PHP Course

Beginner Course on PHP

ABOUT ME

Aditya Dubey

National Science Exhibition | 3rd

Full Stack Web Developer | Entrepreneur | Trainer

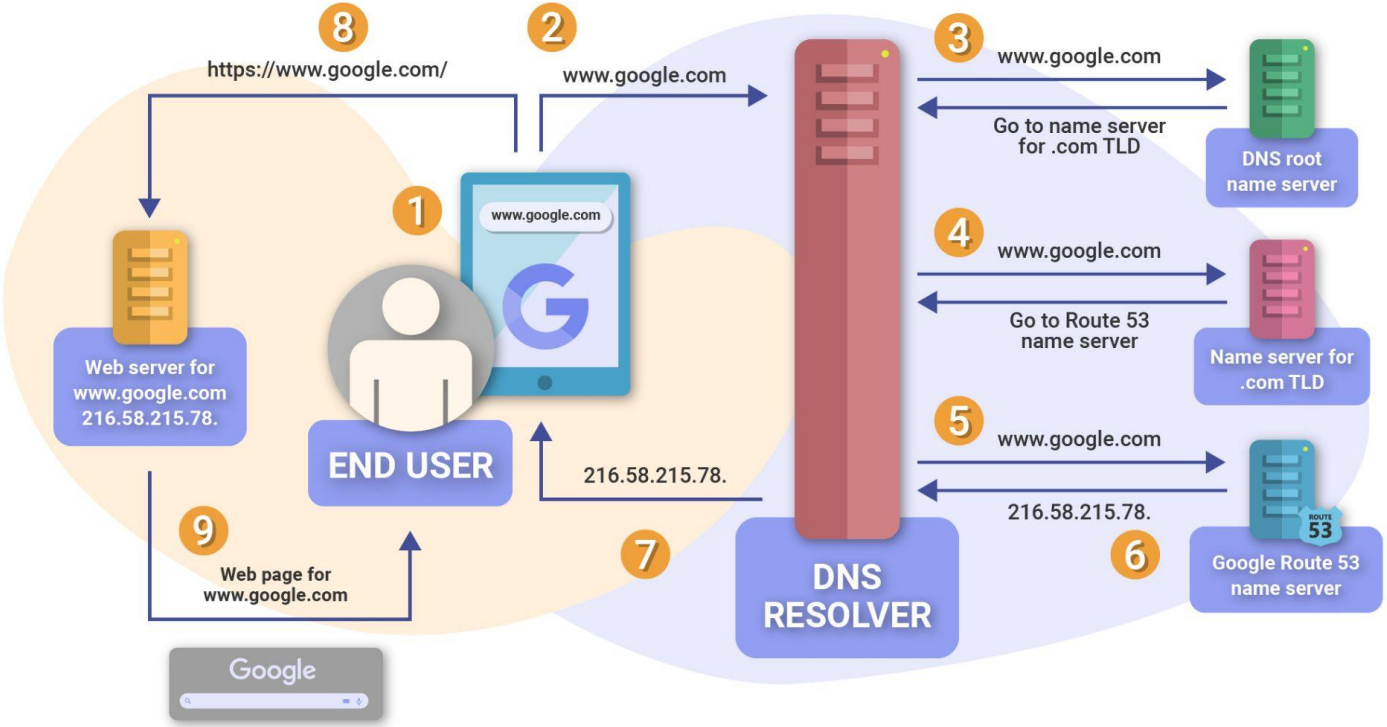
Lets Upgrade

Overview

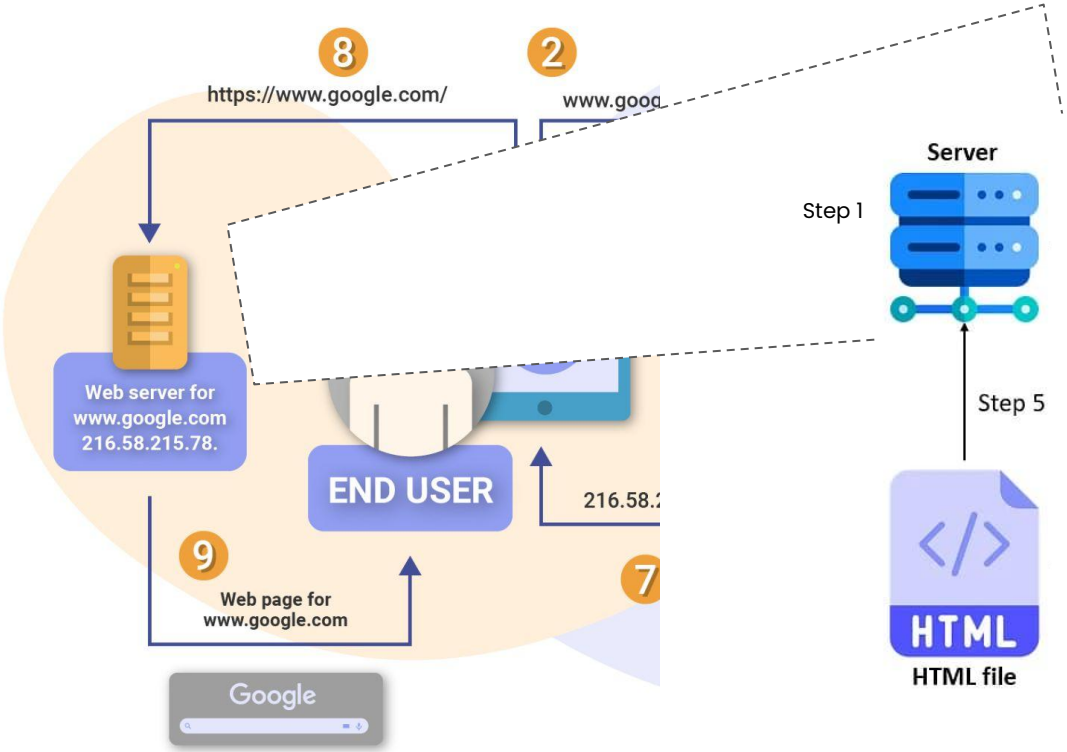
1. What is PHP?
2. What is the difference between programming and scripting Language?
3. What is the basic architecture of a PHP web application?
4. What are the benefits of using PHP?
5. What are some examples of PHP applications?
6. Getting started with PHP.
7. Variables and constants
8. Data types
9. Type casting and type juggling
10. Operators

How Internet Works

Internet


8
https://www.google.com/
2
www.google.com
1
www.google.com
Web server for
www.google.com
216.58.215.78.
END USER
9
Web page for
www.google.com
Google
216.58.215.78.
7
DNS
RESOLVER
3
www.google.com
Go to name server
for .com TLD
DNS root
name server
4
www.google.com
Go to Route 53
name server
Name server for
.com TLD
5
www.google.com
216.58.215.78.
6
ROUTE
53
Google Route 53
name server

Where is php here



What is PHP?

PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

PHP is a server side scripting language that is used to develop Static websites, Dynamic websites and Web applications. PHP scripts can only be interpreted on a server that has PHP installed.

A PHP file contains PHP tags and ends with the extension “.php”.

Programming Vs Scripting Language

Programming Language:

1. Has all the features needed to develop complete applications.
2. The code has to be compiled before it can be executed
3. Does not need to be embedded into other languages

Scripting Language:

1. It i mostly used in conjunction with other technologies.
2. The code is Interpreted on runtime.
3. Is usually embedded into other software environments.

Benefits of using PHP

- PHP is **open source and free.**
- Short learning curve compared to other languages such as JSP, ASP etc.
- Large community document
- Most web hosting servers support PHP by default unlike other languages such as ASP that need IIS. This makes PHP a cost effective choice.
- PHP is regular updated to keep abreast with the latest technology trends.
- Other benefit that you get with PHP is that it's a **server side scripting language**; this means you only need to install it on the server and client computers requesting for resources from the server do not need to have PHP installed; only a web browser would be enough.

Benefits of using PHP

- PHP has **in built support for working hand in hand with MySQL**; this doesn't mean you can't use PHP with other database management systems. You can still use PHP with
  - Postgres
  - Oracle
  - MS SQL Server
  - ODBC etc.
- PHP is **cross platform;** this means you can deploy your application on a number of different operating systems such as windows, Linux, Mac OS etc.

Where is PHP Used ?

Facebook

Wordpress

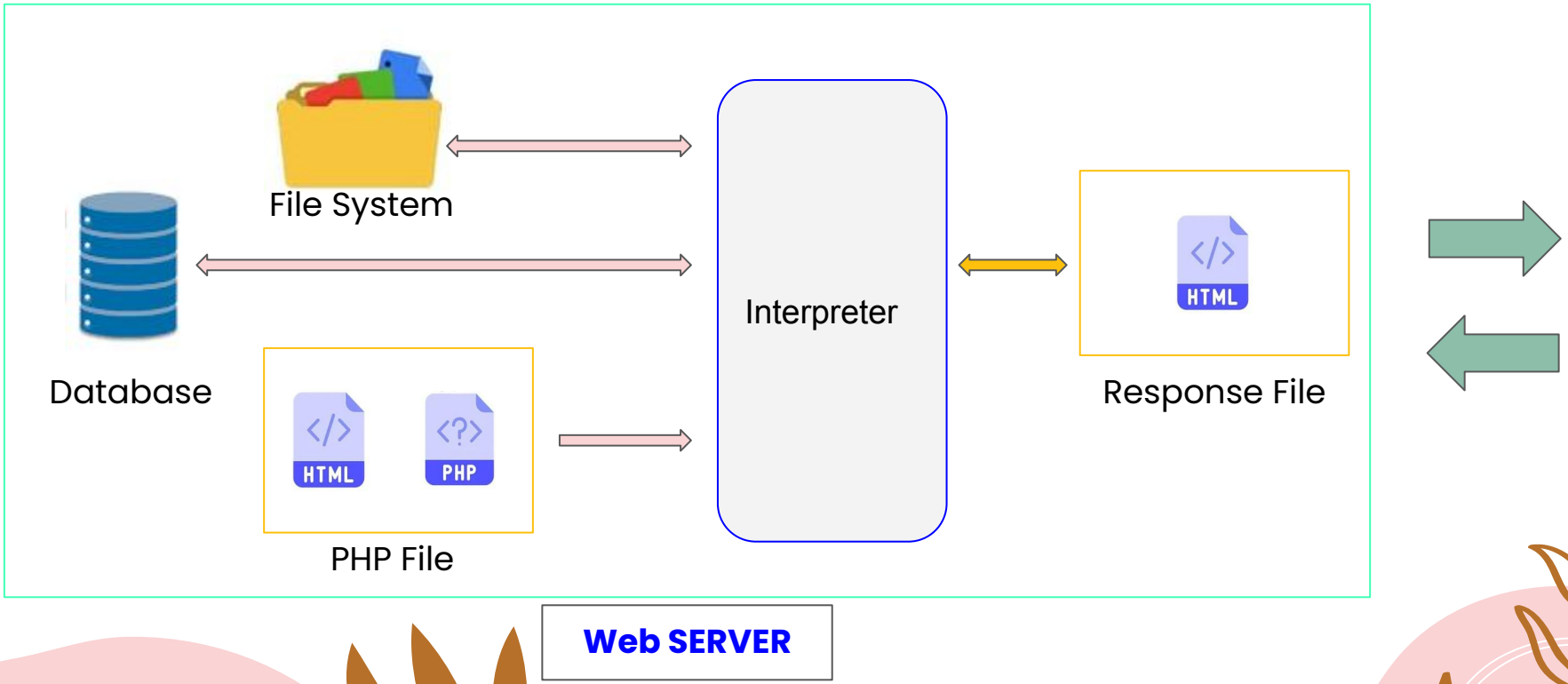
Wikipedia

Yahoo !

79-80% of all sites on Internet

In terms of market share, there are over 20 million websites and application on the internet developed using PHP scripting language.

Basic architecture of a PHP web application



01. Getting Started

Namaste World !

Online Interpreter

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_syntax

Program:

Syntax:

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Namaste World!";
?>
</body>
</html>
```

Result:

Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

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)

Constant

To create a constant, use the define() function.

Syntax

define(*name*, *value*, *case-insensitive*)

Parameters:

- *name*: Specifies the name of the constant
- *value*: Specifies the value of the constant
- *case-insensitive*: Specifies whether the constant name should be case-insensitive. Default is false

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

Type Casting and Type Juggling

Casting:

Typecasting is a way to convert one data type variable into different data types. A type can be cast by inserting one of the casts in front of the variable.

```
$a = “1”;

$b=(int)$a;
```

Juggling:

If an integer value is assigned to a variable, it becomes an integer. If a string value is assigned to the variable, it becomes a String.

```
$a=1;

$b='2';

$c=$a+$b;
```

Type casting and Type Juggling

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

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators

Arithmetic Operators

+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

Assignment Operators

x = y	x = y	The left operand is set to the value of the expression on right
x += y	x = x + y	Addition
x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

Comparison Operators

==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and the type is same
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or not same type
>	Greater than	$x > $y	Returns true if $x is greater than $y

Comparison Operators

<	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 or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less or equal to $y
<=>	Spaceship	$x <=> $y	Returns -1 , 0 , 1 for $x is <, =, > respectively.

Increment / Decrement Operators

++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

PHP Logical Operators

and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

Study Material

https://github.com/itsAdityaDubey/phpBeginners

Recap

1. What is PHP?
2. What is the difference between programming and scripting Language?
3. What are the benefits of using PHP?
4. What are some examples of PHP applications?
5. What is the basic architecture of a PHP web application?
6. Getting started with PHP.

THANKS!

Do you have any questions?
students@webhizzy.in
+9015090561

@itsadityadubey