***# What is PHP? List out the popular CMS in PHP. Explain the different between static and dynamic website.***

PHP (Hypertext Preprocessor) is a widely-used open-source scripting language designed for web development. It can be embedded into HTML and is particularly well-suited for server-side scripting. PHP allows developers to create dynamic web pages that interact with databases, handle sessions, and perform various server-side tasks.

**Popular CMS in PHP: WordPress**: The most popular CMS, ideal for blogs and websites. **Drupal**: A flexible and powerful CMS for complex sites and applications. **Joomla**: User-friendly CMS for websites and online applications. **Magento**: A robust platform for e-commerce sites. **Laravel**: While primarily a framework, it’s often used to build custom CMS. **TYPO3**: An enterprise-level CMS with extensive features. **Craft CMS**: A flexible and customizable CMS for developer

|  |  |
| --- | --- |
| **Dynamic website** | **Static website** |
| Content can change based on user interactions or database queries. | Fixed content that does not change unless manually updated. |
| Built using server-side languages (like PHP) and may include databases (like MySQL). | Generally built using HTML, CSS, and sometimes JavaScript. |
| May be slower due to server processing and database retrieval. | Faster load times since they consist of pre-rendered HTML files. |
| Highly interactive; supports features like user accounts, forms, and real-time updates. | Limited interactivity; often lacks features like user logins or database interactions. |
| Ideal for blogs, e-commerce sites, and applications requiring frequent updates. | Best for small sites, portfolios, or informational pages. |

***# Explain different types of loop with its syntax in php.***

**While :** The while loop executes a block of code while a condition is true. **Syntax:** while (condition) { // Code to be executed } **example:** <?php $i = 0; while ($i < 10) { echo $i; $i++; } ?>

**Do while** : The do while statement will always execute the block of code once, it will then check the condition and repeat the loop while the condition is true. Syntax: do { // Code to be executed } while (condition);

Example: $i=0; do { echo $i; $i++; } while ($i<10;

**For**: The for loop is used when the number of iterations is known. Syntax: for (initialization; condition; increment/decrement) { }

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

**Foreach :** The foreach loop is specifically used to iterate over arrays. It simplifies the process of accessing $sql = "INSERT INTO users (username, email) VALUES ('hari', 'hari@example.com')";

if ($conn->query($sql) === TRUE) {

echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error; } array elements. Syntax: foreach ($array as $value) { // Code to be executed }

Example: $fruits = array("Apple", "Banana", "Cherry"); foreach ($fruits as $fruit) { echo $fruit; }

***# What is PHP variables?***

Variable can have short names ( like x and y) or more descriptive names (age, car name, total volume).Rules of PHP variables:

--A variables starts with the $ sign, followed by the name of the variable. – A variable name must begin with a letter or the underscore character . – A variable name can only contain alpha- numeric characters and underscores( A-Z, 0-9 and \_). – A variable name should not contain spaces.

***# Write the code in PHP to connect to a databases. Also , explain about different operation that can be performed in a database with example code.***

<?php $ser = "localhost"; $user = "root"; $pass = ""; $database = "database name"; $conn = mysqli\_connect ($ser, $user, $pass, $database); if (!$conn) { echo "Connection failed "; } else { echo "Connection successfully"; } ?>

**Database Operations**

Once you have established a connection, you can perform various operations on the database:

**1)Create (Insert)**: Inserting data into a table. **Example:** <?php  if(isset($\_POST['submit']))   { $name=$\_POST[‘name’]; $class=$\_POST[‘class’]; $sql = "INSERT INTO users values (‘$name’, ‘$class' );"; if ($conn->query($sql) === TRUE) { echo " data are inserted"; } else { echo "data are not inserted”; } } ?>

**2) Update:** Modifying existing data in a table. **Example:** <?php  if(isset($\_POST['update']))   { $name=$\_POST[‘name’]; $class=$\_POST[‘class’]; $sql = "update users set name=’$name’ where class=’$class’;”; if ($conn->query($sql) === TRUE) { echo " data are updated"; } else { echo "data are not updated”; } } ?>

3) **Delete:** Removing data from a table. **Example:** <?php  if(isset($\_POST['delete']))   { $name=$\_POST[‘name’]; $class=$\_POST[‘class’]; $sql = ”delete from users where name=’$name’;”; if ($conn->query($sql) === TRUE) { echo " data are deleted"; } else { echo "data are not deleted”; } } ?>

***# What are XML?***

XML stands for eXtensible Markup Language. A markup language is used to provide information about a document . Tags are added to the document to provide the extra information. HTML tags tell a browser how to display the document .XML tags give a reader some idea what some of the data means.

**# Advantage of XML**: - Easy data sharing, text documents are readable between any device. – Documents can be modified with any text editor. – Possible to understand the contents of the xml documents just by looking at it with text editor. – Easy to manipulate via programming languages.

***#Explain about the structure of XML.***

. Here is an explanation of its XML structure:

**1. Prolog**

The XML prolog is an optional component that can appear at the beginning of an XML document. It typically includes the XML declaration and may also contain information about the document's character encoding.

**2. Root Element**

Every XML document must have a single root element that contains all the other elements. This root element is also known as the document element.

**3. Elements**

Elements are the building blocks of XML. An element consists of a start tag, content, and an end tag. Elements can contain other elements, attributes, text, and more.

**4. Attributes**

Attributes provide additional information about elements. They are defined within the start tag of an element.

**5. Nested Elements**

XML allows elements to be nested within other elements to create a hierarchical structure.

**6. Comments**

Comments can be added to XML documents to include notes or explanations. They are ignored by XML parsers.

**7. Processing Instructions**

Processing instructions provide directions to applications processing the XML document.

**## write the importance of client side validation.**

The importance of client side validation:

* Improved user Experience
* Reduced server load
* Faster validation
* Enhanced security
* Data Integrity
* Customizable Feedback
* Better Resource Utilization

***#What are PhP framework? Why are they important? Explain its type.***

A framework is a structure that developers choose to build their application. It determines the structure of the application and facilitates it to connect with many different API’s. A proficient **PHP framework** enables developers to develop PHP application faster, efficiently and assist in building stable applications thereby reducing the amount of repetitive coding for PHP programmers.

**Important of PHP framework:**

1. Code Reusability
2. Security
3. Performance
4. Scalability
5. Best Practices
6. Community Support

Types of PHP Frameworks

PHP frameworks can be categorized based on their design philosophy, architecture, and specific use cases. Here are some popular types:

**Full-Stack Frameworks:** These frameworks provide a complete solution for web development, including frontend and backend development tools. **Microframeworks:** Lightweight frameworks designed for small applications or services, providing only essential features without the overhead of full-stack frameworks. **Component-Based Frameworks:** Frameworks that provide a collection of independent components or libraries that can be used together or separately.

**Content Management System (CMS) Frameworks:** Frameworks designed to facilitate the creation and management of web content, often used for building websites and blogs.

***## PHP function:*** We will now explore how to create your own functions. To keep the script from being executed when the page loads, you can put it into a function. A function will be executed by a call to the function. You may a function from anywhere within a page. A function will be executed by a call to the function. Given the function a name that reflect what the function does. The function name can start with a letter or underscore(not a number).

**## Inheritance:** Inheritance is a mechanism in which one class acquires the property of another class. With inheritance, we can reuse the fields and methods of the existing class. Hence, inheritance facilitates Reusability and is an important concept of OOPs.

**Five types of inheritance**: **Single inheritance -** Subclasses inherit characteristics from a single superclass. **Multiple inheritance** - A subclass may have more than one superclass and inherit characteristics from all of them. **Multilevel inheritance -** A subclass may have its own subclasses. In other words, a subclass of a superclass can itself be a superclass to other subclasses. **Hierarchical inheritance** - A base class acts as the parent superclass to multiple levels of subclasses. **Hybrid inheritance** - A combination of one or more of the other inheritance types.

**# Define AJAX . What are the advantage of using AJAX ?**

AJAX is a client side technology, that can help increase the speed and usability of an application web pages by updating only part of the page at a time, rather than requiring the entire page to be reloaded after a user-initiated change. Using ajax, the pages of your application can exchange small amounts of data with the server without going through a form submit. AJAX is a technique for creating fast and dynamic web page.

Advantage of AJAX:

1. Improved User Experience
2. Reduced Bandwidth Usage
3. Asynchronous Processing
4. Seamless Data Retrieval
5. Enhanced Performance

**# What is web server ?**

Specialized software that responds to client represents by providing resources (web pages, documents etc). Based on the client/server model. When user enter URL into web browser, they request specific documents from web server . Maps URL to file on server and returns requested document to client .

**# Explain the concept of dynamic content and architecting a web application**.

**Dynamic content** is what keeps your site fresh & encourage repeated visits from your customers. This type of content will appear different to each website visitor at any given time.

- Gives the business the ability to exploit trending topics

- Tap into the quality function deployment element of Google’s algorithm

- Short

- Time-sensitive

- Easier to share on social media

**Architecting a web application** involves designing its structure, components, and interactions to ensure it meets functional and non-functional requirements such as scalability, performance, security, and maintainability. Here's an overview of the key components and considerations in architecting a web application:

* Client-Side
* Server-side
* Database
* Communication Protocol
* Scalability
* Caching
* Security

**## Write a program in PHP to print the Fibonacci series.**

<?php function fibonacci($n) { $fib = [0, 1]; for ($i = 2; $i < $n; $i++) { $fib[$i] = $fib[$i - 1] + $fib[$i - 2]; } return $fib; } $n = 10; $fibSeries = fibonacci($n); echo "Fibonacci series for the first $n numbers: "; foreach ($fibSeries as $value) { echo $value . " "; } ?> **## PHP Arrays: Definition**: Arrays in PHP are used to store multiple values in a single variable. PHP supports both indexed arrays (numeric indices) and associative arrays (named keys). **Types of Arrays:** i**ndexed Arrays:** Arrays with numeric indices. **Associative Arrays:**  Arrays with named keys. **Multidimensional Arrays:**

**## Explain about the importance of database normalization in web development.**

The importance of database normalization in web development:

**1. Elimination of Redundant Data** Normalization reduces redundancy by ensuring that data is stored only once.

**2. Improved Data Integrity and Consistency** : Normalization enforces data integrity constraints, ensuring that the data is accurate and consistent across the database.

**3. Simplified Database Maintenance** Normalized databases are easier to maintain and modify over time.

**4. Enhanced Query Performance** While normalization can sometimes lead to more complex queries, it can also optimize the database for better performance in other areas.

**5. Modular and Scalable Design**

Normalization helps in designing modular databases where each table represents a specific entity or relationship, making the database more scalable and easier to extend.

**## What are the advantages of using PHP web development**

* The advantage of using php web development: PHP is Open Source. Open Source means you never need to rely on the manufacturer to release the next version if something doesn’t work or pay for expensive upgrades.
* PHP is extendible.
* Large amount of databases are supported.
* PHP is platform independent. It will run on most of the platforms.
* Compatible with servers like IIS and APACHE.
* Low development and maintenance cost with very high performance and reliability

**##Write the use of Alter and Update command in database with example.**

**ALTER Command :** The ALTER command is used to modify the structure of an existing database object, such as a table. This can include adding, deleting, or modifying columns, changing data types, and adding or dropping constraints.

**Examples:**

**Adding a Column**:

ALTER TABLE users ADD COLUMN email VARCHAR(100);

This adds a new column email to the users table.

**Dropping a Column**:

ALTER TABLE users DROP COLUMN email;

This removes the email column from the users table.

**Modifying a Colum:**

ALTER TABLE users MODIFY COLUMN username VARCHAR(100) NOT NULL;

This changes the username column to have a maximum length of 100 characters and sets it to NOT NULL.

**Renaming a Column**

ALTER TABLE users CHANGE COLUMN username user\_name VARCHAR(50);

This renames the username column to user\_name and keeps the data type as VARCHAR(50).

**2. UPDATE Command**

The UPDATE command is used to modify existing data within a table. This can include changing values of specific columns for one or more rows based on a condition.

**Examples:**

**Updating Specific Rows** UPDATE users SET email = 'newemail@example.com' WHERE username = 'hari';

This updates the email column to 'newemail@example.com' for the row where the username is 'hari'.

**Updating All:**

UPDATE users SET status = 'active';

**## Explain different string function used in php with example.**

**1. strlen()** : Returns the length of a string. **Example** $string = "Hello, World!";echo strlen($string); // Output: 13

**2. strtolower()**: Converts a string to lowercase. **Example** $string = "Hello, World!";echo strtolower($string); // Output: hello, world!

**3. strtoupper()** : Converts a string to uppercase. **Example**:$string = "Hello, World!";echo strtoupper($string); // Output: HELLO, WORLD!

**4. ucfirst()** : Converts the first character of a string to uppercase. **Example**: $string = "hello, world!";echo ucfirst($string); // Output: Hello, world!

**5. ucwords()**: Converts the first character of each word in a string to uppercase. **Example**: $string = "hello, world!";echo ucwords($string); // Output: Hello, World!

**## explain about constructors and destructors .**

## 

|  |  |
| --- | --- |
| Constructors | Destructors |
| Accepts one or more arguments. | No arguments are passed. Its void. |
| It has same name as the class. | function name is \_destruct() |
| It has same name as the class. | t has same name as the class with prefix tilda. |
| Constructor is involved automatically when the object is created. | Destructor is involved automatically when the object is destroyed. |
| Constructors can be overloaded. | Destructors cannot be overloaded. |
| Allocates memory. | It deallocates memory. |

**## What is database ?**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a [database management system (DBMS)](https://www.oracle.com/database/what-is-database/#WhatIsDBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

**## Explain about the logical and physical structure of XML.**

**Physical Structure:** **XML Declaration**: The first line in the XML document that defines the XML version and encoding. **Document Type Declaration (DTD):** Specifies the document structure with a list of legal elements and attributes. **Elements and Tags**: The actual data containers within the document **Namespaces**: Used to avoid element name conflicts by qualifying names with a namespace identifier.

**Logical Structure:**  **Elements**: The fundamental building blocks of XML **Attributes**: Provide additional information about elements. Attributes are always in the opening tag of an element. **Text Content**: The actual data within an element.

**## What do you mean by Operators ?**

Operators are symbols or keywords that tell the compiler or interpreter to perform specific operations. These operations can be arithmetic, logical, relational, assignment, or more. Operators are essential in programming as they allow manipulation and comparison of variables and values to perform various tasks.

**### GET and POST GET** is an HTTP method used to request data from a specified resource. When you use GET, the data is appended to the URL as query parameters, making it visible in the URL. It is typically used for retrieving data without causing any side effects on the server. **Key Characteristics:** --Parameters are included in the URL. --Limited amount of data can be sent. --Data is visible in the URL, which can be bookmarked. --Safe and idempotent (does not change the state of the server).

**POST POST** is an HTTP method used to send data to a server to create or update a resource. When you use POST, the data is sent in the body of the request, making it not visible in the URL. It is commonly used for submitting forms or uploading files. **Key Characteristics:** --Parameters are included in the request body.--Can handle a larger amount of data compared to GET.--Data is not visible in the URL.--Not safe and not idempotent (can change the state of the server).

**### Document Type Definition (DTD)**

**Definition:** A Document Type Definition (DTD) defines the structure and the legal elements and attributes of an XML document. It acts as a blueprint for XML documents, specifying which elements can appear and in what order, as well as the attributes and their values that elements can possess. **Key Components:** **Element Declarations:** --Defines the allowed elements and their structure. **Attribute Declarations:** --Specifies the allowed attributes for an element and their types. **Entities:** --Represents a reusable piece of text or binary data. **Advantages: --**Ensures document validity by defining the structure. --Facilitates data sharing and consistency across different systems. --Simplifies the parsing and validation of XML documents. **Disadvantages:** --Less flexible compared to XML Schema (XSD). --Limited data type support (only basic types like CDATA, ID, IDREF).

**### Explain how to integrate html with php ?** When it comes to integrating PHP code with HTML content, you need to enclose the PHPcode with the PHP start tag <?php and the PHP end tag ?>. The code wrapped between thesetwo tags is considered to be PHP code, and thus it'll be executed on the server side before therequested file is sent to the client browser.Let’s have a look at a very simple example, which displays a message using PHP code. Createthe index.php file with the following contents under your document root .**Example** <!DOCTYPE html><html><head><title>How to put PHP in HTML - Simple Example</title></head><body><h1><?php echo "This message is from server side." ?></h1></body></html>

## **Explain HTML 4.0:** HTML 4.0 was the final version of the Hypertext Markup Language (HTML) before the Extensible Markup Language (XHTML) and remains the set of markup on which most large Web sites today are based. Like all HTML levels, HTML 4.0 was the official "recommendation" of the World Wide Web Consortium (W3C), the group that suggests industry standards for the Web. Among new features introduced in HTML 4.0 were: --The cascading style sheet, the ability to control Web page content at multiple levels --The ability to create richer forms -- Support for frames (which is already supported by the major browsers) --Enhancements for tables that make it possible to use captions to provide table content for Braille or speech users .

***### Transaction::*** A transaction is an action or series of actions that are being performed by a single user orapplication program, which reads or updates the contents of the database.A transaction can be defined as a logical unit of work on the database. This may be an entire program, a piece of a program, or a single command (like the SQL commands such as INSERT or UPDATE), and it may engage in number of operations on the database.

|  |  |
| --- | --- |
| **Server-Side Scripting** | **Client-Side Scripting** |
| Works in the back end which could not be visible at the client end | Works at the front end and script are visible among the user |
| Requires server interaction | Dose not need interaction with the server |
| PHP, ASP. Net, ruby on rails, could Fusion, Python. | HTML, CSS, JAVAScript etc. |
| Could effectively customize the web pages and provide dynamic website | Can reduce the load to the server |
| Relatively secure | Insecure |

**## Explain Java Script .** JavaScript is a high-level, interpreted programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. It enables the creation of dynamic and interactive web pages. JavaScript can be used on both the client side (in the browser) and the server side (using environments like Node.js).

**## Uses of javaScript**: --Client-Side Web Development -- Server-Side Development --- Mobile and Desktop Apps --Game Development --Internet of Things **## Advantages of JavaScript** --Speed. Client-side JavaScript is very fast because it can be run immediately within the client-side browser. --Simplicity. JavaScript is relatively simple to learn and implement. --Popularity. JavaScript is used everywhere on the web. --Interoperability. JavaScript plays nicely with other languages and can be used in a huge variety of applications. --Server Load. Being client-side reduces the demand on the website server. **## Disadvantages of JavaScript** --Client-Side Security. Because the code executes on the users’ computer, in some cases It can be exploited for malicious purposes. This is one reason some people choose to disableJavaScript. --Browser Support. JavaScript is sometimes interpreted differently by different browsers.

**## Define XML schema with example.**

XML schemais a language which is used for expressing constraint about XML documents. There are so many schema languages which are used now a days for example Relax- NG and XSD (XML schema definition). An XML schema is used to define the structure of an XML document***. Example:*** <?xml version = "1.0" encoding = "UTF-8"?> <xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"> <xs:element name = "contact"> <xs:complexType> <xs:sequence> <xs:element name = "name" type = "xs:string" /> <xs:element name = "company" type = "xs:string" /> <xs:element name = "phone" type = "xs:int" /> </xs:sequence> </xs:complexType> </xs:element> </xs:schema>

**####**  **SQL short notes:**

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL is a standard language for accessing and manipulating databases. •SQL stands for Structured Query Language •SQL lets you access and manipulate databases •SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

**#### SOAP:** SOAP is an acronym for Simple Object Access Protocol. It is an XML-based messagingprotocol for exchanging information among computers. SOAP is an application of the XMLspecification.--SOAP is an application communication protocol--SOAP is a format for sending and receiving messages--SOAP is platform independent --SOAP is based on XML

|  |  |
| --- | --- |
| XML | HTML |
| eXtensible Markup Language | Hypertext Markup Language |
| XML is dynamic because it is used the trans-port of dat | HTML is static because its main function is in the display of data |
| XML can preserve white space | White spaces are not preserved in HTML |
| Any error in the codes shall not given the final outcome | Small errors in the coding can be ignored and the outcome can be achieved |
| The size of the document may be large | No lengthy documents. Only the syntax needs to be added for best-formatted output |
| The main purpose is to focus on the transport of data and saving the data | Focusses on the appearance of data . Enhances the appearance of text |

**## Why PHP is considered server-side scripting language?** PHP is considered a server-side scripting language for several reasons: ***\*Execution on the Server***: PHP scripts are executed on the server before the HTML is sent to the client’s browser. This means the client only receives the output of the PHP script (usually HTML), not the actual PHP code. *\*****Integration with Web Servers:*** PHP is designed to work closely with web servers such as Apache and Nginx. When a PHP page is requested, the web server processes the PHP code and sends the resulting output to the client. ***\*Dynamic Content Generation:*** PHP can interact with databases, handle forms, and perform various server-side tasks to generate dynamic content. This makes it suitable for building interactive and data-driven websites. \****Access to Server Resources:*** PHP can access and manipulate server resources, such as file systems and databases, enabling it to perform complex operations that client-side languages (like JavaScript) cannot. ***\*Security:*** Since PHP code runs on the server, sensitive information such as database passwords and API keys can be keptsecure, as they are not exposed to the client's browser**.**

**## Why php is consider client-side scripting language**?  **Execution on Client**: Client-side scripts (like JavaScript) run directly in the client's browser after the server sends the webpage.  **Interactivity**: JavaScript can manipulate the DOM (Document Object Model), handle events, validate form inputs on the client side, and create interactive user interfaces without requiring a server round-trip.  **Immediate Feedback**: Client-side scripts can provide immediate feedback to user actions without needing to communicate with the server.

**## Explain about PHP variables with example?**

The scope of a variables is the part of the script where the variable can be referenced/ used. PHP has four different variables scopes: Local, global, static, parameter

**Local variables :**A variables declared within a php function is local and can only be accessed within that function: Example : <?php $a=5; //global scope function myTest() { echo $a; //local scope } myTest(); ?>

**\*Global scope:** A variables that is define outside of any function, has a global scope. Global variables can be accessed from any part of the script, EXCEPT from within a function **.**Example: <?php $x=5; //global scope $y=10; //global scope function myTest() { global $x,$y; } myTest(); echo $y; ?>

**Static scope:** when a function is completed, all of its variables are normally deleted. However, sometimes you want a local variable to not be deleted. Example: <?php function myTest() { static $x=0; echo$x; $x++; } myTest(); myTest(); myTest(); ?>

**## Explain one dimensional and multi-dimensional array.**

**\*Multi-dimensional :** An array can also contain another array as a value, which in turn can hold other arrays as well. In such a way we can create two- or three-dimensional array. Example: <?php // A two-dimensional array : $cars=array ( array(“volvo”,100,80), array (“BMW”,60,59), array (“Toyota”,110,100) ); ?>

\***One-dimensional** array is a simple list of elements, where each element can be accessed by its index. It is essentially a linear collection of data items of the same type. **Characteristics:** **Single Level**: Contains only one row of data. **Indexed**: Elements are accessed using a single index. **Homogeneous**: Usually, all elements are of the same data type.

**## What are conditional statement with suitable example?**

Conditional statements are used to perform different actions based on different conditions. They allow you to execute certain pieces of code depending on whether a particular condition is true or false. Types of Conditional Statements in PHP: 1) if statement 2) if...else statement 3) if...elseif...else statement 4) switch statement 1.**if Statement:** The if statement executes a block of code if a specified condition is true**. Syntax:** if (condition) { // code to be executed if condition is true } **2. if...else Statement :** The if...else statement executes one block of code if a condition is true, and another block of code if the condition is false. **Syntax:** if (condition) { // code to be executed if condition is true } else { // code to be executed if condition is false } **3. if...elseif...else Statement :** The if...elseif...else statement allows you to specify multiple conditions and execute different blocks of code for each condition. **Syntax:** if (condition1) { // code to be executed if condition1 is true } elseif (condition2) { // code to be executedif condition2 is true } else { // code to be executed if none of the conditions are true }