



# CSC-318

# Web Technology

(BSc CSIT, TU)

Ganesh Khatri  
kh6ganesh@gmail.com

# Chapter 6 : PHP Introduction

- Web Server Introduction
- Server Side Scripting with PHP
- PHP Variables
- PHP Types
- PHP Comments
- PHP Inbuilt Functions
- PHP Conditional Statements
- PHP Loops

# Web Server

- is a computer or computer software that listens and responds to a client computer's request made through a web browser
- in other words, a web server is server software, or hardware dedicated to running said software, that can satisfy WWW client requests
- machine that hosts web pages and other web documents
- provides web documents and other online services using HTTP
- web server can, in general, contain one or more websites



## Web Server

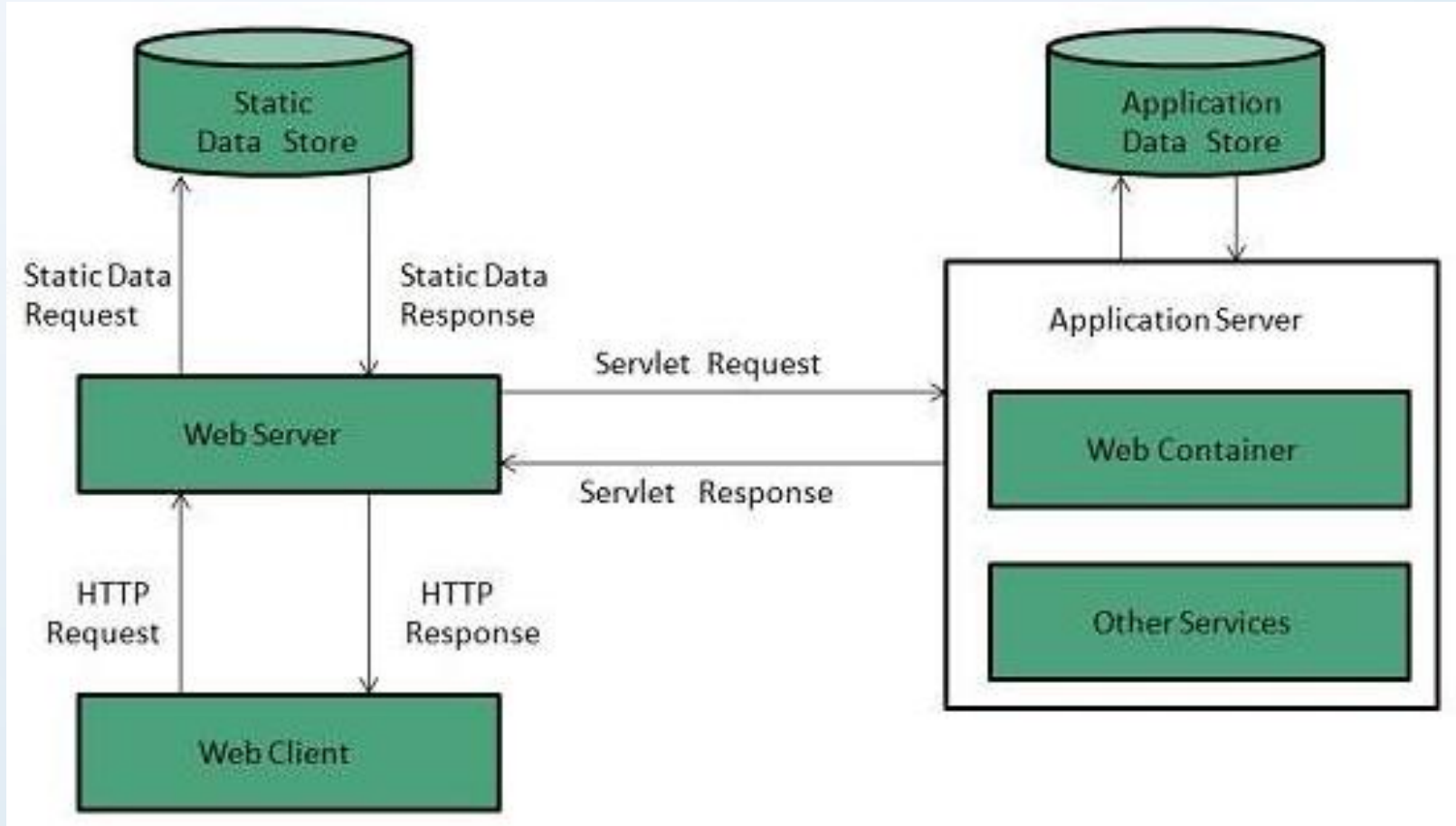
# Web Server

- processes incoming network requests over HTTP and several other related protocols
- primary function of a web server is to store, process and deliver web pages to web clients
- communication between client and server takes place using the Hypertext Transfer Protocol (HTTP)
- pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to the text content



## Web Server

# Web Server Working



# Web Server Working

- Web server respond to the client request in either of the following two ways
  - Sending the file to the client associated with the requested URL
  - Generating response by invoking a script and communicating with database
- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response
- If the requested web page is not found, web server will the send an HTTP response:Error 404 Not found
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response



# Examples of Web Servers

- Apache Web Server
  - most popular web server in the world developed by Apache Software Foundation
  - open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, Mac OS X and more
  - About 60% of the web server machines run the Apache Web Server
- Internet Information Servicer (IIS)
  - is a high performance Web Server from Microsoft
  - runs on Windows platform
  - is tightly integrated with the operating system so it is relatively easy to administer it.
- Lighttpd
  - this is fast, secure and consumes much less CPU power
  - can run on Windows, Mac OS X, Linux and Solaris operating systems
- Sun Java System Web Server
- Jigsaw Server

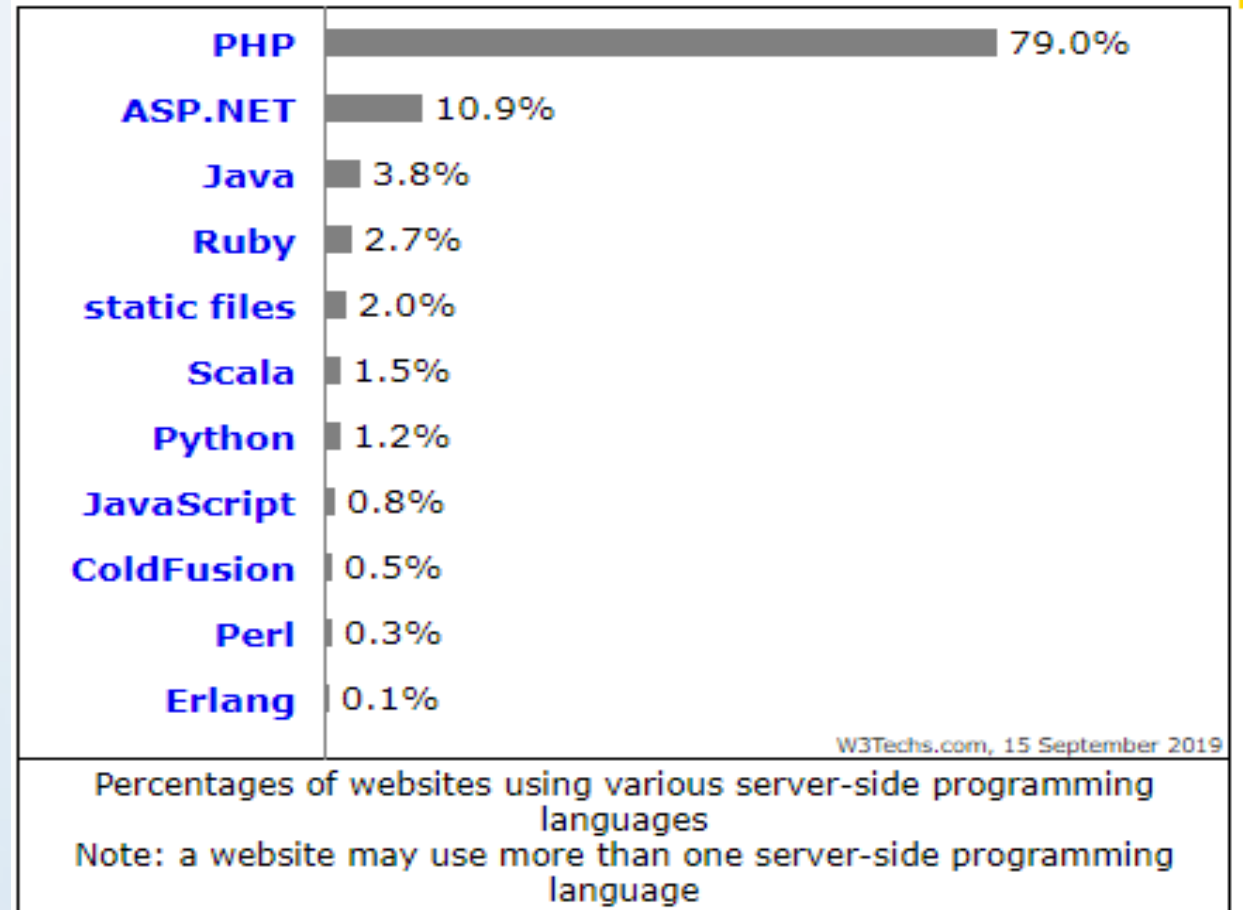
# Server Side Scripting with PHP

- Stands for Hypertext Pre-Processor
- Script that run on server is called server side script
- There are several languages used for Server-Side Programming including
  - PHP
  - Python
  - Ruby
  - Java
  - C#
  - C
  - Perl
  - Javascript\*



# PHP Programming

- This module will be using PHP because
  - It is widely used in the industry
- (Source : <https://w3techs.com>) ( Sep, 2019 )



# PHP Programming

- It's easy to find help
  - If you have a problem, someone else would have had it before and found a solution. Googling your problem will usually give you the solution!
- The documentation is very good
- It's free and open source + works on all operating systems.
- It's easy to learn compared with others

# PHP Programming

- PHP is most often used to generate the HTML that is finally sent to the browser
- This can be things like:
  - Including information from a database
  - Performing calculations and inserting them into the HTML
  - Getting data from elsewhere (e.g. a database) and formatting the result as HTML

# Running PHP Scripts

- PHP Scripts are generally run on a server and accessed via a web browser
- You can install PHP on your desktop/laptop machine and run PHP scripts via the command line
- This requires a web server

# Web Server

- Your browser (e.g. Chrome, Firefox) will connect to the web server
- The server sends the HTML code for the requested file
- The browser displays the HTML it receives
- The HTML does not have to be a simple .html file, it can be generated by a program on the server

# PHP Scripts

- You cannot run PHP scripts in a web browser!
- Web browsers do not understand PHP code
- You must connect the browser to a web server
- The server then runs the PHP code and sends the resulting HTML to the browser



# Your Own Web Server

- It's possible to set up a web server on your desktop/laptop
- This involves installing and configuring PHP
- As well as installing a piece of software to serve the pages, listen to connections on HTTP and return the requested files
- The server we will use is Apache although others are available

# Your Own Web Server

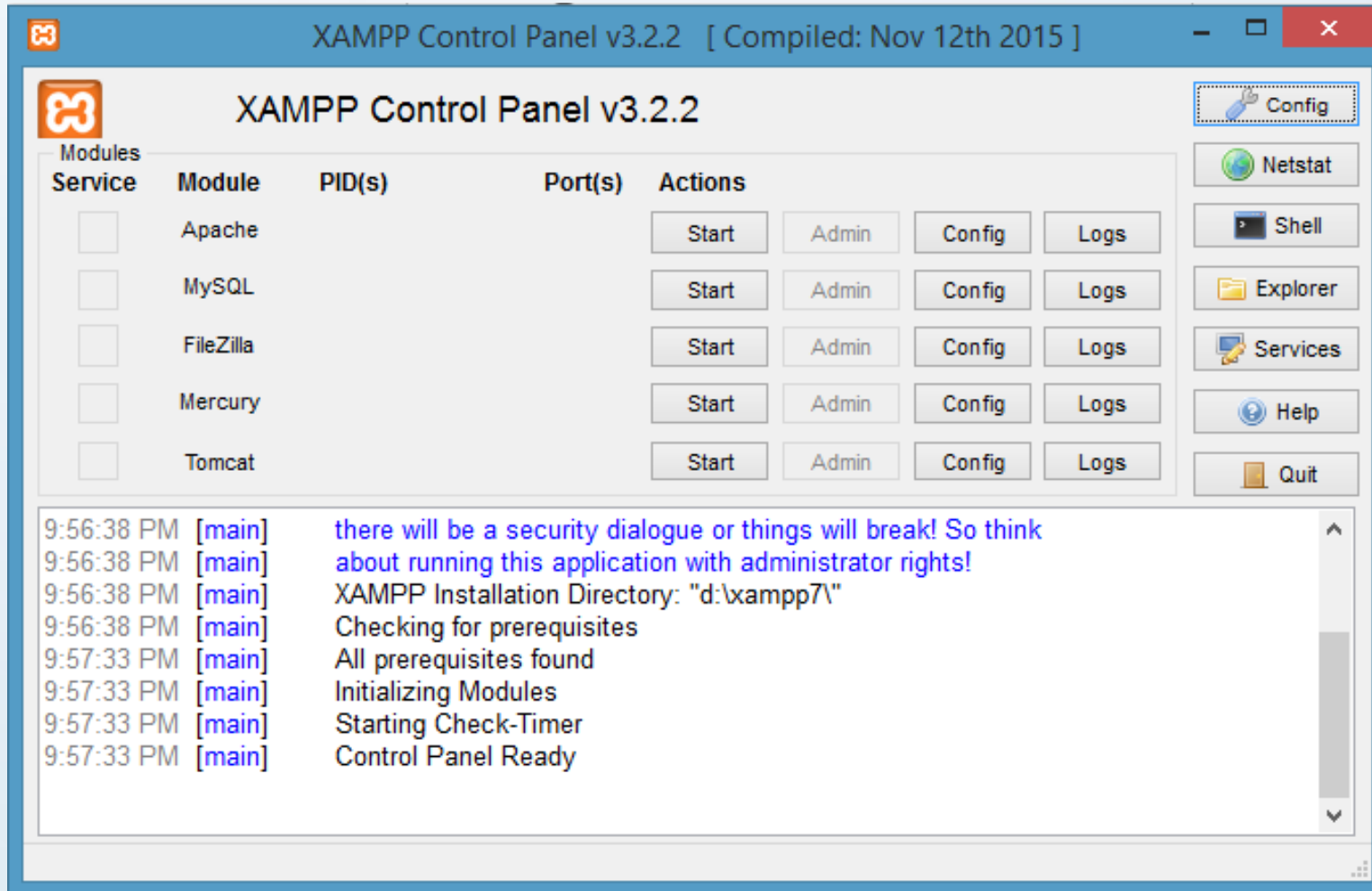
- There are several ways of getting a server installed on your machine
- Install PHP, Apache and MySQL manually and configure them yourself
  - Unless you know what you're doing, this can be difficult
  - There are a lot of different configuration options for both PHP and Apache, knowing how to set them up is art in itself

# Getting Started

- Download latest version of XAMPP from <https://www.apachefriends.org>
- Install it on any drive (like C, D, E etc)
- Open xampp-control.exe file located inside installed “xampp” folder.
- Start Apache and MYSQL servers
- Create a project folder(any name : project) inside htdocs folder
- project is the website root directory and create a file “index.php”
- Now run the project in browser by typing “localhost/project”

# Getting Started

➤ XAMPP Control looks like this :



# PHP Programming

- Once you have your server running you can write your first PHP script
- PHP code must be saved in files with a .php extension
- If you put PHP code in .html files it will not work
- However, you can include HTML in PHP files

# PHP Programming

- All PHP code is written between PHP tags
- PHP has start and end tags `<?php` and `?>`

```
<?php  
// your code here  
?>
```



# PHP Print

- To print to the screen use the echo command followed by your text in single quotes
- All statements must end with a semicolon ( ; )

```
<?php  
echo 'hello world';  
?>
```

# Single Vs Double Quotes

- PHP supports the use of single and double quotes
- They are mostly interchangeable
- However, by convention single quotes (apostrophes) are used in PHP
- Single quotes make it easier to work with HTML

```
<?php
echo '<div class="content">My content</div>';
echo "<div class=\"content\">My Content</div>";
?>
```

# Frirst PHP File

- By saving a file with a .php extension e.g. test.php inside the project directory, it will be accessible on <http://localhost/project/test.php>
- When you connect to the page, the server processes the code and anything output using echo is displayed in the browser
- You can use echo to print HTML
- Anything inside PHP tags (`<?php ... ?>` ) is processed on the server before being sent to the browser
- If you view the source of the page in the browser, you will not see PHP code, only the HTML that has been printed

# PHP Variables

- Variables must start with a dollar symbol ( \$ )
- A variable is an identifier for a piece of information
- You can save something under a label that you choose and retrieve it later in the program

```
$num = 123;  
  
echo $num;
```

# Comments

- Comments can be included in your code to explain what it is doing
- Comments are not processed by the computer
- The contents of comments don't have a strict structure and can contain any text
- Comments do not have an impact on the program. It will run exactly the same whether they are included or not

# Comments

```
// this is a single line comment  
  
/*  
    This is a multi-line comment  
*/
```

```
// echo 'test';
```

This code will not  
be run even though  
it is valid



# Types

- PHP is a loosely typed language
- This means you do not have to declare variables with a type
- In Java, for instance, you have to declare a variable before you can use it: You must declare what type that variable will store, and that variable can then only store a value of that type

```
String myVariable;  
myVariable = "hello";
```

Java Code, not PHP

# Types

- In PHP, you can create a variable without giving it a type
- That variable can later store other types:

```
//Create a variable to store an integer
$myIntVariable = 123;

//Create a variable to store a string,
$myStringVariable = 'A string';

//However, the type is not fixed: You can store any type in any variable:
$myIntVariable = 'A string';

$myStringVariable = 456;
```

# PHP Inbuilt Functions

- The rand() function can be used to generate a random number:
- Each time you run the script it will generate a random number

```
<?php  
echo rand();  
?>
```

```
Output:  
185902316
```

# PHP Inbuilt Functions

- The pi() function calculates mathematical PI to 14 decimal places

```
<?php  
echo pi();  
?>
```

```
Output:  
3.1415926535898
```

# PHP Inbuilt Functions

- Functions can take arguments
- The rand() function will generate a random number from 0 – 2147483647

```
<?php  
echo rand();  
?>
```

```
Output:  
727321813
```

- You can give functions arguments these are values which change the way the function works

# Arguments

- You can supply a min/max value to the rand function by providing two numbers inside the brackets:

```
<?php  
echo rand(1, 10);  
?>
```

Output:  
6

- Each argument is separated by a comma
- Different functions take different numbers of arguments
- Those arguments are specific to that function



# if Statement

- You can use an if statement to inspect the value of a variable and run some code when that condition is met
- The == operator inspects two values to see if they are equal

```
$num = 123;  
  
if ($num == 124) {  
    echo 'num is equal to 124';  
}  
  
if ($num == 123) {  
    echo 'num is equal to 123';  
}
```

Output:  
num is equal to 123

# else Statement

- You can combine an if statement and an else statement to run one piece of code if the condition is met, and another if it is not:

```
$num = 123;  
  
if ($num == 124) {  
    echo 'num is equal to 124';  
}  
else {  
    echo 'num is not equal to 124';  
}
```

Output:  
num is not equal to 124

# Concatenation

- String concatenation is a fancy term for joining strings together
- In PHP you can join strings with the . Operator
- This can be useful to reduce the number of echo commands required

```
$num = rand(1,6);  
$num2 = rand(1,6);  
  
echo 'You rolled a '  
echo $num;  
echo ' and a '  
echo $num2;  
echo '. '  
if ($num == 6 && $num2 == 6) {  
    echo 'You win!';  
}  
else {  
    echo 'You lose';  
}  
  
echo '<a href="dice.php">Roll again</a>';
```



```
$num = rand(1,6);  
$num2 = rand(1,6);  
  
echo 'You rolled a ' . $num . ' and a ' . $num2 . '. '  
  
if ($num == 6 && $num2 == 6) {  
    echo 'You win!';  
}  
else {  
    echo 'You lose';  
}  
  
echo '<a href="dice.php">Roll again</a>';
```

# Comparison Operators in PHP

- There are several comparison operators in PHP
- `$a == $b` equality, `$a` and `$b` are equal
- `$a != $b` not equals, `$a` and `$b` are not equal
- `$a > $b` true if `$a` is greater than `$b`
- `$a < $b` true if `$a` is less than `$b`
- For a complete list see
  - <https://www.php.net/manual/en/language.operators.comparison.php>

# PHP Types

➤ The types in PHP are:

- Double
- Boolean
- String
- Integer
- Array
- Object
- Resource
- Null

# Loops

- There are 2 main types of loop in PHP
  - For
  - While
- For loops for a predefined number of iterations. The number of times that the loop will occur is known upfront
- While loops while a condition is being met

# Loops

- Loops can be used to run the same code a number of times.
- A for loop is used when you know the number of iterations (a posh word for the number of times the loop will run!)
- This code will print "Text" ten times
- Any code between the opening and closing brace will be run

# for Loop

- It's possible to make use of the loop counter inside the loop
- The variable declared in the first part of the for statement (in this example called `i` ) will store the number of the current iteration
- This variable can be used like any variable, calculations, printing it

```
for ($i = 0; $i < 10; $i++) {  
    echo $i;  
}
```

Output:  
0123456789



# while Loop

- While loops will keep looping while a predefined condition is true
- You must affect the condition inside the loop to halt it!

```
$loop = true;
$counter = 0;

while ($loop === true) {

    $counter++;

    echo '<p>' . $counter . '</p>';

    if ($counter === 5) {
        $loop = false;
    }

}
```

Output in browser:

```
1
2
3
4
5
```

# Exercise 1

1. Using a loop, print out a list of all the numbers from 1-10 inside a an unordered list using <ul> and <li> tags
2. Change the exercise to print out the textual representation "One", "Two", "Three", "Four", etc
3. Write a program that uses a loop to display all the odd numbers from 21 to 99
  - **Grade A: Can you do this without an if statement?**
4. Write a program that prints the nine times table up to 12 x 9 (9, 18, 27, etc)
  - **Grade A: Can you do this without an if statement and without using the multiplication operator?**
5. For this exercise you may (and probably should!) use the multiplication operator. Print the nine times table from 9 – 900 in the format
  - 1 x 9 = 9
  - 2 x 9 = 18
  - 3 x 9 = 27