

**MET's Institute of Engineering**

**Bhujbal Knowledge City, Adgaon, Nashik.**

**Department of Computer Engineering**

# **'SERVER SIDE SCRIPTING LANGUAGE'**

**Prepared By**

**Prof. Anand N. Gharu**

**(Assistant Professor)**

**Computer Dept.**

**CLASS : TE COMPUTER 2019**

**SUBJECT : WT (SEM-II)**

**UNIT : V**

**17 April 2023**

Note: The material to prepare this presentation has been taken from internet and are generated only for students reference and not for commercial use.

# SYLLABUS

---

**PHP:** Introduction to PHP, uses of PHP, general syntactic characteristics, Primitives, operations and expressions, output, control statements, arrays, functions, pattern matching, form handling, files, cookies, session tracking, using MySQL with PHP, WAP and WML.

**Introduction to ASP.NET:** Overview of the .NET Framework, Overview of C#, Introduction to ASP.NET, ASP.NET Controls, Web Services.

**Overview of Node JS.**

# Syllabus

## PHP :

- Introduction to PHP, uses of PHP,
- general syntactic characteristics,
- Primitives, operations and expressions, output, control statements,
- arrays,
- functions,
- pattern matching,
- form handling, files,
- cookies, session tracking,
- using MySQL with PHP

## ASP.Net

- Overview of the .NET Framework,
- Overview of C#,
- Introduction to ASP.NET,
- ASP.NET Controls,
- Web Services

## WAP & WML

## Overview of Node JS.

PHIP

# Outline

Introduction to PHP,

Features, sample

code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

String manipulation,

Arrays & Functions,

Form handling,

Cookies & Sessions,

using MySQL with PHP,

# Introduction to PHP

---

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

**PHP was created by Rasmus Lerdorf in 1994** but appeared in the market in 1995. PHP 7.4.0 is the latest version of PHP, which was released on 28 November. Some important points need to be noticed about PHP are as followed:

# Introduction to PHP

- PHP stands for **Hypertext Preprocessor**.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.



# Introduction to PHP

---

- PHP is an acronym for "**PHP: Hypertext Preprocessor**"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

## What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

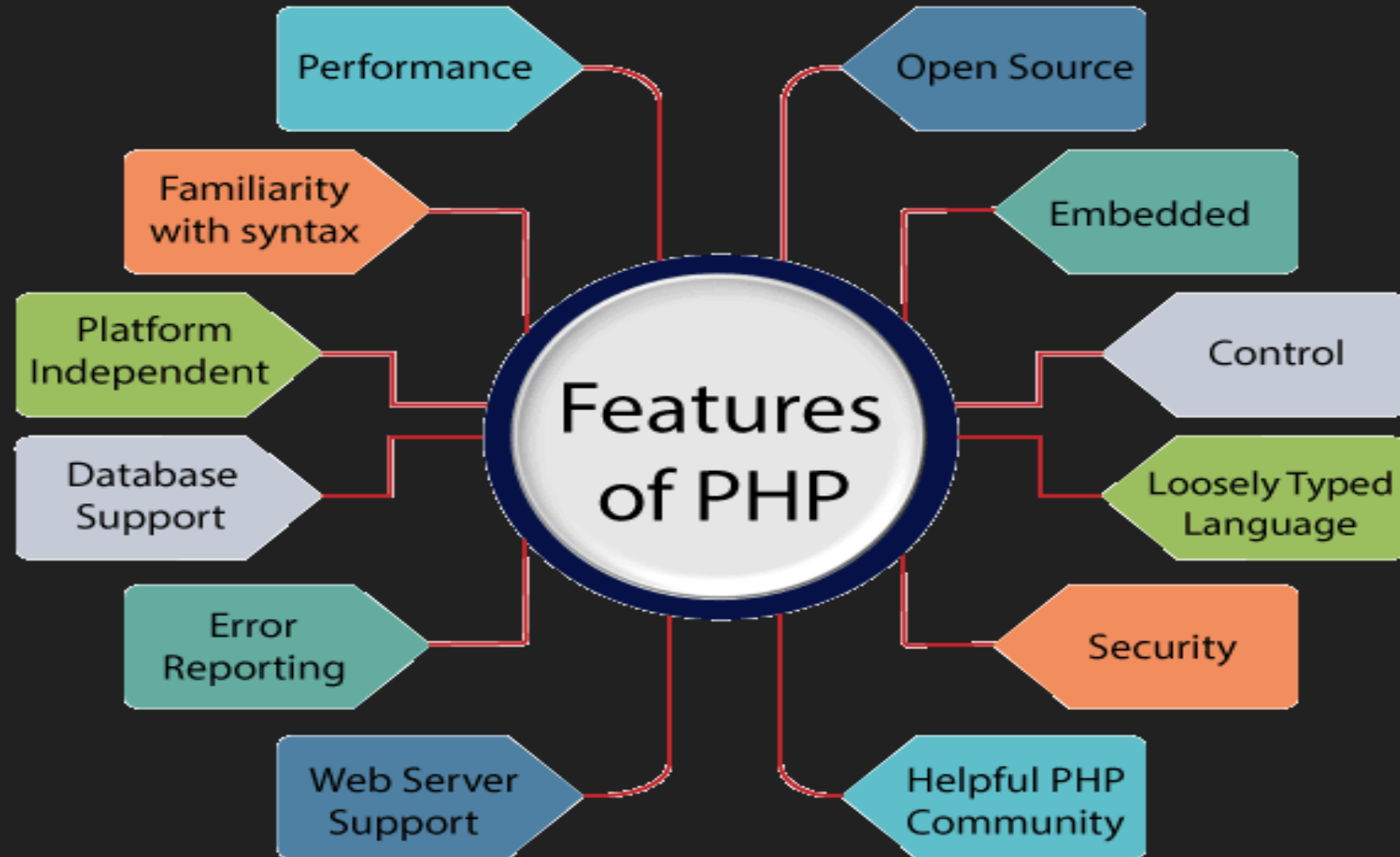


# Introduction to PHP (Uses of PHP)

---

- **What Can PHP Do?**
- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

# Features of PHP



# Features of PHP

---

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: [www.php.net](http://www.php.net)
- PHP is easy to learn and runs efficiently on the server side
- PHP provides security.
- PHP determines the error.
- **Performance faster** : execute faster than JSP and ASP.
- PHP code can easily be embedded in html tags, scripts.
- **Loosly Coupled** : PHP allows us to use a variable without declaring its datatype.

# PHP Syntax

---

## Basic PHP Syntax

A PHP script can be placed anywhere in the document. A PHP script starts with `<?php` and ends with `?>`:

```
<?php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php".

# sample code –

## Example 1 to print Hello World using PHP

---

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

# sample code –

## Example 2 variable declaration

---

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>
```

Out Put

```
"Hello world!"
5
10.5
```

# PHP Variables

---

- **Rules for PHP variables:**

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

# sample code –

## Example 3- To output text and a variable

```
<?php  
$txt = "W3Schools.com";  
echo "I love $txt";  
?>
```

```
<?php  
$txt = "W3Schools.com";  
echo "I love " . $txt .  
;  
?>
```

I love W3Schools.com!



# Program for addition of two numbers

---

```
<?php
$n1=5;
$n2=6;
$sum=$n1+$n2;
Echo "Summation is".$sum;
?>
```

Output = Summation is 11

# PHP Variables Scope

---

- In PHP, variables can be declared anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes:
  - local
  - global
  - static

# PHP Variables Scope- Global and Local Scope

## Example

```
<?php  
$x = 5; // global scope  
  
function myTest() {  
    echo "Variable x inside function  
    is: $x";  
}  
  
myTest();  
  
echo "Variable x outside function  
is: $x";  
?>
```

## Output

```
Variable x inside function is:  
Variable x outside function is: 5
```

A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function:

# PHP Variables Scope- Global and Local Scope

## Example

```
<?php
function myTest()
{
    $x = 5; // local scope
    echo "Variable x inside function
is: $x";
}
myTest();
echo "Variable x outside function
is: $x";
?>
```

## Output

```
Variable x inside function is: 5
Variable x outside function is:
```

A variable declared **within** a function has a **LOCAL SCOPE** and can only be accessed within that function:

# PHP Variables Scope-

## The global Keyword

### Example

```
<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest(); // run function
echo $y; // output the new
value for variable $y
?>
```

### Output

15

The **global** keyword is used to access a global variable from within a function

# PHP Comparison Operators

Operator	Name	Example	Result
==	Equal	$\$x == \$y$	Returns true if \$x is equal to \$y
===	Identical	$\$x === \$y$	Returns true if \$x is equal to \$y, and they are of the same type
!= <>	Not equal	$\$x != \$y$	Returns true if \$x is not equal to \$y
>	Greater than	$\$x > \$y$	Returns true if \$x is greater than \$y
<	Less than	$\$x < \$y$	Returns true if \$x is less than \$y
>=	Greater than or equal to	$\$x >= \$y$	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	$\$x <= \$y$	Returns true if \$x is less than or equal to \$y

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

**conditions & Loops,**

Functions,

String manipulation,

Arrays & Functions,

Form handling, Cookies

& Sessions, using

MySQL with PHP,

# Conditions and Loops

---

Conditional  
Statements

- If else
- Elseif  
ladder
- Switch

Loop  
Statements

- While
- Do while
- For
- Foreach



# If...else

---

## Syntax

```
If(condition)
    statements;
else
    statements;
```

## Example

```
<?php
$n=5;
If($n%2 == 0)
Echo "Number is Even";
Else
Echo "Number is Odd";
?>
```

# Elseif

---

## Syntax

If(condition)

statements;

Elseif(condition)

statements;

Else

statements;

## Example

```
<?php
$day=date("l");
If($day == "Saturday"))
Echo "Happy Weekend";
Elseif($day == "Sunday"))
Echo "Happy Sunday";
Else
Echo "Nice Working day";
?>
```

# Switch case

## Syntax

```
Switch(expression)
{
  Case
    constant_express
ion: statements;
    break;
  Default:
    statement
s;
```

## Example

```
<?php
$favcolor = "red";
switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    default:
        echo "Your favorite color is neither red,
blue!";
} ?>
```

# While Loop

## Syntax

```
while (condition is  
true)  
{  
    code to be  
    executed;  
}
```

## Example

```
<?php  
$x = 1;  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

# Do-While Loop

## Syntax

```
do {  
    code to be  
    executed;  
}  
while (condition is  
      true);
```

## Example

```
<?php  
$x = 1;  
do {  
    echo "The number is: $x  
    <br>";  
    $x++;  
} while ($x <= 5);  
?>
```

# For Loop

---

## Syntax

```
for (init counter; test counter; increment  
counter)  
{  
    code to be executed;  
code to be executed;  
}
```

## Example

```
<?php  
for ($x = 0; $x <= 10; $x++)  
{  
    echo "The number is: $x <br>";  
}  
?>
```

# Foreach Loop

---

## Syntax

```
foreach ($array as  
    $value) { code to be  
    executed;  
}
```

## Example

```
<?php  
$colors = array("red", "green", "blue");  
foreach ($colors as $value)  
    {  
        echo "$value <br>";  
    }  
?>
```

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

**Functions,**

String manipulation,

Arrays & Functions,

Form handling,

Cookies & Sessions,

using MySQL with PHP,



# User Defined Functions

---

## Syntax

```
function  
functionName() {  
code to be executed;  
}
```

## Example

```
<?php  
function writeMsg()  
{  
    echo "Hello world!";  
}  
writeMsg();  
?>
```

# Parameterized Functions

## Example

```
<?php
function Add($a,$b) {
    $sum=$a+$b;
    echo "Sum is $sum";
}
Add(10,20);
?>
```

## Output

Sum is 30

# Returning value through function

## Example

```
<?php
function Add($a,$b) {
    $sum=$a+$b;
    return $sum;
}
$Result=Add(10,20);
echo "Sum is $Result";
?>
```

## Output

Sum is 30

# Setting default values for function parameter

## Example

```
<?php
function Add($a,$b=300)
Add($a,$b=300)
{
    $sum=$a+$b;
    echo "Sum is $sum";
}
Add(10);
Add(10,20);
?>
```

## Output

```
Sum is 310
Sum is 30
```

# Dynamic Function Calls

## Example

```
<?php  
function Hello() {  
    echo "Hello How R U?";  
}  
$fh = "Hello";  
$fh();  
?>
```

## Output

Hello How R U?

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

**String manipulation,**

Arrays & Functions,

Form handling,

Cookies & Sessions,

using MySQL with PHP,

# String Manipulation

String Function	Description	Example
<code>strlen()</code>	returns the length of a string	<pre>&lt;?php echo strlen("Hello world!"); ?&gt; // outputs 12</pre>
<code>str_word_count()</code>	counts the number of words	<pre>&lt;?php echo str_word_count("Hello world!"); ?&gt; // outputs 2</pre>
<code>strrev()</code>	reverses a string	<pre>&lt;?php echo strrev("Hello world!"); ?&gt; // outputs !dlrow olleH</pre>
<code>strpos()</code>	searches for a specific text within a string.	<pre>&lt;?php echo strpos("Hello world!", "world"); ?&gt; // outputs 6</pre>
<code>str_replace()</code>	replaces some characters with some other characters in a	<pre>&lt;?php echo str_replace("Ram", "Holly", "Hello Ram"); ?&gt; //output Hello Holly</pre>

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

String manipulation,

**Arrays & Functions,**

Form handling,

Cookies & Sessions,

using MySQL with PHP,



# Arrays

---

## Create an Array in PHP

- In PHP, the array() function is used to create an array:
  - array();
- In PHP, there are three types of arrays:
  - **Indexed arrays** - Arrays with a numeric index
  - **Associative arrays** - Arrays with named keys
  - **Multidimensional arrays** - Arrays containing one or more arrays

# Arrays- Indexed Arrays

---

- There are two ways to create indexed arrays:
  1. The index can be assigned automatically as below :
    - `$cars = array("Volvo", "BMW", "Toyota");`
  - The index can be assigned manually:
    2. The index can be assigned manually:
      - `$cars[0] = "Volvo";`  
`$cars[1] = "BMW";`  
`$cars[2] = "Toyota";`

# Arrays- Indexed Arrays

---

- **To create and print array**

## Example-

```
<?php
$scars = array("Volvo", "BMW", "Toyota");
echo "I like " . $scars[0] . ", " . $scars[1] . " and " . $scars[2] . ".";
?>
```

- **Get The Length of an Array - The count() Function**

## Example

- ```
<?php
$scars = array("Volvo", "BMW", "Toyota");
echo count($scars);
?>
```

# Arrays- Indexed Arrays

---

Loop Through an Indexed Array Example

```
<?php
$scars = array("Volvo", "BMW", "Toyota");
$arrlength = count($scars);

for($x = 0; $x < $arrlength; $x++)
{
    echo $scars[$x];
    echo "<br>";
} ?>
```

```
<?php
$scars = array("Volvo", "BMW", "Toyota");

foreach($scars as $a)
{
    echo $a;
}
?>
```

# Arrays- Associative Arrays

---

- Associative arrays are arrays that use named keys that you assign to them.
- There are two ways to create an associative array:
- Method -1
  - `$age = array("Peter"=>"35", "Ben"=>"37");`
- Method- 2
  - `$age['Peter'] = "35";`
  - `$age['Ben'] = "37";`

# Arrays- Associative Arrays

## Example

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```

- **Output:**
- Peter is 35 years old.

## Loop Through an Associative Array

### Example

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
foreach($age as $x => $x_value) {
    echo "Key=" . $x . "Value=" . $x_value;
    echo "<br>";
} ?>
```

- **Output:**
- Key=Peter, Value=35  
Key=Ben, Value=37  
Key=Joe, Value=43

# Arrays- Multidimensional Arrays

PHP understands multidimensional arrays that are two, three, four, five, or more levels deep.

**PHP - Two-dimensional Arrays:** A two-dimensional array is an array of arrays First, take a look at the following table:

| Name  | Stock | Sold |
|-------|-------|------|
| Volvo | 22    | 18   |
| BMW   | 15    | 13   |

## Example

```
$cars = array (  
    array("Volvo",22,18),  
    array("BMW",15,13) );
```

# Functions on Array

---

- **print\_r()** –Prints all elements of an array in standard format
- **extract()**-Converts array into variables
- **compact()**-Converts group of variables into array
- **is\_array()** –to check weather a particular elements is an array or not.
- **sort()** - sort arrays in ascending order
- **rsort()** - sort arrays in descending order,
- **asort()**-sorts associative arrays in **ascending order**, on **values**
- **ksort()**-sorts associative arrays in **ascending order**,on **keys**
- **arsort()**- sorts associative arrays in **descending order**,on **values**
- **krsort()**-sorts associative arrays in **descending order**,on **keys**



# Functions on Array: **print\_r()**

## Example

```
<?php
$a =
array ('a' => 'apple', 'b' =>
'banana') print_r ($a);
?>
```

## Output

```
Array
(
    [a] => apple
    [b] => banana
)
```

# Functions on Array: **extract()**

## Example

```
<?php
$my_array = array("Rno" => "1",
"Name" => "Anand",
"Class" => "TE Comp");
extract($my_array);
echo $Rno;
echo $Name;
echo $Class;
?>
```

## Output

```
1
Anand
TE Comp
```

# Functions on Array: compact()

## Example

```
<?php
$firstname = "Peter";
$lastname = "Griffin";
$age = "41";
$result = compact("firstname",
"lastname", "age");
print_r($result); ?>
```

## Output

```
Array
(
    [firstname] => Peter
    [lastname] => Griffin
    [age] => 41
)
```

# Functions on Array: `sort()`

```
<?php
$numbers = array(4, 6, 2, 22, 11);

sort($numbers); //sort in ascending
print_r
($numbers);
rsort($numbers) //sort in descending
; print_r
($numbers);
?>
```

## Output :

```
Array
(
    [0] => 2
    [1] => 4
    [2] => 6
    [3] => 11
    [4] => 22
)
Array
(
    [0] => 22
    [1] => 11
    [2] => 6
    [3] => 4
    [4] => 2
)
```

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

String manipulation,

Arrays & Functions,

**Form handling,**

Cookies & Sessions,

using MySQL with PHP,

# Form Handling- using Post Method

---

a.html

```
<form action="welcome.php" method="post">  
method='post'> Name:  
  <input type="text" name='t1'>  
  <br>  
  <input type="submit">  
</form>
```

Welcome.php

```
Welcome  
<?php  
echo $_POST["t1"];  
$_POST["t1"];  
?>
```

# Form Handling- using Get Method

---

a.html

```
<form action="welcome.php"
method="get"> Name:
<input type="text" name="t1">
<input type="submit">
</form>
```

Welcome.php

```
Welcome
<?php
echo
$_GET["t1"];
?>
```

# Form Handling-

## Difference between get and post method

---

`$_GET` is an array of variables passed to the current script via the **URL parameters**.

`$_POST` is an array of variables passed to the current script via the **HTTP POST method**.

### When to use GET?

Information sent from a form with the GET method is **visible to everyone**. GET also has limits on the amount of information to send. The limitation is about 2000 characters.

### When to use POST?

Information sent from a form with the POST method is **invisible to others** and has **no limits** on the amount of information to send.

However, because the variables are not displayed in the URL, it is not possible to bookmark the page.



# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

String manipulation,

Arrays & Functions,

Form handling,

**Cookies & Sessions,**

using MySQL with PHP,

# Cookie

---

- **What is a Cookie?**
- A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.
- **Create Cookies With PHP-** is created with the `setcookie()` function.
- **Syntax**
- `setcookie(name, value, expire, path, domain, secure, httponly);`
- Only the *name* parameter is required. All other parameters are optional.

# Cookie-Create/Retrieve a Cookie

---

```
<?php  
setcookie("name", "Amit", time() + (86400 * 30), "/"); // 86400 = 1 day  
?>
```

```
<?php if(isset($_COOKIE["name"]))  
{  
    $nm=$_COOKIE["name"];  
    echo "Hello",$nm;  
}  
else { echo "Coocke is not set"; } ?>
```

# Cookie-Modifying a Cookie

---

To modify a cookie, just set (again) the cookie using the `setcookie()` function:

# Cookie- Deleting Cookies

---

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

# Sessions

---

- A session is a way to store information (in variables) to be used across multiple pages.
- Unlike a cookie, the information is not stored on the users computer.
- **What is a PHP Session?**
- When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.
- Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.
- So; Session variables hold information about one single user, and are available to all pages in one application.

# Sessions- Start a Session

---

- A session is started with the `session_start()` function.
- Session variables are set with the PHP global variable: `$_SESSION`.
- **Example - `demo_session1.php`**

```
<?php
session_start(); // Start the session ?>
<html>
<body>
<?php
$_SESSION["favcolor"] = "green"; // Set session
variable echo "Session variables are set."; ?>
</body>
</html>
```

# Sessions: Get Session Variable Values

- Next, we create another page called "demo\_session2.php". From this page, we will access the session information we set on the first page ("demo\_session1.php").
- session variable values are stored in the global `$_SESSION` variable
- **Example- demo\_session2.php**
  - ```
<?php
session_start(); ?>
<html>
<body>
<?php
// Echo session variables that were set on previous page
echo "Favorite color is ". $_SESSION["favcolor"];
?>
</body>
</html>
```



# Sessions- Modify a Session

---

To change a session variable, just overwrite it:

```
<?php
session_start(); ?>
<html>
<body>
<?php
// to change a session variable, just overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>
</body>
</html>
```

# Sessions: Destroy a Session

---

To remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()`:

## Example-

```
<?php  
session_start(); ?>
```

```
<html>  
<body>
```

```
<?php  
session_unset(); // remove all session  
session_destroy(); variables  
?> // destroy the session
```

```
</body>  
</html>
```

# Outline

Introduction to PHP,

Features,

sample code,

PHP script working,

PHP syntax,

conditions & Loops,

Functions,

String manipulation,

Arrays & Functions,

Form handling,

Cookies & Sessions,

**Using MySQL with PHP,**

# Open a Connection to MySQL

---

```
<?php
// Create connection
$conn = new mysqli("localhost", "root", "");
//MySQLi extension (the "i" stands for improved)

// Check connection
if(!$conn){
    die('Could not connect: '.mysqli_connect_error());
}
echo 'Connected successfully<br/>';
?>
```

# Select Data From a MySQL Database

---

```
<?php
$conn = mysqli_connect('localhost', 'root', 'root', 'db1');
if(!$conn){
    die(mysqli_connect_error());
}
echo 'Connected successfully<br>';

$sql = 'SELECT * FROM STUD';
$rs=mysqli_query($conn, $sql);
$rows= mysqli_num_rows($rs); ?>
```

# Select Data From a MySQL Database

---

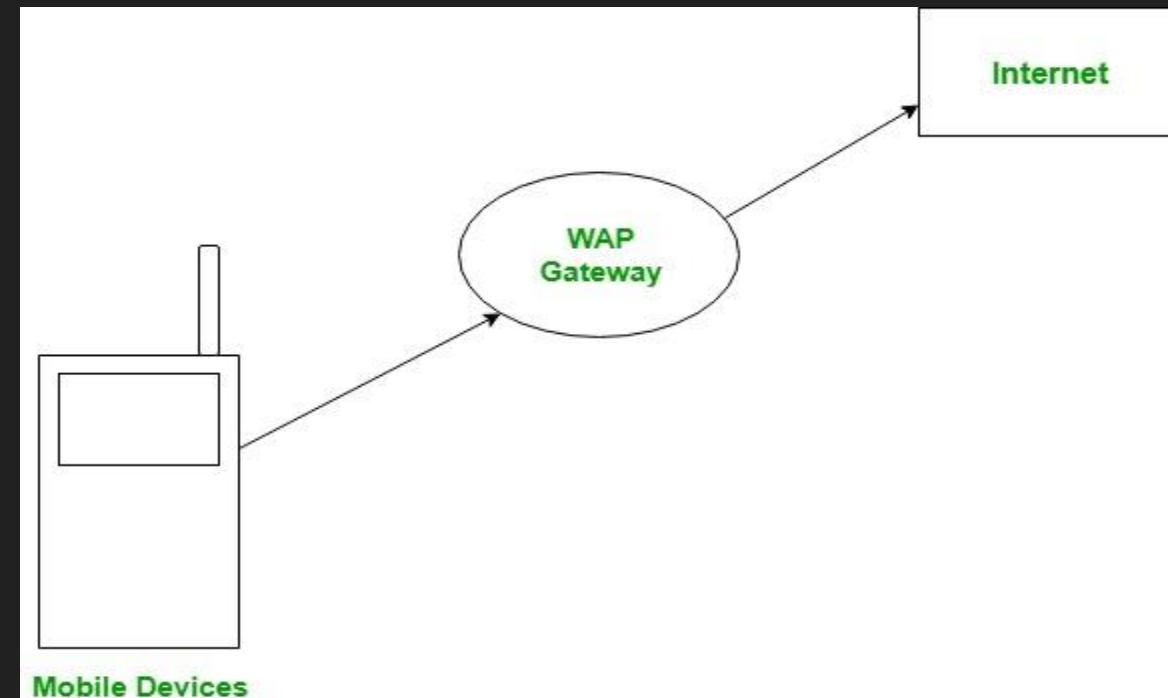
```
if($nrows > 0) {
  while($row = mysqli_fetch_assoc($rs)){
    echo "ID :{$row['id']} <br>";
    echo "FNAME : {$row['firstname']} <br>";
    echo "LNAME : {$row['lastname']} <br>";
    echo "-----<br>";
  }
}
else { echo " No result"; }
mysqli_close($conn);
?>
```

**WAP &**

**WML**

# WAP – Introduction (Wireless Application Protocol)

The basic aim of WAP is to provide a web-like experience on small portable devices - like mobile phones and PDAs



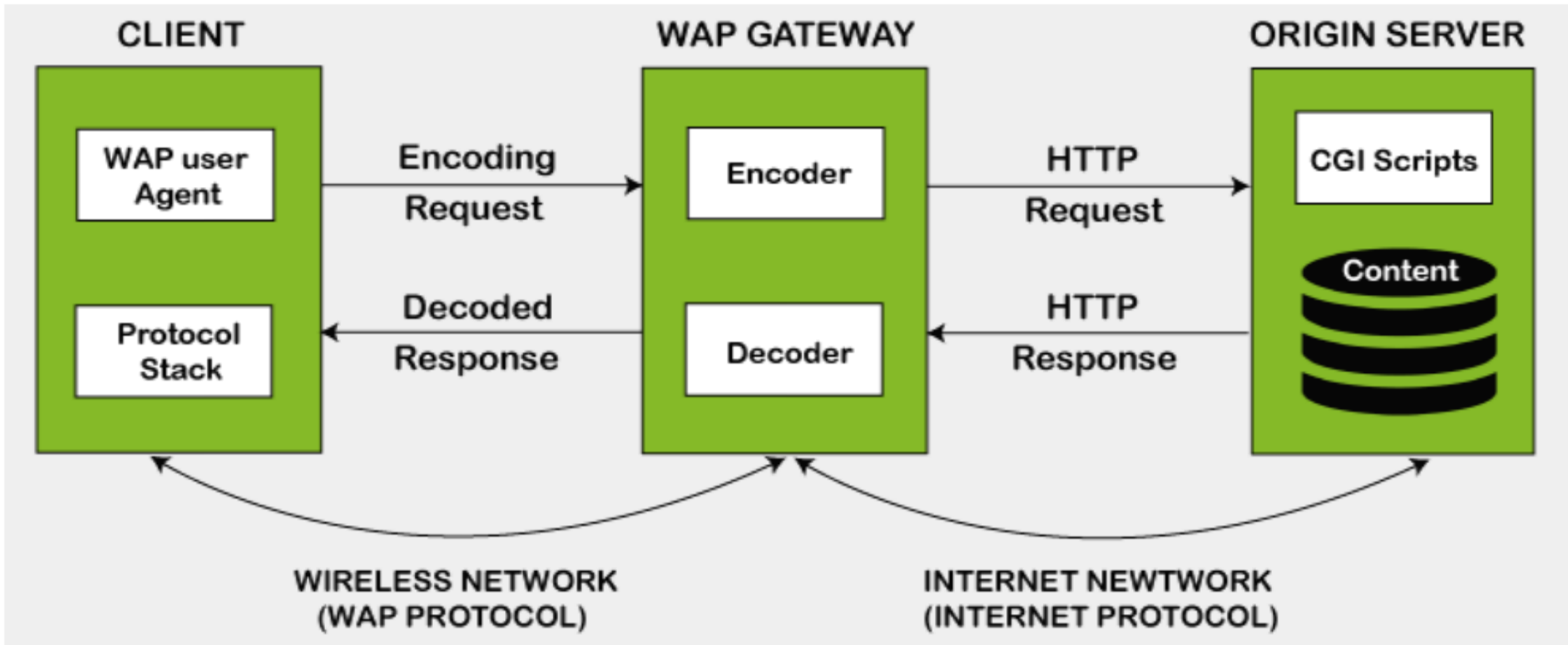


# WAP -Introduction

---

- **Purpose of WAP**
  - To enable easy, fast delivery of relevant information and services to mobile users.
- **Type of devices that use WAP**
  - Handheld digital wireless devices such as mobile phones, pagers, two-way radios, smart phones and communicators -- from low-end to high-end.
- **Type of OS that use WAP**
  - It can be built on any operating system including Palm OS, EPOC 32, Windows CE, FLEXOS, OS/9, Java O

# Working of WAP - Introduction

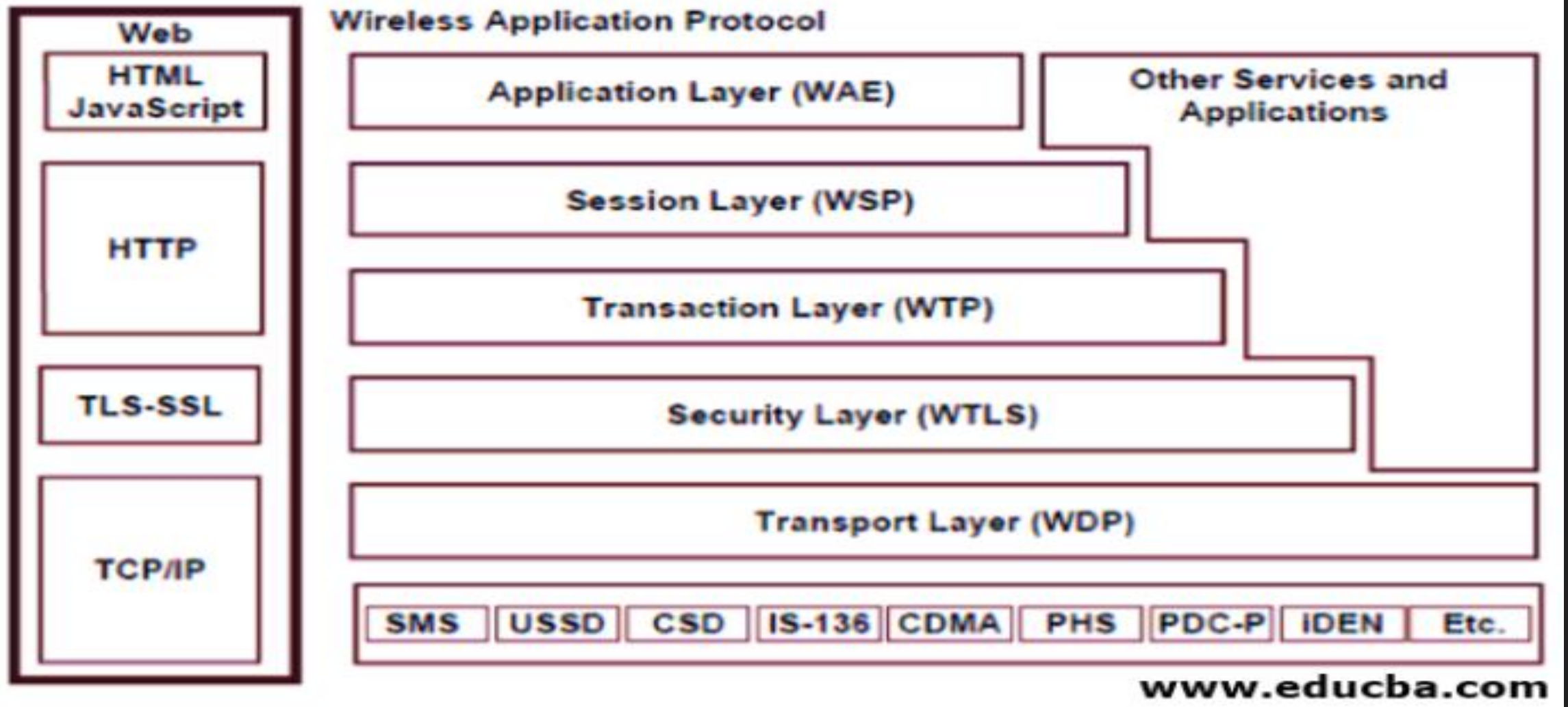


# Working of WAP -Introduction

---

- The WAP model consists of 3 levels known as Client, Gateway and Origin Server.
- When a user opens the browser in his/her mobile device and selects a website that he/she wants to view, the mobile device sends the URL encoded request via a network to a WAP gateway using WAP protocol.
- The request he/she sends via mobile to WAP gateway is called as encoding request.
- The sent encoding request is translated through WAP gateway and then forwarded in the form of a conventional HTTP URL request over the Internet.
- When the request reaches a specified Web server, the server processes the request just as it would handle any other request and sends the response back to the mobile device through WAP gateway.
- Now, the WML file's final response can be seen in the browser of the mobile users

# Components of WAP Architecture



# Components of WAP Architecture

---

## **Application Layer:**

This layer contains the Wireless Application Environment (WAE). It contains mobile device specifications and content development programming languages like WML.

## **Session Layer:**

This layer contains Wireless Session Protocol (WSP). It provides fast connection suspension and reconnection.

## **Transaction Layer:**

This layer contains Wireless Transaction Protocol (WTP). It runs on top of UDP (User Datagram Protocol) and is a part of TCP/IP and offers transaction support.

## **Security Layer:**

This layer contains Wireless Transaction Layer Security (WTLS). It offers data integrity, privacy and authentication.

## **Transport Layer:**

This layer contains Wireless Datagram Protocol. It presents consistent data format to higher layers of WAP protocol stack.

# Advantages of WAP

---

- WAP is a very fast-paced technology.
- It is an open-source technology and completely free of cost.
- It can be implemented on multiple platforms.
- It is independent of network standards.
- It provides higher controlling options.
- It is implemented near to Internet model.
- By using WAP, you can send/receive real-time data.
- Nowadays, most modern mobile phones and devices support WAP.

# Disadvantages of WAP

---

- The connection speed in WAP is slow, and there is limited availability also.
- In some areas, the ability to connect to the Internet is very sparse, and in some other areas, Internet access is entirely unavailable.
- It is less secured.
- WAP provides a small User interface (UI).

# Wireless Markup Language

---

- **WML** is the markup language defined in the WAP specification. WAP sites are written in **WML**, while web sites are written in HTML. **WML** is very similar to HTML. Both of them use tags and are written in plain text format.
- *WML* (Wireless Markup Language), formerly called **HDML (Handheld Devices Markup Languages)**, is a language that allows the text portions of Web pages to be presented on cellular telephones and personal digital assistants (PDAs) via wireless access.



# Wireless Markup Language(WML)

---

The topmost layer in the WAP (Wireless Application Protocol) architecture is made up of WAE (Wireless Application Environment), which consists of WML and WML scripting language.

- WML stands for **Wireless Markup Language**
- WML is an application of XML, which is defined in a document-type definition.
- WML takes care of the small screen and the low bandwidth of transmission.
- WML is the markup language defined in the WAP specification.
- WAP sites are written in WML, while web sites are written in HTML.
- WML is very similar to HTML. Both of them use tags and are written in plain text format.
- WML files have the extension ".wml". The MIME type of WML is "text/vnd.wap.wml".
- WML supports client-side scripting. The scripting language supported is called WMLScript.

# Wireless Markup Language(WML)

---

- **Text and Images:** WML gives a clue about how the text and images can be presented to the user. The final presentation depends upon the user. Pictures need to be in WBMP format and will be monochrome.
- **User Interaction:** WML supports different elements for input like password entry, option selector and text entry control. The user is free to choose inputs such as keys or voice.
- **Navigation:** WML offers hyperlink navigation and browsing history.
- **Context Management:** The state can be shared across different decks and can also be saved between different decks.

# Wireless Markup Language

**WML follows a deck and card.**

- A WML document is made up of multiple cards.
- Cards can be grouped together into a deck.

**A WML deck is similar to an HTML page.**

- A user navigates with the WML browser through a series of WML cards.

# WML Elements

`<p> </p>`

text

`<a href=...> </a>`

hyperlink (anchor)

`<do> </do>`

action

`<go href=.../>`

goto wml page

`<timer>`

trigger event (units = tenths of a second)

`<input/>`

input user text

`<prev/>`

return to previous page

`$(...)`

value of variable display

`<img src=... />`

image

`<postfield name=...`

set variable

`<select > <option> <option> </select>`

select box

# WML Structure

```
< ? xml version="1.0" ? >  
<!DOCTYPE wml ...>  
<wml>  
  <card>  
    <p>  
      Text....  
    </p>  
    <p>  
      Text.....  
    </p>  
  </card>  
  <card>  
  ...  
  </card>  
</wml>
```

# WML Example

```
<?xml version="1.0"?>
<wml>
  <card id="one" title="First Card">
    <p>
      This is the first card in the deck
    </p>
  </card>
  <card id="two" title="Second Card">
    <p>
      This is the second card in the deck
    </p>
  </card>
</wml>
```

# WML Scripts

---

- WMLScript (Wireless Markup Language Script) is the client-side scripting language of WML(Wireless Markup Language).
- A scripting language is similar to a programming language, but is of lighter weight.
- With WMLScript, the wireless device can do some of the processing and computation.
- This reduces the number of requests and responses to/from the server.

# WML Scripts

---

## WML Script Operators

WML Script supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators



# WML Scripts

## WML Script Control Statements

Control statements are used for controlling the sequence and iterations in a program.

Statement	Description
if-else	Conditional branching
for	Making self-incremented fixed iteration loop
while	Making variable iteration loop
break	Terminates a loop
continue	Quit the current iteration of a loop

# WML Scripts

---

## WML Scripts Comments

There are two types of comments in WMLScript –

**Single-line comment** – To add a single-line comment, begin a line of text with the // characters.

**Multi-line comment** – To add a multi-line comment, enclose the text within /\* and \*/.

# WAP Applications

---

- Banking:
- Finance:
- Shopping:
- Ticketing:
- Entertainment:
- Weather:
- E- Messaging:

Asp.Net

# ASP.NET

---

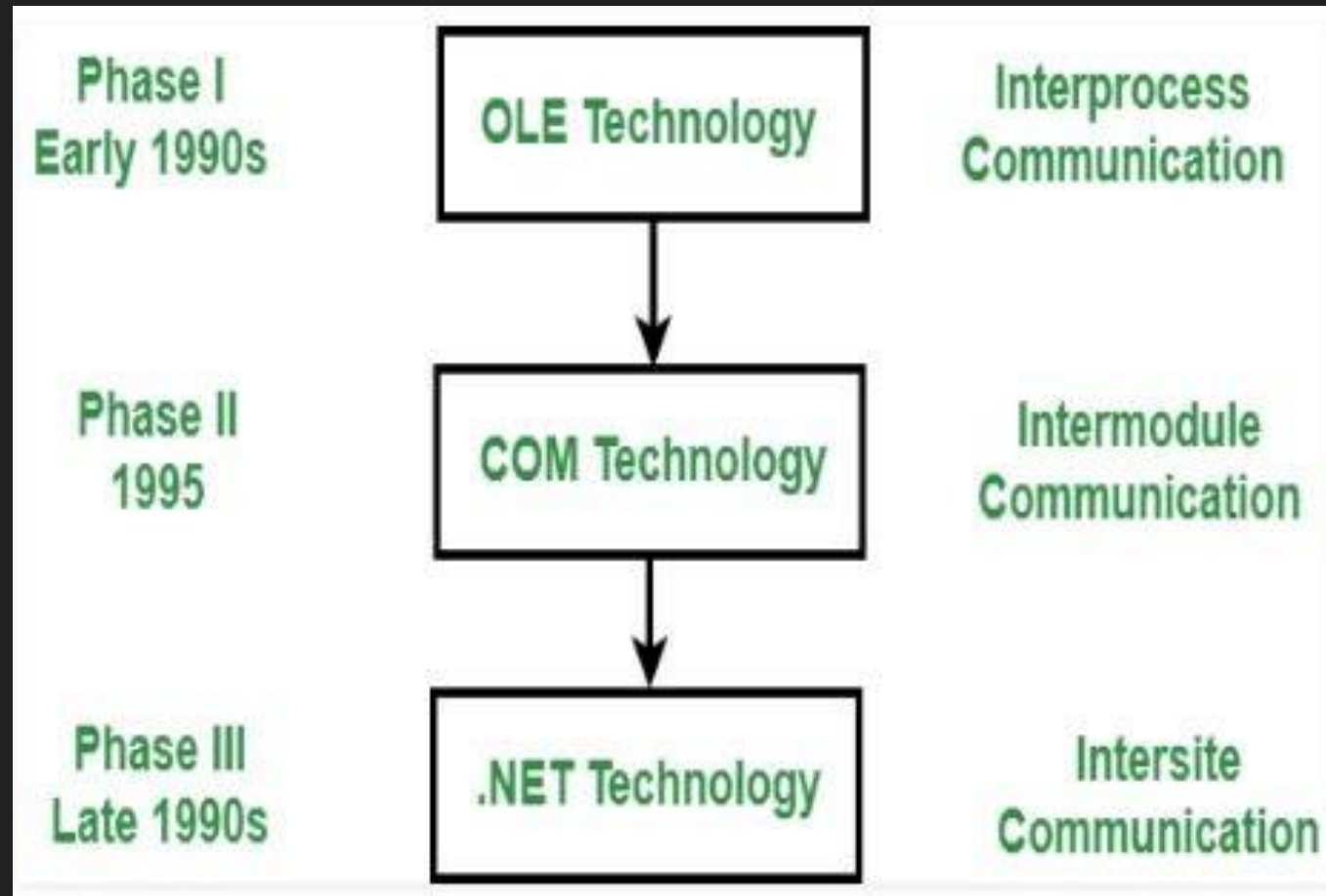
- Overview of the .NET Framework,
- Overview of C#,
- Introduction to ASP.NET,
- ASP.NET Controls,
- Web Services

# .NET history

---

- .NET Framework **supports more than 60 programming languages** in which **11 programming languages are designed and developed by Microsoft**. The remaining Non-Microsoft Languages are supported by .NET Framework but not designed and developed by Microsoft.
- There are **three significant phases of the development of .NET technology**.
  - OLE Technology (**Object Linking and Embedding**)
  - COM Technology (**Component Object Model**)
  - .NET Technology (**Network Enabled Technologies**)

# .NET history



# .NET history

---

- The .Net framework was meant to create applications, which would run on the Windows Platform.
- The **first version** of the .Net framework was released in the **year 2002**.
- The version was called .Net framework 1.0. The Microsoft .Net framework has come a long way since then, and the current version is .Net Framework 4.7.2.
- The Microsoft .Net framework can be used to **create both – Form-based and Web-based applications**.
- **Web services** can also be developed using the .Net framework.



# .NET history

---

11 Programming Languages which are designed and developed by Microsoft are:

C#.NET

VB.NET

C++.NET

J#.NET

F#.NET

JSCRIPT.NET

WINDOWS POWERSHELL

IRON RUBY

IRON PYTHON

C OMEGA

ASML(Abstract State Machine Language)

# .NET Framework Architecture

---

- .Net Framework Architecture is a programming model for the .Net platform that provides an execution environment and integration with various programming languages for simple development and deployment of various Windows and desktop applications.
- It consists of class libraries and reusable components.

- WinForms

- **ADON.NET Framework Architecture**

# .NET Framework Architecture

---

## 1. Common Language Runtime

The “Common Language Infrastructure” or CLI is a platform in .Net architecture on which the .Net programs are executed.

The CLI has the following key features:

1. Exception Handling
2. Garbage Collection
3. Language
4. Compiler
5. Common Language Interpreter

# .NET Framework Architecture

---

## 2. Class Library

- The .NET Framework includes a set of standard class libraries.
- A class library is a collection of methods and functions that can be used for the core purpose.
  - For example, there is a class library with methods to handle all file-level operations.
  - So there is a method which can be used to read the text from a file.
  - Similarly, there is a method to write text to a file.

# .NET Framework Architecture

---

## 3. Languages

The types of applications that can be built in the .Net framework is classified broadly into the following categories.

1. **WinForms** – This is used for **developing Forms-based applications**, which would run on an end user machine. Notepad is an example of a client-based application.
2. **ASP.Net** – This is used for **developing web-based applications**, which are made to run on any browser such as Internet Explorer, Chrome or Firefox.
3. **ADO.Net** – This technology is used to **develop applications to interact with Databases** such as Oracle or Microsoft SQL Server.

**Overview of**

**C#**

# C# introduction

---

## What is C#?

C# is pronounced "C-Sharp".

It is an **object-oriented programming language** created by Microsoft that runs on the

.NET Framework. C# has roots from the C family, and the language is **close to other**

**popular languages like C++ and Java.** The first version was released in year 2002.



# C# introduction

---

C# is used for:

- Mobile applications
- Desktop applications
- Web applications
- Web services
- Web sites
- Games
- VR
- Database applications
- And much, much more!

# C# -Example

---

Program.cs

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

# C#- Example Explained

---

## Example explained

- **Using System** means that we can use classes from the System namespace.
- **namespace** is used to organize your code, and it is a container for classes and other namespaces.
- **class** is a container for data and methods, which brings functionality to your program.
- Another thing that always appear in a C# program, is the Main method.
- **Console** is a class of the System namespace, which has a **WriteLine()** method that is used to output/print text.
- If you omit the using System line, you would have to write System.Console.WriteLine() to print/output text.

# C# Variables

---

**int** - stores integers (whole numbers), without decimals, such as 123 or -123 (**int myNum = 5;** )

**double** - stores floating point numbers, with decimals, such as 19.99 or -19.99 (**double**

**myDoubleNum = 5.99D;**) **char** - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes (**char myLetter = 'D';** )

**string** - stores text, such as "Hello World". String values are surrounded by double quotes (**string myText = "Hello";** )

**bool** - stores values with two states: true or false (**bool myBool = true;** )

# C# Variables

---

## Syntax

```
type variableName = value;
```

## Example

```
string name = "John"; Display Variables
```

**Example** string name = "John";

```
Console.WriteLine("Hello " + name);
```

## Declare Many Variables

```
int x = 5, y = 6, z = 50;
```

# c#- Get User Input

---

Use `Console.ReadLine()` to get user input.

Example

```
Console.WriteLine("Enter username:");
```

```
// Create a string variable and get user input from the keyboard and store it in
```

```
the variable string userName = Console.ReadLine();
```

# C# Assignment Operators

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	$x / y$
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$x++$
--	Decrement	Decreases the value of a variable by 1	$x--$

# C# Conditional Statements

---

C# has the following conditional statements:

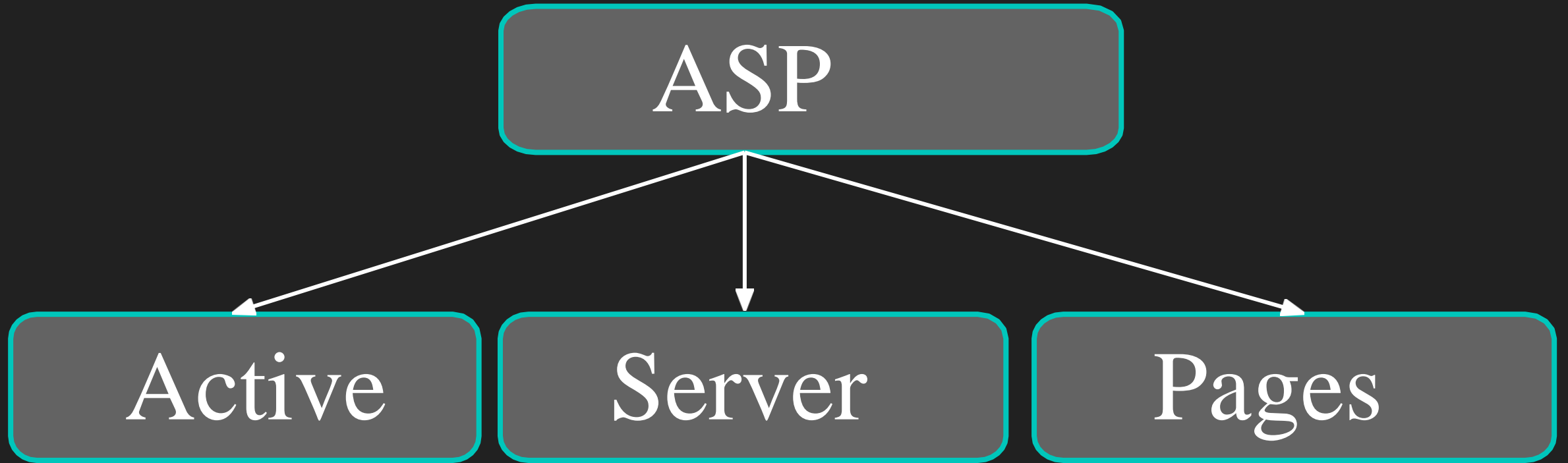
- Use **if** to specify a block of code to be executed, if a specified condition is true
- Use **else** to specify a block of code to be executed, if the same condition is false
- Use **else if** to specify a new condition to test, if the first condition is false
- Use **switch** to specify many alternative blocks of code to be executed



Asp.Net

# ASP.NET introduction

---



# ASP.NET introduction

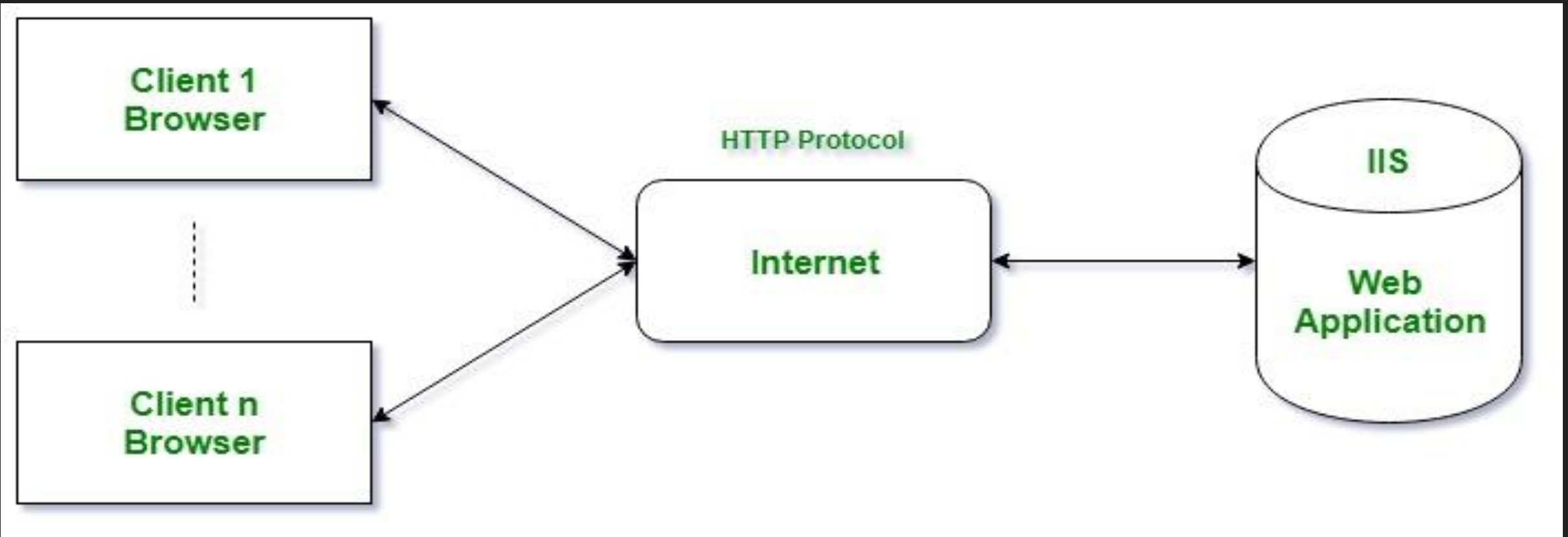
---

- ASP.NET is a **web application** framework **developed and marketed by Microsoft** to allow programmers to **build dynamic web sites.**
- ASP is a development framework for building web pages.
- ASP supports many different development models:
  - Classic ASP
  - ASP.NET Web Forms
  - ASP.NET MVC
  - ASP.NET Web Pages
  - ASP.NET API
  - ASP.NET Core

# ASP.NET introduction

## What is Web Application?

A web application is an application installed only on the web server which is accessed by the users using a web browser like Microsoft Internet Explorer, Google Chrome, Mozilla FireFox, Apple Safari, etc.



# ASP.NET introduction

---

The ASP.NET application codes can be written in any of the following languages:

- C#
- Visual Basic.Net
- Jscript
- J#

# ASP.NET introduction

---

ASP supports a lot of development models which are as follows:

1. **Classic ASP**: It is the first **server side scripting language** developed by Microsoft.
2. **ASP.NET**: It is **web development framework**
3. **ASP.NET Core**:, it is considered as an important **redesign of ASP.NET with the feature of open-source** and **cross-platform**.
4. **ASP.NET Web Forms**: These are the **event-driven application model** which are not considered a part of the new **ASP.NET Core**. These are used to provide the **server-side events** and controls to develop a web application.
5. **ASP.NET MVC**: It is the Model-View-Controller application model which can be merged with the new **ASP.NET Core**. It is used to **build dynamic websites as it provides fast development**.
6. **ASP.NET Web Pages**: These are the **single page application** which can be merged into **ASP.NET Core**.
7. **ASP.NET API**: It is the **Web Application Programming Interface(API)**.

# ASP.NET Life Cycle

---

ASP.NET life cycle specifies, how:

- ASP.NET processes pages to produce dynamic output
- The application and its pages are instantiated and processed
- ASP.NET compiles the pages dynamically

The ASP.NET life cycle could be divided into two groups:

1. Application Life Cycle
2. Page Life Cycle

# ASP.NET Life Cycle

---

The **Application life cycle phases** are:

1. Request Phase
2. Response Phase



# ASP.NET Life Cycle

---

The **page life cycle phases** are:

1. Page Request
2. Start
3. Initialization
4. Load
5. Postback event handling
6. Rendering
7. Unloading

# ASP.NET Life Cycle

Stage	Description
<b>Page request</b>	This stage occurs before the lifecycle begins. When a page is requested by the user, ASP.NET parses and compiles that page.
<b>Start</b>	In this stage, page properties such as Request and response are set. It also determines the Request type.
<b>Initialization</b>	In this stage, each control's UniqueID property is set. Master page is applied to the page.
<b>Load</b>	During this phase, if page request is postback, control properties are loaded with information.
<b>Postback event handling</b>	In this stage, event handler is called if page request is postback. After that, the Validate method of all validator controls is called.
<b>Rendering</b>	Before rendering, view state is saved for the page and all controls. During the rendering stage, the page calls the Render method for each control, providing a text writer that writes its output to the OutputStream object of the page's Response property.
<b>Unload</b>	At this stage the requested page has been fully rendered and is ready to terminate. At this stage all properties are unloaded and cleanup is performed.

# ASP.NET - Basic Controls

---

1. Button Controls
2. Text Boxes and Labels
3. Check Boxes and Radio Buttons
4. List Controls
5. The ListItemCollection
6. HyperLink Control
7. Image Control

# ASP.NET Web Services

---

A web service is a web-based functionality accessed using the protocols of the web to be used by the web applications.

There are three aspects of web service development:

1. Creating the web service
2. Creating a proxy
3. Consuming the web service

**Nodejs**

# Node.js : Overview

---

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

# Node.js : Overview

---

## What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

## What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

# Why Node.js?

A common task for a web server can be to open a file on the server and return the content to the client.

## How PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

## How Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

- Node.js eliminates the waiting, and simply continues with the next request.
- Node.js runs single-threaded, non-blocking, asynchronous programming, which is very memory efficient.



# References

- <https://www.javatpoint.com/php-tutorial>
- <https://www.studytonight.com/php/introduction-to-php>
- <https://www.geeksforgeeks.org/wml-introduction/>
- [https://www.tutorialspoint.com/asp.net/asp.net\\_introduction.htm](https://www.tutorialspoint.com/asp.net/asp.net_introduction.htm)
- <https://learn.microsoft.com/en-us/aspnet/overview>
- [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)

# Thank You

---

[gharu.anand@gmail.com](mailto:gharu.anand@gmail.com)

**Blog : [anandgharu.wordpress.com](http://anandgharu.wordpress.com)**

**Prof. Gharu Anand N.**