Information
Technology
Institute

php

Prepared By: Eng. Noha Salah

July 2022

PHP

Day 1

Day 1 Contents

- Brief History of PHP.
- Why PHP?
- What can PHP do?
- Who uses PHP
- PHP Environment
- PHP Overview (Variables, Operators, Constants, Strings, Arrays, Flow control, ...)

Brief History of PHP

- Originally created by Rasmus Lerdorf in 1994. It was initially developed for HTTP usage logging and server-side form generation in Unix [PHP originally stood for Personal Home Page].
- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is now produced by The PHP development team http://www.php.net
- PHP 7.4.27 is the latest stable release.

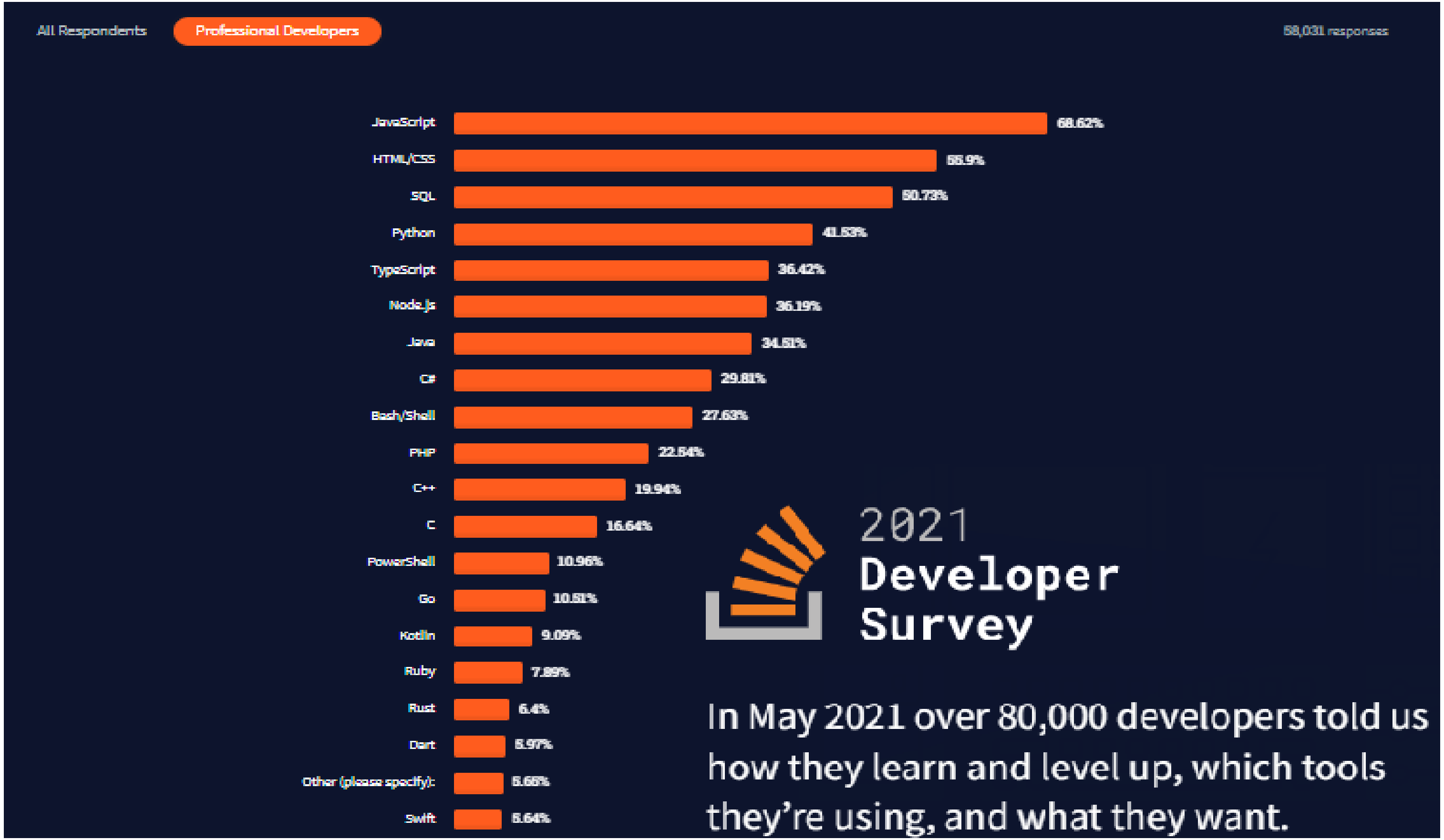
Why PHP?

- PHP is a server side scripting language commonly used to produce dynamic web pages on web servers.
- Open source with huge community.
- Embedded code within HTML.
- Platform-independent.
- compatible with almost all servers used today (Apache, IIS, etc.)
- Supports and interacts with many databases (MySQL , Oracle, Sybase, PostgreSQL, etc.)
- PHP has a syntax similar to C and Java.
- PHP is an interpreted language.

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 to access some pages of your website.
- PHP can encrypt data

Most Popular Technologies



In May 2021 over 80,000 developers told us how they learn and level up, which tools they're using, and what they want.

Who uses PHP

YAHOO!
WIKIPEDIA
The Free Encyclopedia
Joomla!
Woo

According to W3Techs' data, PHP is used by 78.9% of all websites with a known server-side programming language. So almost 8 out of every 10 websites that you visit on the Internet are using PHP in some way.

PHP Environment

In order to develop and run PHP Web pages three vital components need to be installed on your computer system:

Web Server - **Database** - **PHP Parser**

XAMPP is the most popular open source (completely free) PHP development environment that will install Apache, MySQL, PHP,

Perl, phpMyAdmin, and an FTP server .It is suitable for Linux, Solaris, and Windows systems and available at : **https://www.apachefriends.org/index.html**

PHP Fundamentals

overview

Syntax

- PHP is embedded within html pages within the tags: <?php .... ?>
- The short version of these tags can also be used: <? .... ?>
- Each line of PHP is terminated, like MySQL, with a semicolon ;
- Comments in PHP : // or # to make a single-line comment
  /* .... */ to make a Multi-lines comment block.
- Whitespace Insensitivity.
- Case Sensitivity for variables but classes, functions, and user-defined functions are not case-sensitive.

PHP- Output Statements

There are **two** basic ways to get output data to the screen:

echo and *print*

echo has no return value while print has a return value of 1 so it can be used in expressions.

echo can take multiple parameters while print can take only.

The echo / print statement can be used with or without parentheses: echo or echo().

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)

PHP- Variable Types

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

- 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 'I love PHP.'
- Arrays − are named and indexed collections of other values.
- Objects − are instances of 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).

PHP- Predefined Variables

Superglobals are built-in variables that are always available in all scopes.

- They are structured as associative arrays.
- $GLOBALS , $_SERVER , $_GET , $_POST , $_FILES , $_COOKIE , $_SESSION , $_REQUEST , $_ENV , $_PHP_SELF , $php_errormsg

PHP- Constants Types

- A constant is a name or an identifier for a simple value
- Constants are Global (used across the entire script)
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).
- By convention, constant identifiers are always uppercase.
- You cannot alter the value of constant after declaration or undefined:

```
define("ROOT_LOCATION", "/usr/local/www/");

$directory = ROOT_LOCATION;
```

PHP- Magic Constants

There are five magical constants returned as each is declared :

__LINE__ : The current line number of the file.

__FILE__ : The full path and filename of the file.

__FUNCTION__ : The function name as it was declared (case-sensitive).

__CLASS__ : The class name as it was declared (case-sensitive).

__METHOD__ : The class method name as it was declared (case-sensitive).

PHP- Operators

- Arithmetic Operators: + , - , * , / , % , ++ , --
- Assignment Operators : = , += , -= , *= , /= , %=
- Comparison Operators: == , === , != , !==, < , > <=, >=
- Logical Operators : and , or , ! , && , | |
- Conditional Operator : $variable = (1 < $x) ? 'value1' : 'value2';

PHP- Conditional Statements

You can use conditional statements in your code to make your decisions by any of the following three decision making statements:

- if...else statement
- elseif statement − is used with the if...else statement to execute a set of code if one of the several condition is true
- switch statement − is used if you want to select one of many blocks of code to be executed. The switch statement is used to avoid long blocks of if..elseif..else code.

PHP- Loops Types

Loops in PHP are used to execute the same block of code a specified number of times. PHP supports following four loop types.

- for − loops through a block of code a specified number of times.
- while − loops through a block of code if and as long as a specified condition is true.
- do...while − loops through a block of code once, and then repeats the loop as long as a special condition is true.
- foreach − loops through a block of code for each element in an array.

PHP- Loops Types

continue and break keywords used to control the loops execution.

- The PHP break keyword is used to terminate the execution of a loop prematurely.
- The PHP continue keyword is used to halt the current iteration of a loop but it does not terminate the loop.