

ADVANCE PHP IMPORTANT QUESTIONS AND ANSWERS

Q1. Short Answer Type Questions.

a) Which are XML Special entities?

Ans:- XML special entities are predefined entities used in XML to represent special characters that have a reserved meaning in XML syntax. Some examples include:

- **&**; for ampersand (&)
- **<**; for less-than symbol (<)
- **>**; for greater-than symbol (>)
- **"**; for double quote (")
- **'**; for single quote (')
- ** **; for non-breaking space

b) What is AJAX?

Ans:- AJAX stands for Asynchronous JavaScript and XML. It is a technique used in web development to create interactive and dynamic web pages. AJAX allows web pages to send and receive data asynchronously with the server without interfering with the display and behavior of the existing page.

c) What is sticky form?

Ans:- A sticky form is a form on a web page where the input values entered by the user are retained (or "stick") even after the form is submitted or the page is refreshed. This is achieved by storing the input values in the browser's local storage or by having the server return the input values back to the form when the page is reloaded.

d) What is setcookie() function?

Ans:- The **setcookie()** function is a PHP function used to set a cookie in the client's browser. Cookies are small pieces of data that are sent from a website and stored in the user's web browser while the user is browsing. Cookies are commonly used for session management, user authentication, and tracking user behavior.

e) Define Template.

Ans:- A template is a pre-designed layout or format used as a starting point for creating documents or web pages. In web development, a template refers to a file containing a mixture of HTML, CSS, and placeholders for dynamic content that can be used to generate multiple web pages with a consistent design and structure.

f) What is Encapsulation?

Ans:- Encapsulation is an object-oriented programming concept that refers to the bundling of data (attributes) and methods (functions) that operate on the data into a single unit, called a class. Encapsulation helps to hide the internal state of an object and only expose the necessary functionality through a well-defined interface.

g) What is the \$_SERVER variable?

Ans:- **\$_SERVER** is a PHP superglobal variable that holds information about the server and the execution environment. It is an associative array containing various server-related information such as server name, script name, request method, and more.

h) Enlist XML elements?

Ans:- XML elements are the building blocks of an XML document. They consist of a start tag, an end tag, and the content between them. Some commonly used XML elements include:

- `<element>`
- `<nested_element>`
- `<element_with_attributes attribute="value">content</element_with_attributes>`

i) What is Content Management system?

Ans:- A Content Management System (CMS) is a software application or set of related programs that are used to create and manage digital content. CMS platforms provide users with the ability to easily create, edit, organize, and publish content on the web without requiring specialized technical skills.

j) What is SOAP?

Ans:- SOAP stands for Simple Object Access Protocol. It is a protocol used for exchanging structured information in the implementation of web services in computer networks. SOAP defines a standardized format for XML-based messages and a set of rules for exchanging those messages between applications over the internet.

k) State the purpose of Extend Keyword.

Ans:- The **extend** keyword in programming, particularly in object-oriented languages like Java or PHP, is used to create a subclass that inherits properties and methods from a parent class (also called a superclass). The purpose of the **extend** keyword is to establish a hierarchical relationship between classes, allowing the subclass to reuse and extend the functionality of the parent class.

l) What is Class?

Ans:- In object-oriented programming, a class is a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that objects of the class will have. Objects are instances of classes, and they encapsulate data and behavior relevant to the concept represented by the class.

m) What is \$_REQUEST variable?

Ans:- `$_REQUEST` is a PHP superglobal variable that is used to collect data submitted through both the GET and POST methods. It is an associative array that contains the values of variables passed to the current script via HTTP request parameters, such as form submissions or URL parameters.

n) What is Serialization?

Ans:- Serialization is the process of converting data structures or object states into a format that can be easily stored, transmitted, or reconstructed later. In programming, serialization is commonly used for tasks like data persistence (e.g., saving objects to a file) or data interchange between different systems.

o) What is Document object Model in PHP?

Ans:- The Document Object Model (DOM) in PHP refers to a set of PHP functions and classes that allow developers to manipulate XML and HTML documents as a hierarchical tree structure. With DOM functions, developers can create, modify, and

delete elements, attributes, and text nodes within XML or HTML documents programmatically.

p) Describe any two content management system software.

Ans:- Two examples of content management system (CMS) software are:

1. WordPress: WordPress is a popular open-source CMS primarily used for creating websites and blogs. It offers a user-friendly interface, a wide range of themes and plugins for customization, and robust community support.
2. Joomla: Joomla is another open-source CMS known for its flexibility and extensibility. It is suitable for building various types of websites, including corporate websites, online magazines, and e-commerce sites. Joomla offers a range of features such as user management, content management, and template customization.

q) State the purpose of Final Keyword?

Ans:- The **final** keyword in programming, particularly in object-oriented languages like Java or PHP, is used to restrict the inheritance of a class or the overriding of a method. When a class or method is marked as **final**, it cannot be subclassed or overridden by any other class or method.

r) What is meaning of Self Processing form?

Ans:- A self-processing form, also known as a self-submitting form, is a web form that processes its own data upon submission. This means that the form action attribute points to the same script or URL that the form is displayed on. When the form is submitted, the data is sent back to the same script for processing.

s) What is AJAX Script?

Ans:- An AJAX script is a JavaScript script that utilizes the AJAX (Asynchronous JavaScript and XML) technique to send and receive data from a web server asynchronously without interfering with the display and behavior of the existing web page. AJAX scripts enhance user experience by allowing dynamic updates to web content without requiring a full page reload.

t) What is the use of abstract class?

Ans:- The use of an abstract class in object-oriented programming is to provide a blueprint for other classes to inherit from. An abstract class cannot be instantiated itself; instead, it serves as a template for its subclasses.

u) Enlist XML element.

Ans :- XML elements are the basic building blocks of an XML document. Some common XML elements include:

- `<element>`
- `<nested_element>`
- `<element_with_attributes attribute="value">content</element_with_attributes>`

v) What are different technologies are used in Ajax?

Ans:- Different technologies used in AJAX (Asynchronous JavaScript and XML) include:

- HTML and CSS for structuring and styling web content.
- JavaScript for implementing asynchronous communication with the server and manipulating the DOM dynamically.

- XMLHttpRequest (XHR) object for making HTTP requests to the server asynchronously.
- JSON (JavaScript Object Notation) for exchanging data between the client and the server, although XML can also be used.
- XML (Extensible Markup Language) for exchanging structured data between the client and the server, although JSON is more commonly used in modern AJAX applications.

w) What is Web Service?

Ans:- A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It provides a standardized way of integrating web-based applications using open standards such as XML, SOAP, WSDL, and others. Web services enable communication and data exchange between different platforms and technologies.

x) Define UDDI.

Ans :- UDDI stands for Universal Description, Discovery, and Integration. It is a directory service used to publish and discover web services. UDDI provides a standardized way for businesses to list their services and for other businesses to find and consume those services over the internet.

y) What is PHP frame work.

Ans:- A PHP framework is a collection of pre-written code and libraries that provides a structured approach to developing web applications in PHP. PHP frameworks offer reusable components, follow design patterns, and provide features like MVC (Model-View-Controller) architecture, database abstraction, form handling, and security mechanisms to streamline the development process and maintain code consistency.

z) Define Template of object oriented.

Ans:- a template refers to a blueprint or a generic design for creating objects. It defines the structure and behavior of objects but does not itself instantiate objects. Templates typically include class declarations with properties and methods, specifying the attributes and behaviors that objects instantiated from the template will have. Templates serve as a foundation for creating objects with consistent characteristics and behaviors.

Q2. Long Answer Type Questions.

1) What is Document Object Model in PHP?

Ans:- The Document Object Model (DOM) in PHP refers to a set of PHP functions and classes that allow developers to manipulate XML and HTML documents as a hierarchical tree structure. With DOM functions, developers can create, modify, and delete elements, attributes, and text nodes within XML or HTML documents programmatically.

This allows for tasks such as:

1. **Accessing Elements:** Developers can access individual elements within the document, retrieve their attributes and content, and traverse the document tree to locate specific elements.

2. **Modifying Content:** DOM allows for the modification of existing content within the document. This includes adding new elements, removing existing ones, updating attributes, and changing text content.
3. **Creating New Documents:** Developers can create entirely new XML or HTML documents from scratch using the DOM classes and methods. This is useful for generating dynamic content or constructing documents based on user input.
4. **Parsing and Serializing:** DOM provides methods for parsing XML or HTML strings into a DOM document and serializing a DOM document back into a string representation. This facilitates the exchange of document data between different systems.

Example:-

```
<?php
$doc = new DOMDocument();
$doc->load('example.xml');
$root = $doc->documentElement;
$children = $root->childNodes;
foreach ($children as $child) {
    echo $child->nodeName . ' : ' . $child->nodeValue . '<br>';
}
?>
```

2) Explain class and object with example.

Ans:- A class is a blueprint for creating objects. It defines the properties (attributes) and behaviors (methods) that objects of the class will have. An object, on the other hand, is an instance of a class. When you instantiate a class, you create an object that has access to all the properties and methods defined within the class.

Example:-

```
<?php
class Car {
    public $brand;
    public $model;
    public function __construct($brand, $model) {
        $this->brand = $brand;
        $this->model = $model;
    }
    public function getInfo() {
        return "This is a {$this->brand} {$this->model}.";
    }
}
$car1 = new Car("Toyota", "Camry");
$car2 = new Car("Honda", "Accord");
echo $car1->getInfo();
echo $car2->getInfo();
```

?>

3) Explain Setting Response Headers.

Ans:- Setting response headers in PHP is a crucial aspect of web development, as it allows you to provide additional information about the HTTP response being sent from the server to the client's browser. These headers can convey various details such as the content type, cache-control directives, HTTP status codes, and more.

The **header()** function in PHP is used to set HTTP headers in the response. It takes a single argument, which is the header string to be sent. The syntax for the **header()** function is:

header(string \$header, bool \$replace = true, int \$http_response_code = null): void

Here's a breakdown of the parameters:

- **\$header:** The header string to be sent.
- **\$replace:** (Optional) A boolean value indicating whether the header should replace a previous similar header. By default, it's set to **true**, which means the header will replace a previous header of the same type if one exists. If set to **false**, the header will be added to any existing headers of the same type.
- **\$http_response_code:** (Optional) The HTTP response code to send. This parameter is used to set the HTTP status code along with the header. If provided, the specified status code will be sent in the response.

Example:-

```
header("Content-Type: application/json");
```

```
header("HTTP/1.1 200 OK");
```

```
header("Cache-Control: no-cache");
```

4) Differentiate between GET & POST Methods.

Ans:-

Feature	GET Method	POST Method
Operation	Used to retrieve information from the server.	Used to send data to the server to create/update a resource.
Data Location	Appends data to the URL, visible to all.	Includes data in the request body, not displayed in the URL.
Idempotency	Idempotent; the same request can be repeated with no further changes.	Non-idempotent; repeating the same request can lead to different results.
Data Size	Limited by the URL length; less data can be sent.	No limitations on data size; suitable for large amounts of data.
Caching	Can be cached.	Not cached by default.
Security	Less secure as data is exposed in the URL.	More secure; data is concealed within the request body.
Use Case	Ideal for searching and retrieving data.	Ideal for transactions and updating data.

5) Explain XML document structure in details

Ans:- The major portion of XML document includes the following:

1. Prolog:

- The XML prolog is an optional component that appears at the beginning of an XML document.
- It provides information about the XML document, such as the XML version and encoding.

- The XML prolog starts with `<?xml` and ends with `?>`.

Example:- `<?xml version="1.0" encoding="UTF-8"?>`

2. XML Body:

- The root element is the top-level element in the XML document.
- All other elements are contained within the root element.
- There can only be one root element in an XML document.

Example: `<rootElementName>`

`<!-- Other elements and content -->`

`</rootElementName>`

3. Document Type Declaration (DTD) or XML Schema (Optional):

- The DTD or XML Schema defines the structure and rules for the XML document.
- It specifies the allowable elements, attributes, and their relationships.
- The DTD is declared within the document type declaration, which appears after the XML prolog.

Example with DTD: `<!DOCTYPE rootElementName SYSTEM "example.dtd">`

Example with XML Schema:

`<rootElementName xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="example.xsd">`

4. Elements:

- Elements are the building blocks of an XML document.
- They represent data or structure within the document.
- Elements consist of start tags, end tags, and content.

Example: `<elementName attribute="value">content</elementName>`

5. Attributes:

- Attributes provide additional information about elements.
- They are specified within the start tag of an element.
- Attributes consist of a name and a value.

Example: `<book ISBN="978-0-306-40615-7">XML for Beginners</book>`

6. XML empty elements:

An empty element is an element that is complete by itself; it never contains other elements. Rather than being composed of a start tag, data, and an end tag, the empty element is a combined start and end tag.

7. Comments:

- Comments are used to add descriptive or explanatory notes within the XML document.
- They are enclosed within `<!--` and `-->`.

Example: `<!-- This is a comment -->`

6) What is introspection? Explain `get_class_methods()` and `get_class_vars()` with suitable example?

- **Ans:-** Introspection refers to the ability of a programming language to examine the type or properties of an object at runtime. In PHP, introspection allows you to inspect classes and objects to retrieve information about their methods and properties.
- `get_class_methods()` is a PHP function used to retrieve an array of method names

defined in a class. Similarly, `get_class_vars()` is used to retrieve an array of property names and their values defined in a class.

Example:

```
class MyClass {
    public $prop1;
    protected $prop2;
    private $prop3;
    public function method1() {}
    protected function method2() {}
    private function method3() {}
}
$methods = get_class_methods('MyClass');
print_r($methods);
$vars = get_class_vars('MyClass');
print_r($vars);
```

7) What is Inheritance? Explain with suitable example.

Ans:- inheritance is a fundamental concept in object-oriented programming (OOP) where a new class (subclass) is created based on an existing class (superclass). The subclass inherits the properties and behaviors (methods) of the superclass and can also have its own additional properties and methods.

Example:

```
class Animal {
    public $name;
    public function sound() {
        echo "Animal sound";
    }
}
class Dog extends Animal {
    public function sound() {
        echo "Woof!";
    }
}
$dog = new Dog();
$dog->sound();
```

8) Explain with example how to connect database using PHP and Ajax.

Ans:- To connect to a database using PHP and Ajax, you typically follow these steps:

- **PHP Side:**

- Create a PHP script that establishes a connection to the database using functions like `mysqli_connect()`.
- Execute SQL queries to fetch or modify data in the database.
- Return the result (e.g., JSON encoded data) to the client-side JavaScript.

- **Ajax Side:**

- Write JavaScript code to make an Ajax request to the PHP script.

- Handle the response from the PHP script and update the webpage dynamically.

Example:

```
<?php
$conn = mysqli_connect("localhost", "username", "password", "database");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT * FROM users";
$result = mysqli_query($conn, $sql);
$data = mysqli_fetch_all($result, MYSQLI_ASSOC);
echo json_encode($data);
mysqli_close($conn);
?>
```

Java Script :-

```
$.ajax({
    url: 'get_users.php',
    type: 'GET',
    success: function(response) {
        console.log(response);
    },
    error: function(xhr, status, error) {
        console.error(error);
    }
});
```

9) Explain mouse & keyboards event in JavaScript.

Ans:- Mouse Events:

1. click:

- Fired when the user clicks on an element.
- This event is often used to trigger actions such as opening links or displaying content.

2. dblclick:

- Fired when the user double-clicks on an element.
- This event is less commonly used but can be used to trigger special actions for double-click events.

3. mouseover:

- Fired when the mouse pointer enters the area of an element.
- This event is often used to trigger actions such as showing tooltips or changing the appearance of elements when hovered over.

4. mouseout:

- Fired when the mouse pointer leaves the area of an element.
- This event is often used to revert changes made by the **mouseover** event.

5. mousedown:

- Fired when the mouse button is pressed down on an element.
- This event is useful for implementing actions that require a mouse click and hold, such as dragging elements.

6. **mouseup:**

- Fired when the mouse button is released after being pressed down.
- This event is often used in conjunction with the **mousedown** event for implementing actions like ending a drag operation.

7. **mousemove:**

- Fired when the mouse pointer is moved over an element.
- This event is commonly used to track the movement of the mouse pointer and implement interactive features like drawing or tracking.

Keyboard Events:

1. **keydown:**

- Fired when a key on the keyboard is pressed down.
- This event is often used to capture user input and trigger actions based on the keys pressed.

2. **keypress:**

- Fired when a key that produces a character value is pressed down.
- This event is similar to **keydown**, but it specifically deals with character input and is not triggered for keys that do not produce character values (e.g., Shift, Ctrl, Arrow keys).

3. **keyup:**

- Fired when a key on the keyboard is released after being pressed down.
- This event is often used to capture the moment when a user releases a key and trigger corresponding actions.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mouse & Keyboard Events</title>
</head>
<body>
  <button id="myButton">Click me</button>
  <input type="text" id="myInput">
  <script>
    document.getElementById("myButton").addEventListener("click", function() {
      console.log("Button clicked!");
    });
    document.getElementById("myInput").addEventListener("keydown", function(event)
{
      console.log("Key pressed: " + event.key);
    });
  </script>
```

</body>

</html>

10) Explain features of Joomla/Drupal.

Ans:- Features of Joomla/Drupal:

- **Joomla:**

- **User Management:** Joomla offers robust user management features, including user authentication, access control levels, and user registration.
- **Content Management:** It provides a powerful content management system (CMS) with features like article management, categories, tags, and media management.
- **Extensibility:** Joomla supports a vast ecosystem of extensions, including templates, modules, plugins, and components, allowing users to extend the functionality of their websites.
- **Multilingual Support:** Joomla offers built-in multilingual support, allowing users to create websites in multiple languages.

- **Drupal:**

- **Scalability:** It is highly scalable and suitable for building small personal websites to large enterprise-level applications.
- **Extensibility:** Similar to Joomla, Drupal has a vast ecosystem of modules and themes that extend its functionality and allow users to add new features to their websites.
- **Security:** Drupal is known for its strong security features, including regular security updates, access controls, and user permissions.
- **Multisite Capabilities:** Drupal supports multisite configurations, enabling users to manage multiple websites from a single Drupal installation.

11) What is SOAP? Explain in detail.

Ans:- SOAP stands for Simple Object Access Protocol. It is a protocol used for exchanging structured information in the implementation of web services in computer networks. SOAP defines a standardized format for XML-based messages and a set of rules for exchanging those messages between applications over the internet.

- **Message Structure:** SOAP messages are XML-based and consist of an envelope, header, body, and optional fault element. The envelope element encapsulates the entire SOAP message, while the header contains optional metadata, and the body contains the actual data being exchanged.
- **Transport Independence:** SOAP messages can be transported over various protocols, including HTTP, SMTP, and more. However, it is most commonly used over HTTP.
- **Platform Independence:** SOAP is platform-independent, meaning it can be used with any programming language or platform that supports XML.
- **Interoperability:** SOAP promotes interoperability between different systems by providing a standardized way of exchanging messages. It allows systems developed in different programming languages and running on different platforms to communicate with each other seamlessly.
- **WSDL (Web Services Description Language):** WSDL is an XML-based language used to describe the functionality offered by a web service. It defines the

operations, messages, and bindings supported by the web service, allowing clients to understand how to interact with the service.

- **UDDI (Universal Description, Discovery, and Integration):** UDDI is a directory service that allows businesses to publish and discover web services. It provides a centralized registry where service providers can publish their services, and clients can search for and consume those services.

12) Explain XML MVC framework.

Ans:- An XML MVC (Model-View-Controller) framework is a software architectural pattern used for designing and implementing web applications. In an XML MVC framework, the application's components are divided into three interconnected parts:

- **Model:** Represents the application's data and business logic. It interacts with the database or other data sources to fetch and manipulate data.
- **View:** Represents the presentation layer of the application. It is responsible for displaying the data to the user in a user-friendly format. Views are typically implemented using XML templates that are transformed into HTML or other markup languages.
- **Controller:** Acts as an intermediary between the model and the view. It receives user input, processes it, and updates the model accordingly. Controllers handle user requests, invoke appropriate actions on the model, and determine which view to render.

13) How to create object in PHP? Explain with example.

Ans:- objects are instances of classes. You can create objects using the **new** keyword followed by the class name, optionally passing arguments to the class constructor if it's defined. Here's an example:

```
<?php
class MyClass {
    public $prop;
    public function __construct($value) {
        $this->prop = $value;
    }
}
```

```
$obj = new MyClass("Hello");
echo $obj->prop;
?>
```

14) Write a simple PHP program which implements AJAX for addition of two numbers.

Ans:-

```
<!DOCTYPE html>
<html>
<head>
    <title>AJAX Addition</title>
    <script>
        function addNumbers() {
```

```

var num1 = document.getElementById("num1").value;
var num2 = document.getElementById("num2").value;
var xhr = new XMLHttpRequest();
xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
        document.getElementById("result").innerHTML = xhr.responseText;
    }
};
xhr.open("GET", "addition.php?num1=" + num1 + "&num2=" + num2, true);
xhr.send();
}
</script>
</head>
<body>
    <h2>AJAX Addition</h2>
    Number 1: <input type="text" id="num1"><br>
    Number 2: <input type="text" id="num2"><br>
    <button onclick="addNumbers()">Add</button><br>
    Result: <span id="result"></span>
</body>
</html>

```

15) Explain the structure of WSDL.

Ans:- WSDL is an XML-based language used to describe the functionality offered by a web service. It defines the operations, messages, and bindings supported by the web service, allowing clients to understand how to interact with the service. The structure of a WSDL document typically includes the following components:

- **<definitions>**: The root element of the WSDL document, containing all other elements.
- **<types>**: Defines the data types used in the messages exchanged by the web service. It typically contains XML Schema definitions (XSD) for defining complex data structures.
- **<message>**: Defines the abstract data types used in the web service operations. It specifies the format and structure of input and output messages for each operation.
- **<portType>**: Defines the set of operations supported by the web service and the messages involved in each operation. It describes the abstract interface of the web service.
- **<binding>**: Specifies the protocol and message format used for communication with the web service. It binds the abstract operations defined in the portType to concrete message formats and protocols.
- **<service>**: Specifies the network address (URL) where the web service is available and the binding used for communication.

16) Explain XML Parser.

Ans:- An XML parser is a software component or library that processes XML documents and extracts information from them according to the document's structure and rules. XML parsers are used in applications to read, validate, and manipulate XML data. There are two main types of XML parsers:

- **DOM (Document Object Model) Parser:** Parses an entire XML document and creates a tree-like in-memory representation of the document, allowing for random access to the document structure. DOM parsers are useful for applications that need to traverse and manipulate XML documents extensively.
- **SAX (Simple API for XML) Parser:** Parses an XML document sequentially and generates events (callbacks) as it encounters elements, attributes, and other XML constructs. SAX parsers are more memory-efficient than DOM parsers and are suitable for processing large XML documents or streams where random access to the document structure is not required.

Both DOM and SAX parsers are widely used in various programming languages and platforms, including Java, Python, PHP, and .NET, to handle XML data in applications such as web services, data interchange, configuration files, and more.

17) What are the advantages of AJAX?

Ans:- Advantages of AJAX (Asynchronous JavaScript and XML):

- **Improved User Experience:** AJAX enables web applications to update parts of a webpage dynamically without requiring a full page reload. This results in faster response times and a smoother user experience.
- **Reduced Server Load:** By exchanging data with the server asynchronously, AJAX requests can be made without reloading the entire page. This reduces server load and bandwidth usage, leading to improved performance and scalability.
- **Interactive and Responsive Interfaces:** AJAX enables the development of interactive and responsive user interfaces that can update content dynamically based on user actions, such as form submissions, without interrupting the user's workflow.
- **Enhanced Performance:** AJAX allows for incremental updates to web pages, fetching only the necessary data from the server and updating specific parts of the page. This can lead to significant performance improvements compared to traditional synchronous page loads.
- **Offline Capability:** AJAX techniques, combined with client-side storage mechanisms like localStorage or IndexedDB, can enable web applications to work offline or in low-connectivity environments by caching data locally and synchronizing with the server when connectivity is restored.

18) How articles are created in Drupal R. Joomla?

Ans:- 1. Creating Articles in Drupal:

- In Drupal, content items are typically referred to as "nodes."
- To create a new article (node) in Drupal, you would typically follow these steps:
 - Log in to the Drupal administration interface.
 - Navigate to the "Content" or "Add content" section, depending on the Drupal version and configuration.
 - Click on the "Add content" button.

- Choose the content type for the article (e.g., "Article").
- Fill in the required fields and any additional fields as needed, such as the title, body text, tags, categories, and metadata.
- Optionally, configure settings such as publishing options, URL alias, revisions, and permissions.
- Click the "Save" or "Publish" button to create and publish the article.

2. Creating Articles in Joomla:

- In Joomla, content items are typically referred to as "articles."
- To create a new article in Joomla, you would typically follow these steps:
 - Log in to the Joomla administration interface.
 - Navigate to the "Content" or "Articles" section.
 - Click on the "New" button to create a new article.
 - Fill in the required fields, such as the title and body text.
 - Optionally, configure settings such as publishing options, categories, tags, metadata, and permissions.
 - Click the "Save & Close" or "Save" button to create and publish the article.

Q3. Programs

1) Write a PHP script for the following : Design a form to accept a number from the user. To find sum of the digits of that number. (Use the concept of self - processing page).

Ans:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Sum of Digits</title>
</head>
<body>
  <h2>Enter a Number</h2>
  <form method="post">
    <label for="number">Enter Number:</label>
    <input type="number" name="number" id="number" required>
    <button type="submit" name="submit">Submit</button>
  </form>
  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $number = $_POST["number"];
    $sum = 0;
    $temp = $number;
    while ($temp != 0) {
      $digit = $temp % 10;
```

```

        $sum += $digit;
        $temp = (int)($temp / 10);
    }
    echo "<p>Sum of Digits of $number is $sum</p>";
}
?>
</body>
</html>

```

2) Write a PHP Script to display Server information in table format (use \$_SERVER).

Ans:-

```

<!DOCTYPE html>
<html>
<head>
    <title>Server Information</title>
    <style>
        table {
            border-collapse: collapse;
        }
        table, th, td {
            border: 1px solid black;
            padding: 5px;
        }
    </style>
</head>
<body>
    <h2>Server Information</h2>
    <table>
        <tr>
            <th>Variable</th>
            <th>Value</th>
        </tr>
        <?php
        foreach ($_SERVER as $key => $value) {
            echo "<tr><td>$key</td><td>$value</td></tr>";
        }
        ?>
    </table>
</body>
</html>

```

3) Design a web page to accept student registration details and display it in the next page (use sticky form concept)

Ans:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Registration</title>
</head>
<body>
  <h2>Student Registration</h2>
  <form method="post" action="display.php">
    <label for="name">Name:</label>
    <input type="text" name="name" id="name" required><br>
    <label for="email">Email:</label>
    <input type="email" name="email" id="email" required><br>
    <label for="course">Course:</label>
    <input type="text" name="course" id="course" required><br>
    <button type="submit" name="submit">Register</button>
  </form>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Details</title>
</head>
<body>
  <h2>Student Details</h2>
  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<p>Name: {$_POST['name']}</p>";
    echo "<p>Email: {$_POST['email']}</p>";
    echo "<p>Course: {$_POST['course']}</p>";
  }
  ?>
</body>
</html>
```

4) Write a PHP program which implements Ajax for addition of Two number.

Ans:-

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>AJAX Addition</title>
<script>
    function addNumbers() {
        var num1 = document.getElementById("num1").value;
        var num2 = document.getElementById("num2").value;

        // Create XMLHttpRequest object
        var xhr = new XMLHttpRequest();

        // Define function to handle response
        xhr.onreadystatechange = function() {
            if (xhr.readyState == 4 && xhr.status == 200) {
                document.getElementById("result").innerHTML = xhr.responseText;
            }
        };

        // Open and send AJAX request
        xhr.open("GET", "addition.php?num1=" + num1 + "&num2=" + num2, true);
        xhr.send();
    }
</script>
</head>
<body>
    <h2>AJAX Addition</h2>
    Number 1: <input type="number" id="num1"><br>
    Number 2: <input type="number" id="num2"><br>
    <button onclick="addNumbers()">Add</button><br>
    Result: <span id="result"></span>
</body>
</html>

```

Php script:

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $num1 = $_GET["num1"];
    $num2 = $_GET["num2"];
    $result = $num1 + $num2;
    echo $result;
}
?>

```

5) Write a PHP script for the following :Design a form to accept a number from the user, check whether it is palindrome or not?(Use the concept of self - processing

page).

Ans:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Palindrome Check</title>
</head>
<body>
  <h2>Check Palindrome</h2>
  <form method="post">
    <label for="number">Enter Number:</label>
    <input type="number" name="number" id="number" required>
    <button type="submit" name="submit">Submit</button>
  </form>

  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $number = $_POST["number"];
    $reverse = strrev($number);
    if ($number == $reverse) {
      echo "<p>$number is a palindrome</p>";
    } else {
      echo "<p>$number is not a palindrome</p>";
    }
  }
  ?>
</body>
</html>
```

6) Create a XML file which gives details of books available in “Bookstore”

From following categories

i) Computer

ii) Cooking

iii) YOGA

Ans:-

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <category name="Computer">
    <book>
      <title>Learning Python</title>
      <author>Mark Lutz</author>
      <price>25.00</price>
```

```

</book>
<book>
  <title>Head First Java</title>
  <author>Kathy Sierra</author>
  <price>30.00</price>
</book>
</category>
<category name="Cooking">
  <book>
    <title>The Joy of Cooking</title>
    <author>Irma S. Rombauer</author>
    <price>20.00</price>
  </book>
  <book>
    <title>Mastering the Art of French Cooking</title>
    <author>Julia Child</author>
    <price>35.00</price>
  </book>
</category>
<category name="Yoga">
  <book>
    <title>The Heart of Yoga</title>
    <author>T.K.V. Desikachar</author>
    <price>18.00</price>
  </book>
  <book>
    <title>Light on Yoga</title>
    <author>B.K.S. Iyengar</author>
    <price>22.00</price>
  </book>
</category>
</bookstore>

```

7) Create a form to accept Customers Details and Display it on Next Page.

Ans:-

```

<!DOCTYPE html>
<html>
<head>
  <title>Customer Details Form</title>
</head>
<body>
  <h2>Customer Details Form</h2>

```

```

<form method="post" action="display_customer_details.php">
  <label for="name">Name:</label>
  <input type="text" name="name" id="name" required><br>
  <label for="email">Email:</label>
  <input type="email" name="email" id="email" required><br>
  <label for="phone">Phone:</label>
  <input type="text" name="phone" id="phone" required><br>
  <button type="submit" name="submit">Submit</button>
</form>
</body>
</html>

```

PHP Script:

```

<!DOCTYPE html>
<html>
<head>
  <title>Customer Details</title>
</head>
<body>
  <h2>Customer Details</h2>
  <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      echo "<p>Name: {$_POST['name']}</p>";
      echo "<p>Email: {$_POST['email']}</p>";
      echo "<p>Phone: {$_POST['phone']}</p>";
    }
  ?>
</body>
</html>

```

8) Write a PHP script to Design a form to accept a number from the user to check whether number is palindrome or not. (Use the concept of self processing page).

Ans:-

```

<!DOCTYPE html>
<html>
<head>
  <title>Palindrome Check</title>
</head>
<body>
  <h2>Palindrome Check</h2>
  <form method="post">
    <label for="number">Enter Number:</label>
    <input type="number" name="number" id="number" required>

```

```

        <button type="submit" name="submit">Submit</button>
    </form>
    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $number = $_POST["number"];
        $reverse = strrev($number);
        if ($number == $reverse) {
            echo "<p>$number is a palindrome</p>";
        } else {
            echo "<p>$number is not a palindrome</p>";
        }
    }
    ?>
</body>
</html>

```

9) Write XML script to print the names of the students present in "Student.xml" file.

Ans:-

```

<?xml version="1.0" encoding="UTF-8"?>
<students>
    <student>
        <name>John Doe</name>
    </student>
    <student>
        <name>Jane Smith</name>
    </student>
    <!-- Add more student entries as needed -->
</students>

```

10) Define a class Employee having private member id, name, salary, dept. Define parametrised constructor. Create object and display details of employee having maximum salary.

Ans:-

```

<?php
class Employee {
    private $id;
    private $name;
    private $salary;
    private $dept;
    public function __construct($id, $name, $salary, $dept) {
        $this->id = $id;
        $this->name = $name;
        $this->salary = $salary;
    }
}

```

```

        $this->dept = $dept;
    }
    public function getName() {
        return $this->name;
    }
    public function getSalary() {
        return $this->salary;
    }
    public function displayDetails() {
        echo "ID: $this->id<br>";
        echo "Name: $this->name<br>";
        echo "Salary: $this->salary<br>";
        echo "Department: $this->dept<br>";
    }
}
$Employee1 = new Employee(1, "John", 50000, "IT");
$Employee2 = new Employee(2, "Jane", 60000, "HR");
$Employee3 = new Employee(3, "Smith", 55000, "Finance");
$max_salary_employee = $Employee1;
if ($Employee2->getSalary() > $max_salary_employee->getSalary()) {
    $max_salary_employee = $Employee2;
}
if ($Employee3->getSalary() > $max_salary_employee->getSalary()) {
    $max_salary_employee = $Employee3;
}
echo "Employee with Maximum Salary:<br>";
$max_salary_employee->displayDetails();
?>

```

11) Create an abstract Class shape with method area () and volume (). Derive two classes rectangle (length, breath), Circle (radius). Calculate area and volume of all (Use Method Overriding).

Ans:-

```

<?php
abstract class Shape {
    abstract public function area();
    abstract public function volume();
}
class Rectangle extends Shape {
    private $length;
    private $breadth;
    public function __construct($length, $breadth) {
        $this->length = $length;
        $this->breadth = $breadth;
    }
    public function area() {

```

```

        return $this->length * $this->breadth;
    }
    public function volume() {
        return null; // Rectangles do not have volume
    }
}
class Circle extends Shape {
    private $radius;
    public function __construct($radius) {
        $this->radius = $radius;
    }
    public function area() {
        return pi() * $this->radius * $this->radius;
    }
    public function volume() {
        return null; // Circles do not have volume
    }
}
$rectangle = new Rectangle(5, 4);
$circle = new Circle(3);
echo "Area of Rectangle: " . $rectangle->area() . "<br>";
echo "Area of Circle: " . $circle->area();
?>

```

12) Write script to solve following questions (Use "Student.XML" file)

i) Create a DOM Document Object and load thisXML file

ii) Get the output of this Document to the browser.

Write a script to print the names of the student Present in "Student.XML" file.

Ans:-

```

<?php
$dom = new DOMDocument();
$dom->load("Student.xml");
$students = $dom->getElementsByTagName("name");
foreach ($students as $student) {
    echo $student->nodeValue . "<br>";
}
?>

```

13) Write a PHP Script for the following : Design a form to accept a number from the user. To find Sum of the digits of the number (Use the concept of self processing page).

Ans:-

```

<!DOCTYPE html>
<html>
<head>
    <title>Sum of Digits</title>
</head>
<body>
    <h2>Enter a Number</h2>
    <form method="post">
        <label for="number">Enter Number:</label>
        <input type="number" name="number" id="number" required>
    </form>

```



```

        <button type="submit" name="submit">Submit</button>
    </form>
    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $number = $_POST["number"];
        $sum = 0;
        $temp = $number;

        while ($temp != 0) {
            $digit = $temp % 10;
            $sum += $digit;
            $temp = (int)($temp / 10);
        }
        echo "<p>Sum of Digits of $number is $sum</p>";
    }
    ?>
</body>
</html>

```

14) Define class Employee having private members id, name department, Salary. Define parameterized constructor. Create a Subclass called "Manager" with private member bonus. Create 6 objects of the Manager class and display the details of the Manager having the maximum total salary. (Salary + bonus).

Ans:-

```

<?php
class Employee {
    private $id;
    private $name;
    private $department;
    private $salary;
    public function __construct($id, $name, $department, $salary) {
        $this->id = $id;
        $this->name = $name;
        $this->department = $department;
        $this->salary = $salary;
    }
    public function getSalary() {
        return $this->salary;
    }
    public function displayDetails() {
        echo "ID: $this->id<br>";
        echo "Name: $this->name<br>";
    }
}

```

```

        echo "Department: $this->department<br>";
        echo "Salary: $this->salary<br>";
    }
}
class Manager extends Employee {
    private $bonus;
    public function __construct($id, $name, $department, $salary, $bonus) {
        parent::__construct($id, $name, $department, $salary);
        $this->bonus = $bonus;
    }
    public function getTotalSalary() {
        return $this->getSalary() + $this->bonus;
    }
    public function displayDetails() {
        parent::displayDetails();
        echo "Bonus: $this->bonus<br>";
        echo "Total Salary: " . $this->getTotalSalary() . "<br>";
    }
}
$managers = array(
    new Manager(1, "John", "IT", 50000, 10000),
    new Manager(2, "Jane", "HR", 60000, 12000),
    new Manager(3, "Smith", "Finance", 55000, 11000),
    new Manager(4, "David", "Marketing", 52000, 10500),
    new Manager(5, "Emma", "Operations", 58000, 12500),
    new Manager(6, "Michael", "Production", 54000, 11500)
);
$max_salary_manager = $managers[0];
foreach ($managers as $manager) {
    if ($manager->getTotalSalary() > $max_salary_manager->getTotalSalary()) {
        $max_salary_manager = $manager;
    }
}
echo "<h2>Manager with Maximum Total Salary</h2>";
$max_salary_manager->displayDetails();
?>

```

Q4. Short Notes:

1) Self Processing form.

Ans:- A self-processing form, also known as a self-submitting form, refers to an HTML form that submits data to the same page or script that generated the form. When the form is

submitted, the server-side script processes the form data and typically returns a response to the same page. Self-processing forms are commonly used for simple form submissions, such as search forms or data entry forms.

2) Constructor/Destructor.

- **Ans:- Constructor:** A constructor is a special method in object-oriented programming languages, such as PHP, used for initializing objects of a class. It is called automatically when an object is created and typically initializes the object's properties or performs other setup tasks.
- **Destructor:** A destructor is a special method in object-oriented programming languages that is called automatically when an object is destroyed or goes out of scope. It is used for performing cleanup tasks, such as releasing resources or closing connections, before the object is removed from memory.

3) Serialization.

Ans:- Serialization is the process of converting data structures or objects into a format that can be easily stored, transmitted, or reconstructed later. In PHP, serialization is commonly used for storing complex data structures in files or databases, or for transmitting data over the network. The serialized data can be stored as a string and later deserialized to reconstruct the original data structure.

4) Web services communication models.

Ans:- Web services communication models refer to different protocols and architectures used for communication between web services and clients. Common communication models include:

- **SOAP (Simple Object Access Protocol):** A protocol for exchanging structured information between web services using XML-based messages over various transport protocols, such as HTTP, SMTP, or TCP.
- **REST (Representational State Transfer):** An architectural style for designing networked applications based on a stateless client-server communication model. RESTful web services typically use standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources identified by URIs.

5) Sticky Forms.

Ans :- Sticky forms refer to web forms that retain the data previously entered by the user even after submission or page reload. This is achieved by prefilling the form fields with the data submitted by the user or stored in session or cookies. Sticky forms improve user experience by allowing users to correct errors or modify data without having to re-enter everything from scratch.

6) Encapsulation.

Ans :- Encapsulation is a fundamental concept in object-oriented programming that refers to the bundling of data (attributes or properties) and methods (functions or procedures) that operate on that data into a single unit called a class. Encapsulation hides the internal state of an object and restricts access to it, allowing interactions with the object only through well-defined interfaces (methods). This helps in achieving data abstraction, information hiding, and modular design.

7) WSDL.

Ans :- WSDL is an XML-based language used to describe the functionality offered by a web service. It defines the operations, messages, and bindings supported by the web service, allowing clients to understand how to interact with the service. WSDL documents typically contain detailed information about the web service's endpoints, message formats, transport protocols, and service operations.

8) XML parser.

Ans:- An XML parser is a software component or library that processes XML documents and extracts information from them according to the document's structure and rules. XML parsers are used in applications to read, validate, and manipulate XML data. There are two main types of XML parsers: DOM (Document Object Model) parsers, which create a tree-like in-memory representation of the document, and SAX (Simple API for XML) parsers, which parse the document sequentially and generate events as they encounter elements.

9) XMLHttpRequest object.

Ans:- The XMLHttpRequest object, commonly referred to as XHR, is a JavaScript API that allows client-side scripts to make HTTP requests to a server without reloading the entire page. XHR is commonly used in AJAX (Asynchronous JavaScript and XML) web development to fetch data from a server asynchronously and update parts of a web page dynamically without interrupting the user's workflow. It supports various HTTP methods, such as GET, POST, PUT, DELETE, and can handle different types of data, including XML, JSON, or plain text.