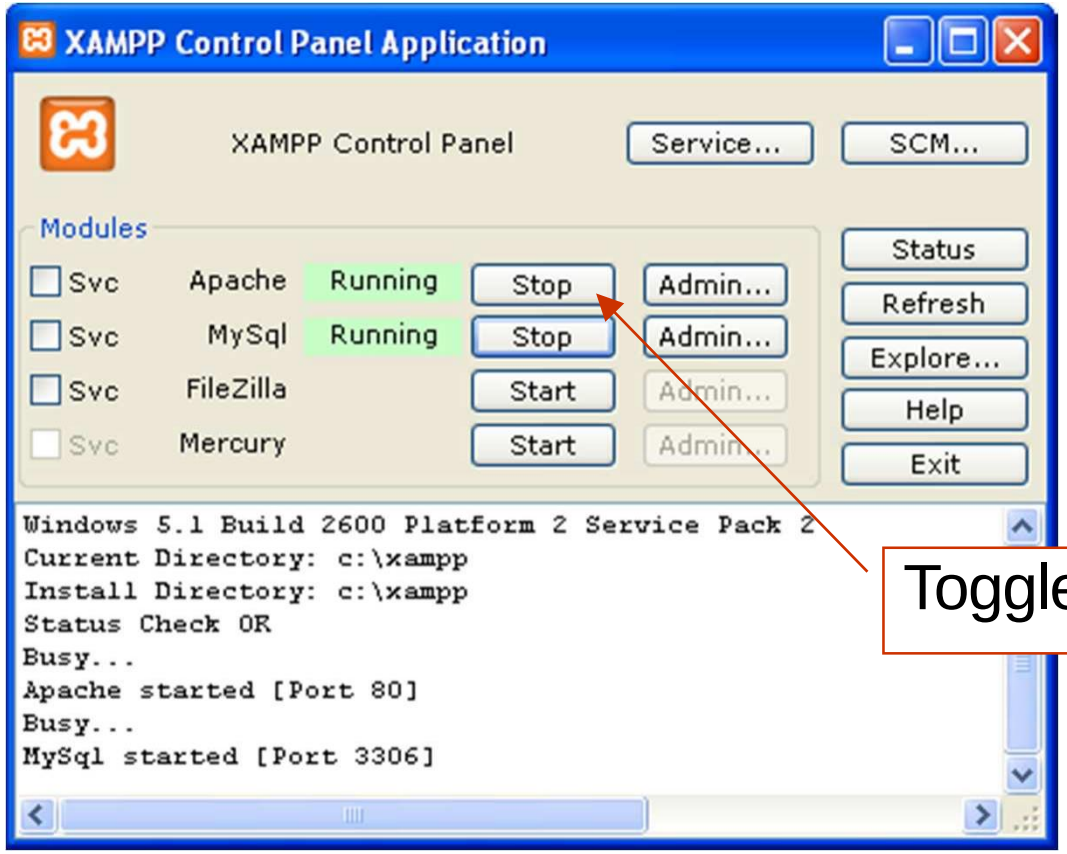


http://www.apachefriends.org/en/faq-xampp-windows.html

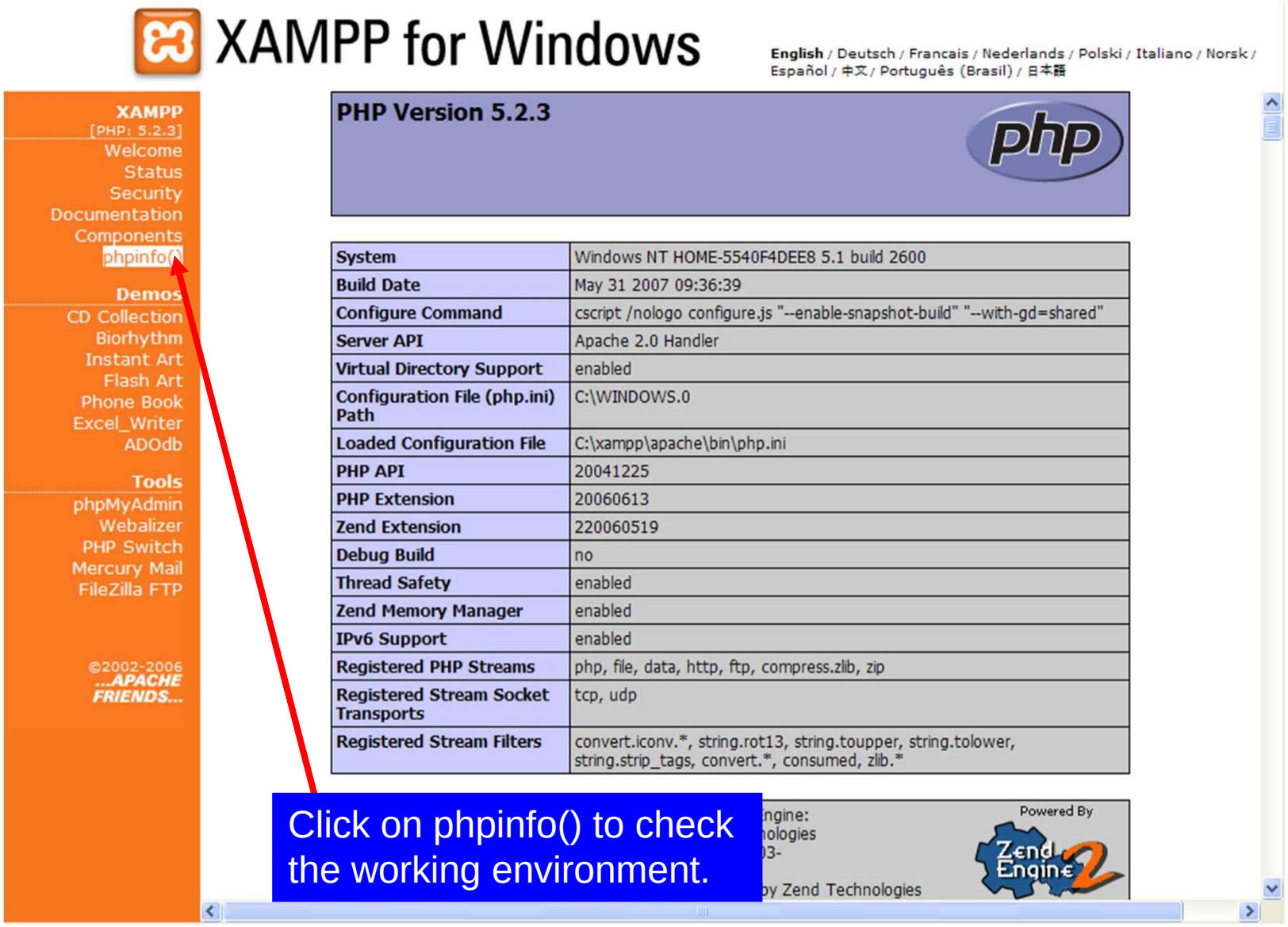
XAMPP Directories

- XAMPP default installation directory is c:/xampp/
- The directory of interest is "c:/xampp/htdocs/" and it's called the webroot (or document root)
  - PHP files are put in the webroot (c:/xampp/htdocs/)
  - c:/xampp/htdocs/ maps to http://localhost/
    - For example, c:/xampp/htdocs/project/script.php maps to € http://localhost/project/script.php
  - If no file is specified, Apache looks for index.php
    - For example, c:/xampp/htdocs/project/ maps to € http://localhost/project/index.php

Starting Apache & MySQL



Check the “working environment”



PHP Fundamentals

- PHP “Hello World” Example
- PHP Variables
- Variable Types
- Working with User Input
- Variable Operators
- Conditional Statements

What is PHP?

- PHP: Hypertext Preprocessor
  - Server-side scripting language
  - Creation of dynamic content
  - Interaction with databases
  - Can be embedded in HTML
  - Open source, written in C
  - First introduced in 1995; at that time PHP stood for Personal Home Page.
  - Similar to Perl and C

Exercise #1: Hello PHP

- The PHP code is usually in files with extension ".php“
- The PHP code can be embedded within HTML code

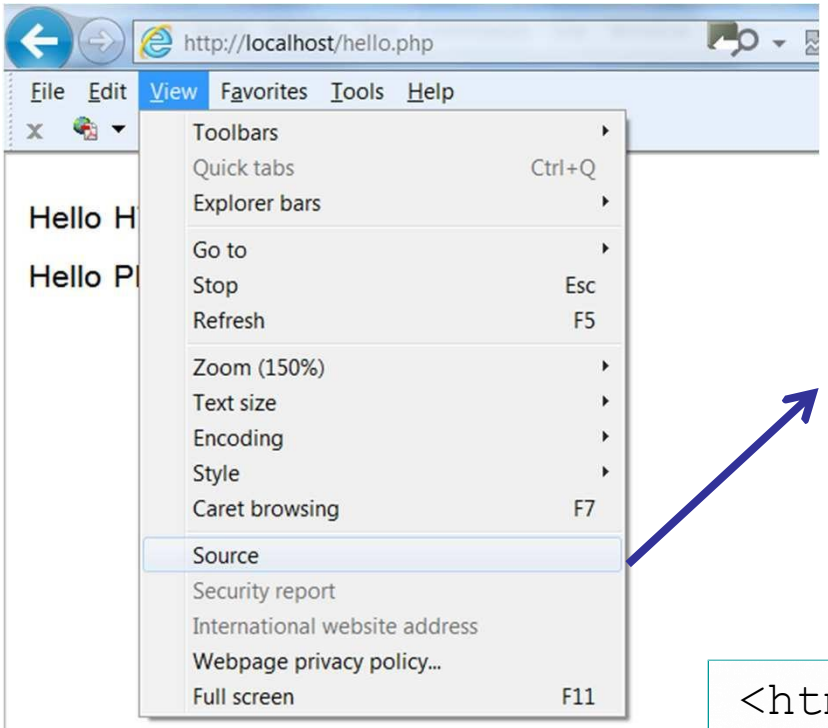
<?php denotes start of PHP code

?> denotes end of PHP code

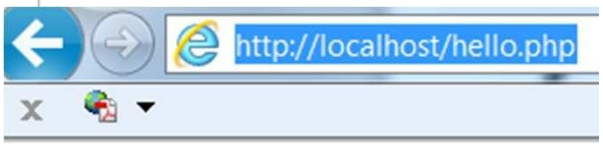
PHP statements must be ended with a semicolon.

```
<html>
<head><title>Hello world page</title></head>
<body>
Here is my HTML!<?php echo "Hello PHP!"; ?>
</body>
</html>
```

Client-side vs. Server-side Script



PHP code is never sent to a client's Web browser; only the HTML output of the processing is sent to the browser.



Hello HTML!

Hello PHP!

```
<html>
<head><title>Hello world
page</title></head>
<body>
Hello HTML!<br>
<?php echo "Hello PHP!"; ?>
</body>
</html>
```

Client-side vs. Server-side Script

- HTML code is processed by browsers as web pages are loading. (client-side)
- PHP code is preprocessed by PHP Web servers that parse requested web pages as they are being passed out to the browser. (Server-side)
- You can embed sections of PHP inside HTML:

```
<BODY>
<p>
<?php $test ="Hello World!";
echo $test; ?>
</p>
</BODY>
```

- Or, you can call HTML from PHP :

```
<?php
echo "<html><head><title>Hello</title>
…
?>
```

PHP Comments

- You can add comments to the code
  - Starting with "//", "#" or block in "/*" and "*/"
  - Only "/*" – "*/" can be used over several lines
  - Comments are NOT executed

PHP Fundamentals

- PHP “Hello World” Example
- PHP Variables
- Variable Types
- Working with User Input
- Variable Operators
- Conditional Statements

PHP Variables

- What are variables?
  - The values stored in computer memory are called **variables.**
  - The values, or data, contained in variables are classified into categories known as **data types.**
  - The name you assign to a variable is called an **identifier.**

PHP Variables

- Prefixed with a $ (Perl style)
- Assign values with = operator
- Example: $name = “John Doe”;
- No need to define type
- Variable names are case sensitive
  - $name and $Name are different

PHP Variables

```
<?php   // declare string variable $output
$output = "<b>Hello PHP!</b>";
Echo $output; €  Hello PHP!
?>
```

- Each variable is declared when it's first assigned value.
- The type of the value determines the type of the variable.

PHP Fundamentals

- PHP “Hello World” Example
- PHP Variables
- Variable Types
- Working with User Input
- Variable Operators
- Conditional Statements

Variable/Data Types

- A **data type** is the specific category of information that a variable contains

- Possible PHP Variable Types are:
  - Numeric (real or integer)
  - Boolean (TRUE or FALSE)
  - String (set of characters)

Variable/Data Types

- Unlike C, PHP is **not** strictly typed.
- PHP decides what type a variable is based on its value.
- PHP can use variables in an appropriate way automatically
- For Example:
  - $HST = 0.12; // HST Rate is numeric
  - echo $HST * 100 . “%”; //outputs “12%”
    - $HST is automatically converted to a string for the purpose of echo statement

Displaying Variables

- To display a variable with the `echo` statement, pass the variable name to the `echo` statement without enclosing it in quotation marks :
  - `$VotingAge = 18;`
  - `echo $VotingAge;`
- To display both text strings and variables, you may used concatenation operator “.” :

```
echo "<p>The legal voting age is “.
$VotingAge. ".</p>";
```

Concat. operator

Period

Naming Variables

- The name you assign to a variable is called an **identifier**

- The following rules and conventions must be followed when naming a variable:
  - Identifiers must begin with a dollar sign ($)
  - Identifiers may contain uppercase and lowercase letters, numbers, or underscores (_).
  - The first character after the dollar sign must be a letter.
  - Identifiers cannot contain spaces or special characters.
  - Identifiers are case sensitive

Declaring and Initializing Variables

- Specifying and creating a variable name is called **declaring the variable**
- Assigning a first value to a variable is called **initializing the variable**
- In PHP, you must declare and initialize a variable in the same statement:
  - `$variable_name = value;`

PHP Strings

- String values
  - Strings may be in single or double quotes

```
<?
$output1 = "Hello PHP!";
$output2 = 'Hello again!';
?>
```

  - Start and end quote type should match
  - Difference between two types of quotes is the escape sequences

Strings escaping

- Special chars in strings are escaped with backslashes (C style)
- Double-quoted string

```
$str1 = "this is \"PHP\""; echo $str1;
€  outputs this is “PHP”
```

- Single-quoted string

```
$str2 = ' "I\'ll be back ", he said. '; echo $str2;
€  outputs "I'll be back", he said.
```

Variables in strings

- Double quoted strings offer something more:

```
$saying = "I'll be back!";
$str1 = “He told me: $saying";
outputs €  He told me: I'll be back!
```

- Variables are evaluated in double-quoted strings, but not in single-quoted strings.

```
$saying = "I'll be back!";
$str2 = ‘He told me: $saying’;
outputs €  He told me: $saying
```

- For single-quoted strings, use concatenation:

```
$saying = "I'll be back!";
$str3 = ‘He told me: ’ . $saying;
outputs €  He told me: I'll be back!
```