Unit-4

Introduction to PHP

Content

- ✓ Introduction to PHP
- ✓ Datatypes
- ✓ Control Statements
- ✓ Loops
- ✓ Functions
- ✓ Embedding PHP in HTML & MySQL

What is PHP?

PHP is a server side scripting language, and a powerful tool for making dynamic and interactive Web pages

Hypertext Pre-processor

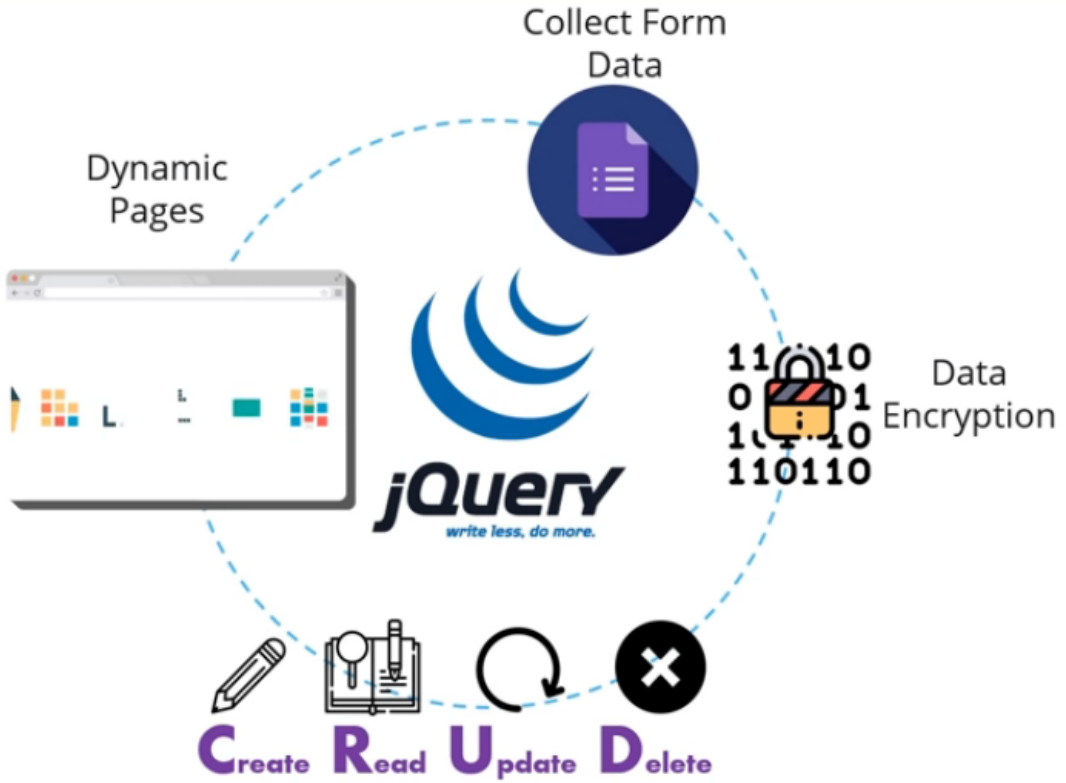
open source

Used for server side logic

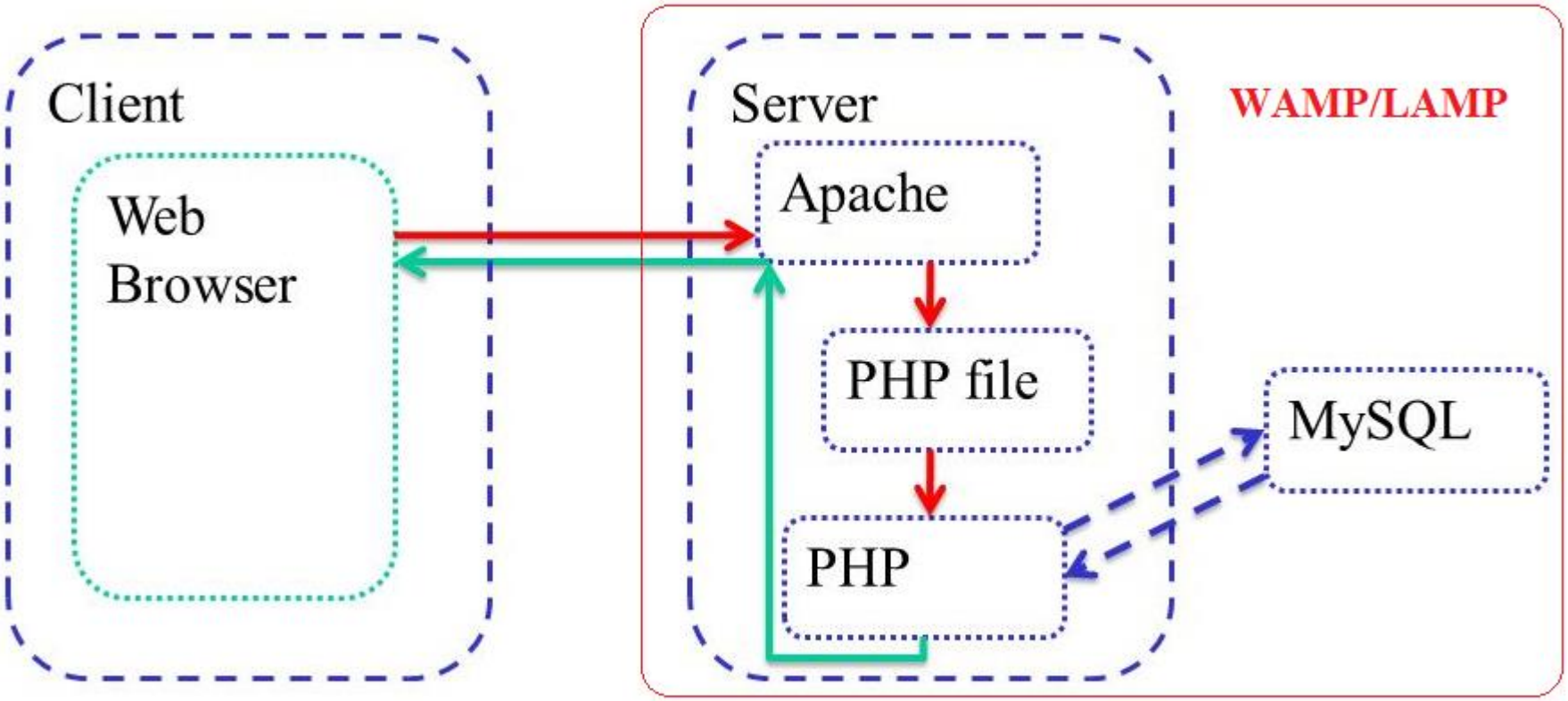
What can PHP do?

PHP Features

- Easy to Learn
- Supports different types of DB
- Cross Platform
- Supports multiple server types



PHP Architecture



This is often referred to as a LAMP (or WAMP) stack:

Linux (or Windows) Apache MySQL PHP

What is PHP?

→ The PHP Hypertext Pre-processor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases.

→ PHP is basically used for developing web based software applications.

→ PHP was written in the C programming language by Rasmus Lerdorf in 1994

→ Rasmus used PHP for monitoring his online resume and related personal information.

→ Originally PHP was an acronym for "Personal Home Page".

→ PHP processor has two modes of operations:

1. Copy mode 2. interpret mode

Applications of PHP

- PHP performs system functions - from files on a system it can create, open, read, write, and close them.
- PHP can handle forms - Gather data from files, save data to a file, through email you can send data, return data to the user.
- Add, delete, modify elements within your database through PHP.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Features of PHP

- → Performance - Script written in PHP executes much faster then those scripts written in other languages such as JSP & ASP.
- → Open Source Software - PHP source code is freely available on the web.
- → Platform Independent - A PHP application developed in one OS can be easily executed in other OS also.
- → Compatibility - PHP is compatible with almost all local servers used today like Apache, IIS etc.
- → Embedded - PHP code can be easily embedded within HTML tags and script.

Environment setup

- → PHP Parser-Parser must be installed to generate HTML output that can be sent to the Web Browser.
- → Database-Oracle and Sybase but most commonly used is freely available MySQL database.
- → Web Server-all Web Server software, including (IIS) but then most often used is freely available *Apache Server*.

Where to add PHP script?

→ All PHP code must be included inside one of the three special markup tags are recognised by the PHP Parser.

→ <?php PHP code goes here ?>

→ <script language="php"> PHP code goes here </script>

PHP: Print construct

→ echo – Used to print message/string on console or web page.

Example: echo "Hello by PHP echo";

→ If more than one parameter is used for displaying output, then use echo with parenthesis

Syntax: void echo (string $arg1 [, string $...])

→ print() – printing multi line or single line string with or without parenthesis

→ printf() – Printing multi line or single line string with parenthesis

→ escaping characters

→ echo “Hello!\“Welcome to PHP \” scripting language”

→ print “Hello!\“Welcome to PHP \” scripting language”

→ printf (“Hello!\“Welcome to PHP \” scripting language”)

PHP Variable Types

The main way to store information in the middle of a PHP program is by using a variable.

Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign ($).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.
- PHP variables are Perl-like.

PHP has a total of eight data types which we use to construct our variables:

PHP Variables

→ A variable is declared using $ sign followed by variable name.
Syntax: $variablename=value;

→PHP variables must start with letter or underscore only.

→PHP variable can't be start with numbers and special symbols.

→ Example:

```
<?php
$a="hello";//letter (valid)
$_b="hello";//underscore (valid)

echo "$a <br/> $_b";
?>
```

Variable scope

→PHP has three different variable scopes:

- →local
- →global
- → static

```
<?php
    $str="hello string";
    $a=23;
    $b=33.333;
    echo "string is: $str <br/>";
    echo "integer is: $a <br/>";
    echo "float is: $b <br/>";
?>
```

```
<?php
    $x=5;
    $y=6;
    $z=$x+$y;
    echo $z;
?>
```

Variable Types

- **Integers:** are whole numbers, without a decimal point, like 4195.
- **Doubles:** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans:** have only two possible values either true or false.
- **NULL:** is a special type that only has one value: NULL.
- **Strings:** are sequences of characters, like 'PHP supports string operations.'
- **Arrays:** are named and indexed collections of other values.
- **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources:** are special variables that hold references to resources external to PHP (such as database connections).

Comments in PHP

```
</head>

<body>

<?php

/*
multiline
*/

// Single line comments

# Another comment

?>

</body>
</html>
```

Date example

```
<?php date_default_timezone_set('UTC');
/* Echos the date
            h : 12 hr format
            H : 24 hr format
            i : Minutes
            s : Seconds
            u : Microseconds
            a : Lowercase am or pm
            l : Full text for the day
            F : Full text for the month
            j : Day of the month
            S : Suffix for the day st, nd, rd, etc
            Y : 4 digit y
         */
?>    echo date('h:i:s:u a, l F jS Y e');
```