

# PHP QUESTION PAPER SOLVED

## Q.1 What is PHP? Discuss its benefits. full content solution in hindi

### PHP क्या है?

PHP एक सर्वर-साइड स्क्रिप्टिंग लैंग्वेज (Server-Side Scripting Language) है, जिसका पूरा नाम है:

**PHP: Hypertext Preprocessor**

PHP का उपयोग मुख्य रूप से वेब डेवलपमेंट (Web Development) के लिए किया जाता है। यह HTML के साथ मिलकर काम करता है और वेबसाइट को डायनामिक (Dynamic) बनाता है।

जब कोई यूजर वेबसाइट पर कोई रिक्वेस्ट भेजता है, तो PHP उस रिक्वेस्ट को प्रोसेस करता है और सर्वर से आवश्यक डेटा लाकर यूजर को HTML के रूप में जवाब देता है।

### PHP का उपयोग कहाँ होता है?

- वेबसाइट और वेब एप्लिकेशन बनाना
- कंटेंट मैनेजमेंट सिस्टम (जैसे WordPress)
- ई-कॉमर्स वेबसाइट्स (जैसे Magento, OpenCart)
- RESTful APIs बनाना
- डाटाबेस से इंटरैक्शन (MySQL, PostgreSQL आदि)

### PHP के लाभ (Benefits of PHP in Hindi)

#### 1. ओपन-सोर्स और फ्री (Open Source & Free)

PHP पूरी तरह से फ्री है। इसे डाउनलोड करके कोई भी उपयोग कर सकता है।

#### 2. सर्वर-साइड स्क्रिप्टिंग

यह सर्वर पर चलता है, जिससे वेबसाइट अधिक सिक्योर और डायनामिक बनती है।

### 3. HTML के साथ आसानी से इंटीग्रेट होता है

PHP को HTML के साथ बहुत आसानी से जोड़ा जा सकता है।

```
<!-- एक सिंपल PHP कोड HTML में: -->
<html>
<body>
  <h1>नमस्ते!</h1>
  <?php echo "यह PHP से जनरेट हुआ है"; ?>
</body>
</html>
```

### 4. डाटाबेस सपोर्ट

PHP कई डाटाबेस को सपोर्ट करता है जैसे:

- MySQL
- PostgreSQL
- SQLite
- MongoDB (थर्ड-पार्टी लाइब्रेरी के ज़रिए)

### 5. क्रॉस-प्लेटफ़ॉर्म

PHP Windows, Linux, Mac आदि सभी प्लेटफ़ॉर्म पर चलता है।

### 6. CMS और Framework सपोर्ट

PHP के ऊपर कई पॉपुलर CMS और फ्रेमवर्क बनाए गए हैं:

- WordPress (CMS)
- Laravel (Framework)
- CodeIgniter, Symfony आदि

### 7. सुरक्षा (Security)

अगर सही तरह से लिखा जाए, तो PHP स्क्रिप्टिंग बहुत सिक्योर होती है। PHP में इनबिल्ट फ़ंक्शन्स होते हैं जैसे:

- htmlspecialchars()
- mysqli\_real\_escape\_string()

जो सुरक्षा बढ़ाते हैं।

### 8. बड़ी कम्युनिटी और सपोर्ट

PHP एक पुरानी और बहुत पॉपुलर भाषा है, जिससे इसके लिए बड़ी मात्रा में डॉक्युमेंटेशन और ऑनलाइन हेल्प उपलब्ध है।

## PHP कैसे काम करता है?

1. यूज़र ब्राउज़र से रिक्वेस्ट भेजता है (जैसे - [www.example.com](http://www.example.com))

2. वेब सर्वर (Apache/Nginx) PHP स्क्रिप्ट को एक्सेक्यूट करता है
3. PHP स्क्रिप्ट डेटा प्रोसेस करता है (डाटाबेस से डेटा ला सकता है)
4. परिणाम HTML के रूप में यूज़र को वापस भेजा जाता है

## Q.2 What are features of PHP full content solution in hindi

### PHP की विशेषताएँ (Features of PHP in Hindi)

#### 1. ओपन-सोर्स (Open Source)

PHP एक **मुफ्त और ओपन-सोर्स** भाषा है। कोई भी इसे डाउनलोड करके उपयोग कर सकता है और इसमें सुधार कर सकता है।

◆ वेबसाइट: <https://www.php.net> से PHP डाउनलोड किया जा सकता है।

#### 2. सरल और उपयोग में आसान (Simple and Easy to Use)

PHP का सिंटैक्स (Syntax) बहुत ही आसान होता है। इसे सीखना और समझना आसान है, खासकर अगर आपको C या Java जैसी भाषाओं की जानकारी है।

```
<?php
echo "नमस्ते PHP!";
?>
```

#### 3. सर्वर-साइड स्क्रिप्टिंग लैंग्वेज (Server-Side Scripting)

PHP एक सर्वर-साइड लैंग्वेज है। यानी PHP कोड **सर्वर पर एक्सेक्यूट** होता है और यूज़र को केवल HTML आउटपुट मिलता है। इससे वेबसाइट अधिक **सिक्क्योर और डायनामिक** बनती है।

#### 4. प्लेटफॉर्म इंडिपेंडेंट (Platform Independent)

PHP को Windows, Linux, macOS जैसे किसी भी ऑपरेटिंग सिस्टम पर चलाया जा सकता है। यानी यह **क्रॉस-प्लेटफॉर्म** सपोर्ट करता है।

#### 5. HTML के साथ आसानी से जुड़ता है (Easily Integrates with HTML)

PHP को HTML के अंदर बहुत आसानी से लिखा और उपयोग किया जा सकता है।

```
<html>
<body>
  <h1>Welcome!</h1>
  <?php echo "यह PHP कोड है।"; ?>
</body>
</html>
```

## 6. डाटाबेस सपोर्ट (Database Support)

PHP बहुत से डाटाबेस को सपोर्ट करता है:

- MySQL (सबसे लोकप्रिय)
- PostgreSQL
- Oracle
- SQLite
- MongoDB (थर्ड-पार्टी के ज़रिए)

## 7. फास्ट परफॉर्मेंस (Fast Performance)

PHP कोड को एक्सेक्यूट करना तेज़ होता है, खासकर जब इसे Apache/Nginx सर्वर पर कैशिंग तकनीकों के साथ उपयोग किया जाता है।

## 8. ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग (OOP Support)

PHP 5 और उससे आगे की वर्ज़न में **OOP concepts** को सपोर्ट किया गया है:

- Classes
- Objects
- Inheritance
- Interfaces

```
class Car {  
    public $brand;  
    function setBrand($name) {  
        $this->brand = $name;  
    }  
}
```

## 9. सुरक्षा फीचर्स (Security Features)

PHP में कई इनबिल्ट सिक्योरिटी फ़ीचर्स होते हैं जैसे:

- htmlspecialchars() – XSS अटैक से बचाव
- password\_hash() – पासवर्ड को सुरक्षित रूप से स्टोर करने के लिए
- filter\_input() – यूज़र इनपुट को फ़िल्टर करने के लिए

## 10. बड़ी कम्युनिटी और सपोर्ट (Large Community & Support)

PHP की बहुत बड़ी यूज़र कम्युनिटी है। Stack Overflow, GitHub, PHP.net जैसी जगहों पर बहुत सारे संसाधन उपलब्ध हैं।

## 11. Frameworks का सपोर्ट (Framework Support)

PHP के लिए बहुत से पॉपुलर फ्रेमवर्क उपलब्ध हैं:

- Laravel

- CodeIgniter
- Symfony
- Zend
- CakePHP

ये फ्रेमवर्क डेवलपमेंट को तेज़, सुरक्षित और स्ट्रक्चर्ड बनाते हैं।

## 12. कंटेंट मैनेजमेंट सिस्टम (CMS) सपोर्ट

PHP के ऊपर बहुत से CMS बने हैं, जैसे:

- WordPress
- Joomla
- Drupal

इनका उपयोग वेबसाइट बनाने में किया जाता है बिना ज़्यादा कोडिंग के।

## 13. AJAX और JavaScript के साथ इंटीग्रेशन (AJAX & JS Integration)

PHP को JavaScript, jQuery और AJAX के साथ आसानी से जोड़ा जा सकता है, जिससे वेबसाइट और भी इंटरैक्टिव बन जाती है।

## 14. Error Reporting के लिए इनबिल्ट फंक्शन (Built-in Error Reporting)

PHP में error detection के लिए कई प्रकार के error reporting mechanisms होते हैं जैसे:

```
error_reporting(E_ALL);
ini_set("display_errors", 1);
```

## Q.3 How will you write & Execute PHP script? full content solution in hindi

## PHP स्क्रिप्ट कैसे लिखें और चलाएं? (How to Write & Execute PHP Script in Hindi)

### Step 1: PHP के लिए जरूरी टूल्स इंस्टॉल करें

PHP स्क्रिप्ट चलाने के लिए आपको एक सर्वर चाहिए जो PHP को सपोर्ट करे। इसके लिए दो लोकप्रिय विकल्प हैं:

## Option 1: XAMPP (सर्वाधिक लोकप्रिय)

XAMPP एक फ्री सॉफ्टवेयर पैकेज है जिसमें तीन चीज़ें होती हैं:

- **Apache** (Web Server)
- **MySQL / MariaDB** (Database)
- **PHP** (PHP Engine)

डाउनलोड लिंक: <https://www.apachefriends.org/index.html>

## Option 2: WAMP / MAMP / LAMP

- **WAMP** (Windows के लिए)
- **MAMP** (Mac के लिए)
- **LAMP** (Linux के लिए)

## Step 2: पहली PHP स्क्रिप्ट लिखें

PHP फाइल को .php एक्सटेंशन से सेव किया जाता है।

उदाहरण के लिए: hello.php

### hello.php:

```
<!DOCTYPE html>
<html>
<head>
  <title>PHP Example</title>
</head>
<body>
  <h1>नमस्ते दुनिया!</h1>
  <?php
    echo "यह PHP कोड द्वारा जनरेट हुआ टेक्स्ट है।";
  ?>
</body>
</html>
```

## Step 3: PHP फाइल को सही फ़ोल्डर में सेव करें

XAMPP इंस्टॉल होने के बाद आपकी PHP फाइल को निम्नलिखित फोल्डर में रखना होगा:

### फ़ोल्डर पथ:

C:\xampp\htdocs\

उदाहरण के लिए:

C:\xampp\htdocs\hello.php

## Step 4: XAMPP सर्वर चालू करें

1. XAMPP Control Panel खोलें

2. **Apache** सर्विस को **Start** करें
3. यदि आप डाटाबेस का उपयोग कर रहे हैं, तो **MySQL** भी Start करें

## Step 5: ब्राउज़र से PHP स्क्रिप्ट चलाएं

1. अपने ब्राउज़र में जाएँ (जैसे Chrome)
2. यह URL टाइप करें:

`http://localhost/hello.php`

यदि आपने फाइल किसी सबफ़ोल्डर में रखी है, जैसे `htdocs/project1/hello.php`, तो URL होगा:

`http://localhost/project1/hello.php`

## Extra: PHP को Command Line से भी चला सकते हैं

अगर आप GUI नहीं चाहते और सिर्फ कोड चलाना है, तो:

### कमांड लाइन से चलाने के लिए:

1. Command Prompt (CMD) खोलें
2. उस फोल्डर पर जाएँ जहाँ आपकी PHP फाइल है:

```
cd C:\xampp\htdocs
```

3. यह कमांड चलाएँ:

```
php hello.php
```

ध्यान दें: CLI में HTML नहीं दिखेगा, सिर्फ PHP का आउटपुट मिलेगा।

## PHP स्क्रिप्टिंग के बेसिक सिंटैक्स

```
<?php
// यह एक कमेंट है
echo "Hello World!";
?>
```

### PHP टैग्स:

- PHP कोड हमेशा `<?php ... ?>` टैग के अंदर लिखा जाता है।

## Common Errors:

Error	कारण
localhost not working	Apache चालू नहीं है
unexpected EOF	PHP टैग्स सही से बंद नहीं किए
echo output नहीं दिख रहा	PHP फाइल को ब्राउज़र से नहीं खोला या HTML गलत है

## Q.4 Write note Oops concepts. full content solution in hindi

### OOPs Concepts in Hindi

(ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग की संपूर्ण जानकारी)

### OOP (ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग) क्या है?

OOP एक प्रोग्रामिंग पैराडाइम (Programming Paradigm) है जिसमें प्रोग्राम को ऑब्जेक्ट्स (Objects) के रूप में डिज़ाइन किया जाता है। इन ऑब्जेक्ट्स में डेटा और उस डेटा पर कार्य करने वाले फंक्शन्स दोनों शामिल होते हैं।

इसका उद्देश्य कोड को:

- पुनः उपयोग योग्य (Reusable)
- आसान मेंटेन (Maintainable)
- सुरक्षित (Secure)
- और अधिक संगठित (Organized) बनाना है।

### OOP के मुख्य सिद्धांत (Main Concepts of OOP in Hindi)

#### 1 Class (क्लास)

क्लास एक ब्लूप्रिंट (Blueprint) होती है, जो ऑब्जेक्ट्स बनाने के लिए एक खाका देती है। इसमें प्रॉपर्टीज (डेटा) और मेथड्स (फंक्शन) होते हैं।

उदाहरण (PHP में):

```
class Car {  
    public $color;  
    public function drive() {  
        echo "गाड़ी चल रही है।";  
    }  
}
```

#### 2 Object (ऑब्जेक्ट)

ऑब्जेक्ट क्लास का एक इंस्टेंस (Instance) होता है। एक क्लास से कई ऑब्जेक्ट बनाए जा सकते हैं।

उदाहरण:

```
$myCar = new Car(); // Car क्लास का एक ऑब्जेक्ट  
$myCar->color = "Red";  
$myCar->drive();
```



### 3 Encapsulation (एन्कैप्सुलेशन)

यह सिद्धांत कहता है कि डेटा और फंक्शन को एक यूनिट (क्लास) में बांध देना चाहिए और बाहर से डायरेक्ट एक्सेस को रोका जाए।

प्रोटेक्टेड डेटा को गेटर (Getter) और सेटर (Setter) के ज़रिए एक्सेस किया जाता है।

उदाहरण:

```
class Person {
    private $name;

    public function setName($n) {
        $this->name = $n;
    }

    public function getName() {
        return $this->name;
    }
}
```

### 4 Inheritance (इनहेरिटेंस)

एक क्लास दूसरी क्लास से गुण (Properties) और क्रियाएं (Methods) प्राप्त कर सकती है। इससे कोड का पुनः उपयोग (Reusability) होता है।

उदाहरण:

```
class Animal {
    public function sound() {
        echo "कोई ध्वनि!";
    }
}

class Dog extends Animal {
    public function bark() {
        echo "भौं भौं!";
    }
}

$d = new Dog();
$d->sound(); // Parent class method
$d->bark();  // Child class method
```

### 5 Polymorphism (पॉलीमॉर्फिज्म)

इसका अर्थ होता है: एक चीज़ के कई रूप। इसमें एक ही नाम के फंक्शन अलग-अलग व्यवहार कर सकते हैं।

दो प्रकार होते हैं:

- **Compile Time Polymorphism (Method Overloading)** → PHP में सीमित है
- **Run Time Polymorphism (Method Overriding)** → PHP में संभव है

उदाहरण:

```
class Shape {
    public function draw() {
        echo "कोई आकृति बनाई जा रही है।";
    }
}
```

```

    }
}

class Circle extends Shape {
    public function draw() {
        echo "सर्कल बन रहा है।";
    }
}

$obj = new Circle();
$obj->draw(); // सर्कल वाला method call होगा

```

## 6 Abstraction (एब्स्ट्रैक्शन)

यह सिद्धांत कहता है कि **सिर्फ जरूरी जानकारी यूज़र को दिखाओ, और बाकी डिटेल्स छुपाओ।**

यह **abstract classes** और **interfaces** के ज़रिए किया जाता है।

```

abstract class Vehicle {
    abstract public function startEngine();
}

class Bike extends Vehicle {
    public function startEngine() {
        echo "बाइक स्टार्ट हो रही है।";
    }
}

```

## निष्कर्ष (Conclusion)

सिद्धांत	कार्य
Class	Object का ढांचा
Object	Class का Instance
Encapsulation	डेटा को छुपाना
Inheritance	कोड का पुनः उपयोग
Polymorphism	एक नाम, कई व्यवहार
Abstraction	केवल ज़रूरी जानकारी दिखाना

## Q.5 How does Php Works? what is timeout in sessions? full content solution in hindi

PHP कैसे काम करता है? (How PHP Works in Hindi)

### PHP कैसे कार्य करता है?

जब हम PHP फाइल को ब्राउज़र में एक्सेस करते हैं (जैसे `http://localhost/index.php`), तो प्रक्रिया कुछ इस तरह होती है:

1. यूज़र ब्राउज़र से रिक्वेस्ट भेजता है।

उदाहरण: Chrome या Firefox में कोई URL टाइप करते हैं।

2. **रिक्वेस्ट सर्वर तक पहुँचती है।**  
सर्वर (जैसे Apache Server) उस फाइल को ढूँढता है।
3. **सर्वर PHP इंजन को बुलाता है।**  
PHP कोड को पढ़ने और प्रोसेस करने के लिए PHP Parser सक्रिय होता है।
4. **PHP कोड सर्वर पर एक्सेक्यूट होता है।**  
PHP फाइल के अंदर लिखा गया कोड सर्वर पर चलता है — जैसे:
  - डाटाबेस से डेटा निकालना
  - कैलकुलेशन करना
  - फाइल सिस्टम से फाइल पढ़ना या लिखना
5. **आउटपुट HTML के रूप में जनरेट होता है।**  
PHP फाइल का जो आउटपुट है, वह HTML में बदल जाता है।
6. **ब्राउज़र को HTML भेजा जाता है।**  
ब्राउज़र को सिर्फ HTML दिखता है, PHP का कोड कभी नहीं दिखता।

## PHP कार्यप्रणाली को सरल भाषा में एक डायग्राम:

User Request → Apache Server → PHP Engine → Execute Code → Output HTML → Send to Browser

## PHP की कार्य प्रक्रिया का उदाहरण:

**index.php:**

```
<?php  
echo "आज की तारीख है: " . date("d-m-Y");  
?>
```

ब्राउज़र में आपको दिखेगा:

आज की तारीख है: 26-04-2025

लेकिन PHP कोड (<?php echo ... ?>) ब्राउज़र में दिखाई नहीं देगा, सिर्फ उसका आउटपुट दिखेगा।

## PHP में Session Timeout क्या है?

### Session क्या होता है?

- **Session** का मतलब है:  
यूज़र के डेटा को एक पेज से दूसरे पेज तक बनाए रखना।

उदाहरण:

जब आप वेबसाइट में लॉगिन करते हैं, और फिर दूसरे पेज पर जाते हैं, तब भी आपका लॉगिन स्टेटस बना रहता है — यह **Session** की मदद से होता है।

## Session Timeout क्या होता है?

Session Timeout का मतलब है:

अगर यूज़र एक निश्चित समय (Time Limit) तक सक्रिय नहीं रहा, तो उसका Session अपने आप समाप्त (Expire) हो जाएगा।

इससे सुरक्षा बढ़ती है, जैसे:

- अगर कोई यूज़र लॉगिन करने के बाद सिस्टम छोड़ दे, तो ऑटोमेटिकली लॉगआउट हो जाए।
- अनधिकृत एक्सेस से बचाव हो।

## PHP में Default Session Timeout:

PHP में डिफ़ॉल्ट सेशन टाइमआउट 24 मिनट (1440 सेकंड) का होता है।

यह सेटिंग PHP के कॉन्फिगरेशन फाइल `php.ini` में होती है:

```
session.gc_maxlifetime = 1440
```

(यह सेकंड में होता है — 1440 सेकंड = 24 मिनट)

## Session Timeout को कैसे बदलें?

आप अपने कोड में भी Session Timeout सेट कर सकते हैं:

```
<?php
session_start();
ini_set('session.gc_maxlifetime', 1800); // 1800 सेकंड यानी 30 मिनट
?>
```

या फिर अपने सर्वर की `php.ini` फाइल में बदलाव कर सकते हैं।

## Session Timeout को Manage करने का उदाहरण:

```
<?php
session_start();

// यदि last activity से 15 मिनट से ज्यादा समय हो गया
if (isset($_SESSION['LAST_ACTIVITY']) && (time() - $_SESSION['LAST_ACTIVITY'] > 900)) {
    session_unset();    // सेशन वेरिएबल्स को हटाओ
    session_destroy();  // सेशन को खत्म करो
}
$_SESSION['LAST_ACTIVITY'] = time(); // हर एक्टिविटी पर समय अपडेट करो
?>
```

**Q.6 Write a note php .INI file full content solution in hindi**

# PHP .INI फाइल पर सम्पूर्ण जानकारी (Full Content Solution in Hindi)

## PHP .ini फाइल क्या है?

**PHP.ini** एक **Configuration File** (कॉन्फिगरेशन फ़ाइल) है, जिसका उपयोग PHP के व्यवहार (behavior) और डिफ़ॉल्ट सेटिंग्स को नियंत्रित (control) करने के लिए किया जाता है।

जब भी PHP इंजन चालू होता है (जैसे Apache सर्वर के साथ या CLI से), सबसे पहले यह .ini फाइल पढ़ता है और उसमें दी गई सेटिंग्स को लागू करता है।

## PHP.ini फाइल का काम क्या है?

- मेमोरी लिमिट सेट करना
- फाइल अपलोड का आकार नियंत्रित करना
- डिफ़ॉल्ट टाइमज़ोन सेट करना
- एरर रिपोर्टिंग नियंत्रित करना
- सेशन से संबंधित सेटिंग्स (जैसे timeout) करना
- डिफ़ॉल्ट एक्सटेंशन्स को इनेबल या डिसेबल करना

## PHP.ini फाइल कहाँ होती है?

PHP इंस्टॉल करने के बाद, PHP.ini फाइल आमतौर पर इन लोकेशन पर मिलती है:

सर्वर	लोकेशन
XAMPP	C:\xampp\php\php.ini
WAMP	C:\wamp\bin\php\php{version}\php.ini
LAMP (Linux)	/etc/php/{version}/apache2/php.ini

आप `phpinfo()` फंक्शन चलाकर भी सही लोकेशन देख सकते हैं:

```
<?php phpinfo(); ?>
```

## PHP.ini में क्या-क्या मुख्य सेटिंग्स होती हैं?

नीचे PHP.ini की कुछ प्रमुख सेटिंग्स दी गई हैं:

### 1 Error Reporting

PHP में एरर दिखानी है या नहीं, यह सेट किया जाता है।

```
error_reporting = E_ALL
display_errors = On
```

- E\_ALL → सभी तरह की Errors दिखाना
- display\_errors → ब्राउज़र में Errors दिखाना

## 2 File Uploads

PHP में कितनी बड़ी फाइल अपलोड हो सकती है, उसे नियंत्रित करना।

```
file_uploads = On
upload_max_filesize = 2M
post_max_size = 8M
```

- file\_uploads = On → फाइल अपलोडिंग चालू
- upload\_max\_filesize → एक फाइल की अधिकतम साइज
- post\_max\_size → फॉर्म के कुल डेटा की अधिकतम साइज

## 3 Memory Limit

एक PHP स्क्रिप्ट अधिकतम कितनी मेमोरी इस्तेमाल कर सकती है:

```
memory_limit = 128M
```

## 4 Session Settings

सेशन से जुड़े नियम सेट करना:

```
session.gc_maxlifetime = 1440
session.save_path = "/tmp"
```

- session.gc\_maxlifetime → सेशन लाइफ टाइम (सेकंड में)
- session.save\_path → सेशन फाइलों को कहाँ स्टोर करना है

## 5 Timezone सेट करना

डिफ़ॉल्ट टाइमज़ोन सेट करना:

```
date.timezone = "Asia/Kolkata"
```

## 6 Extensions इनेबल करना

PHP में एक्स्ट्रा फीचर्स जोड़ने के लिए एक्सटेंशन्स ऑन करना:

```
extension=mysqli
extension=curl
extension=gd
```

## PHP.ini में बदलाव कैसे करें?

1. PHP.ini फाइल को किसी Text Editor (जैसे Notepad++) से खोलें।

2. आवश्यकतानुसार सेटिंग्स को एडिट करें।
3. फाइल को सेव करें।
4. **Apache या अपना Web Server को Restart करें** ताकि बदलाव लागू हो जाएं।

## PHP.ini Editing के दौरान सावधानियाँ

- गलत एडिटिंग से PHP सर्वर क्रैश कर सकता है।
- हमेशा Backup बनाकर रखें।
- फाइल सेव करने के बाद सर्वर को **Restart करना जरूरी** है।
- Syntax में कोई गलती (जैसे टाइपो) न करें।

## उदाहरण: PHP.ini में फाइल अपलोड साइज बढ़ाना

अगर आप चाहते हैं कि 10MB तक की फाइल अपलोड हो सके, तो:

```
upload_max_filesize = 10M  
post_max_size = 20M
```

इसके बाद Apache को Restart करें।

## निष्कर्ष (Conclusion)

विषय	विवरण
PHP.ini क्या है	PHP की मुख्य कॉन्फिगरेशन फाइल
मुख्य कार्य	PHP के व्यवहार को नियंत्रित करना
संपादन कैसे करें	Text Editor से बदलाव करें और सर्वर Restart करें
महत्वपूर्ण सेटिंग्स	Error Reporting, Memory Limit, File Uploads, Sessions, Extensions

**Q.7 What is variable ? Write the rule of declaring variables.**

## Variable क्या है? (What is a Variable in Hindi)

### Variable (चर) क्या होता है?

**Variable** एक कंटेनर (Container) होता है, जिसमें हम कोई भी डेटा स्टोर कर सकते हैं। यह कंप्यूटर मेमोरी में किसी वैल्यू को एक नाम देकर स्टोर करने का तरीका है।

आप Variable का इस्तेमाल करके डेटा को बाद में इस्तेमाल कर सकते हैं, अपडेट कर सकते हैं या प्रोसेस कर सकते हैं।

## आसान भाषा में:

Variable = **नामित बॉक्स** जिसमें हम कुछ सामान (डेटा) रख सकते हैं।

जैसे:

- एक Variable में हम किसी स्टूडेंट का नाम स्टोर कर सकते हैं।
- किसी प्रोडक्ट की कीमत स्टोर कर सकते हैं।
- या फिर किसी वेबसाइट पर लॉगिन यूज़र का ID रख सकते हैं।

## PHP में Variable कैसे काम करते हैं?

PHP में Variable का नाम \$ चिन्ह से शुरू होता है।

**उदाहरण:**

```
<?php
$name = "Rahul";
$age = 25;
$price = 199.99;
?>
```

यहाँ:

- \$name → एक टेक्स्ट (String) स्टोर कर रहा है।
- \$age → एक नंबर (Integer) स्टोर कर रहा है।
- \$price → एक दशमलव संख्या (Float) स्टोर कर रहा है।

## Variables Declare करने के नियम (Rules for Declaring Variables)

PHP में Variable बनाते समय कुछ नियमों का पालन करना जरूरी है

1 Variable का नाम \$ (डॉलर साइन) से शुरू होना चाहिए।

```
$name = "Amit";
```

2 Variable का नाम अल्फाबेट (A-Z, a-z) या अंडरस्कोर \_ से शुरू हो सकता है, नंबर से नहीं।

सही:

```
$student = "Ravi";
$_marks = 80;
```

गलत:



`$student = "Ravi";` // गलत है (नंबर से शुरू नहीं हो सकता)

**3 Variable का नाम केवल अल्फाबेट्स (A-Z, a-z), नंबर्स (0-9), और अंडरस्कोर (\_) का कॉम्बिनेशन हो सकता है।**

सही:

```
$total_marks = 100;  
$marks2 = 90;
```

गलत:

```
$total-marks = 100; // डैश (-) का इस्तेमाल नहीं कर सकते
```

**4 Variable नाम में स्पेस (Space) नहीं होना चाहिए।**

सही:

```
$firstName = "Raj";
```

गलत:

```
$first Name = "Raj"; // Space allowed नहीं है
```

**5 PHP में Variables Case Sensitive होते हैं।**

यानि \$name, \$Name, और \$NAME तीनों अलग-अलग Variables हैं।

```
<?php  
$name = "Arjun";  
$Name = "Vikram";  
  
echo $name; // Arjun  
echo $Name; // Vikram  
?>
```

**6 Variable का नाम अर्थपूर्ण (Meaningful) होना चाहिए।**

अच्छा अभ्यास (Best Practice):

- सही: \$userName, \$productPrice
- गलत: \$x, \$y (समझने में कठिन)

## निष्कर्ष (Conclusion)

बिंदु	विवरण
Variable क्या है	डेटा स्टोर करने का नामित स्थान
शुरुआत कैसे करें	\$ से शुरू करें
नाम में क्या होना चाहिए	अक्षर, नंबर, अंडरस्कोर
केस संवेदनशीलता	हाँ, Case Sensitive होता है

## Q.8 Explain operators and write its types.

# Operators क्या होते हैं? (What are Operators in PHP - Hindi Explanation)

## Operators क्या हैं?

**Operator** वे Symbols होते हैं जिनकी मदद से हम किसी Variable या Value पर कोई Operation (जैसे जोड़, घटाव, तुलना, असाइन करना आदि) कर सकते हैं।

उदाहरण: +, -, \*, ==, =, && आदि सभी Operators हैं।

## उदाहरण:

```
<?php
$a = 10;
$b = 5;
$c = $a + $b; // यहाँ '+' एक Operator है जो जोड़ने का काम कर रहा है
?>
```

# PHP में Operators के प्रकार (Types of Operators in PHP)

PHP में मुख्यतः 7 प्रकार के Operators होते हैं:

## 1 Arithmetic Operators (गणितीय ऑपरेटर्स)

Operator	कार्य (Meaning)	उदाहरण
+	जोड़ (Addition)	\$a + \$b
-	घटाना (Subtraction)	\$a - \$b
*	गुणा (Multiplication)	\$a * \$b
/	भाग (Division)	\$a / \$b
%	शेषफल (Modulus)	\$a % \$b

## 2 Assignment Operators (असाइनमेंट ऑपरेटर्स)

Operator	कार्य	उदाहरण
=	वैल्यू असाइन करना	\$x = 10;
+=	जोड़कर असाइन करना	\$x += 5; (मतलब: \$x = \$x + 5)
-=	घटाकर असाइन करना	\$x -= 3;
*=	गुणा करके असाइन करना	\$x *= 2;
/=	भाग देकर असाइन करना	\$x /= 2;

### 3 Comparison Operators (तुलना करने वाले ऑपरेटर्स)

Operator	कार्य	उदाहरण
==	बराबर है या नहीं	\$a == \$b
!=	बराबर नहीं है	\$a != \$b
===	वैल्यू और टाइप दोनों बराबर	\$a === \$b
!==	वैल्यू या टाइप अलग	\$a !== \$b
>	बड़ा है	\$a > \$b
<	छोटा है	\$a < \$b
>=	बड़ा या बराबर	\$a >= \$b
<=	छोटा या बराबर	\$a <= \$b

### 4 Logical Operators (तार्किक ऑपरेटर्स)

Operator	कार्य	उदाहरण
&& या and	दोनों सही हों	(\$a > 5 && \$b < 10)
`	याor`	
!	उल्टा (NOT)	!(\$a == \$b)

### 5 Increment / Decrement Operators

Operator	कार्य	उदाहरण
++\$X	पहले बढ़ाओ फिर इस्तेमाल करो	प्री-इंक्रीमेंट
\$X++	पहले इस्तेमाल करो फिर बढ़ाओ	पोस्ट-इंक्रीमेंट
--\$X	पहले घटाओ फिर इस्तेमाल करो	प्री-डिक्रीमेंट
\$X--	पहले इस्तेमाल करो फिर घटाओ	पोस्ट-डिक्रीमेंट

### 6 String Operators

Operator	कार्य	उदाहरण
.	Strings को जोड़ना	\$a . \$b
.=	जोड़कर असाइन करना	\$a .= \$b

```
$name = "Rahul";  
$name .= " Sharma"; // Output: Rahul Sharma
```

### 7 Array Operators (एरे ऑपरेटर्स)

Operator	कार्य	उदाहरण
+	दो arrays को जोड़ना	\$a + \$b
==	दोनों arrays में वैल्यू बराबर	\$a == \$b
===	वैल्यू और क्रम दोनों बराबर	\$a === \$b
!=	बराबर नहीं	\$a != \$b

## Q.9 What are Various data types in php?

# PHP में डेटा टाइप्स (Data Types in PHP)

PHP एक **loosely typed** (या dynamically typed) भाषा है, जिसका मतलब है कि PHP में हमें डेटा टाइप्स को पहले से घोषित करने की आवश्यकता नहीं होती। PHP खुद से यह निर्धारित करता है कि किसी वैरिएबल में किस प्रकार का डेटा होगा।

PHP में मुख्यतः 4 प्रकार के डेटा टाइप्स होते हैं:

1. **Scalar Data Types** (मात्रात्मक डेटा प्रकार)
2. **Compound Data Types** (संयोजित डेटा प्रकार)
3. **Special Data Types** (विशेष डेटा प्रकार)
4. **Pseudo-Data Types** (काल्पनिक डेटा प्रकार)

## 1 Scalar Data Types (मात्रात्मक डेटा प्रकार)

यह वे डेटा प्रकार होते हैं जो केवल एक ही वैल्यू को स्टोर करते हैं।

### 1.1 Integer (पूरा अंक)

- यह डेटा प्रकार केवल पूर्णांक (whole numbers) को स्टोर करता है।
- यह साइन (positive or negative) और नॉन-डेसिमल (non-decimal) वैल्यू होती है।

```
<?php
$age = 25; // Integer (पूर्णांक)
$year = -2025; // Negative Integer
?>
```

### 1.2 Float (दशमलव अंक)

- फ्लोट (या Double) दशमलव (decimal) संख्याएँ होती हैं।
- यह संख्याएँ दशमलव के बाद कुछ अंश (fractional part) शामिल करती हैं।

```
<?php
$price = 19.99; // Float (दशमलव अंक)
$distance = 100.5; // Another Float
?>
```

### 1.3 String (स्ट्रिंग)

- स्ट्रिंग एक या एक से अधिक वर्णों का संग्रह होता है।
- PHP में स्ट्रिंग डबल कोड्स (") या सिंगल कोड्स (') के बीच रखी जाती है।

```
<?php
$name = "Rahul"; // String (स्ट्रिंग)
$message = 'Hello, World!'; // Another String
```

?>

## 1.4 Boolean (बूलियन)

- Boolean डेटा प्रकार केवल दो मान ले सकता है: TRUE या FALSE।
- इसका उपयोग सत्य (True) या असत्य (False) की स्थिति को व्यक्त करने के लिए किया जाता है।

```
<?php
$isAdult = true; // Boolean (सत्य)
$isStudent = false; // Boolean (झूठ)
?>
```

## 2 Compound Data Types (संयोजित डेटा प्रकार)

संयोजित डेटा प्रकार वह होते हैं जो एक से अधिक मानों को एक साथ संग्रहित कर सकते हैं।

### 2.1 Array (एरे)

- एरे एक संग्रह (collection) होता है, जिसमें एक से अधिक वैल्यूज हो सकती हैं।
- एरे के अंदर विभिन्न प्रकार के डेटा रखे जा सकते हैं।

```
<?php
$fruits = array("Apple", "Banana", "Mango"); // Indexed Array
$person = array("name" => "John", "age" => 25); // Associative Array
?>
```

### 2.2 Object (ऑब्जेक्ट)

- PHP में एक ऑब्जेक्ट किसी क्लास का एक उदाहरण (instance) होता है।
- ऑब्जेक्ट में प्रॉपर्टीज़ और मेथड्स (methods) होते हैं जो किसी चीज़ का प्रतिनिधित्व (representation) करते हैं।

```
<?php
class Person {
    public $name;
    public $age;

    function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }

    function greet() {
        echo "Hello, my name is " . $this->name;
    }
}

$person1 = new Person("Raj", 30);
$person1->greet();
?>
```

## 3 Special Data Types (विशेष डेटा प्रकार)

PHP में कुछ विशेष डेटा प्रकार भी होते हैं जो विशेष स्थितियों में उपयोग होते हैं।

### 3.1 NULL

- NULL एक विशेष डेटा प्रकार है, जो दर्शाता है कि किसी वेरिएबल की कोई वैल्यू नहीं है।
- NULL किसी वेरिएबल की वैल्यू की अनुपस्थिति (absence) को दर्शाता है।

```
<?php
$var = NULL; // NULL
?>
```

## 4 Pseudo-Data Types (काल्पनिक डेटा प्रकार)

PHP में कुछ काल्पनिक (pseudo) डेटा प्रकार होते हैं जो किसी विशेष प्रकार का डेटा नहीं होते, लेकिन इसका उपयोग तकनीकी दृष्टिकोण से किया जाता है।

### 4.1 Mixed

- यह डेटा प्रकार तब उपयोग होता है जब एक वेरिएबल किसी भी प्रकार का डेटा स्टोर कर सकता है, जैसे Integer, String या Array आदि।

```
<?php
function example($value): mixed {
    return $value;
}
?>
```

### 4.2 Resource

- यह डेटा प्रकार तब होता है जब PHP किसी बाहरी संसाधन (external resource) जैसे कि एक फाइल, डाटाबेस कनेक्शन या कोई अन्य सिस्टम संसाधन के साथ काम करता है।

```
<?php
$file = fopen("example.txt", "r"); // Resource (File handle)
?>
```

## Q.10 What is AJAX ? Explain AJAX events.

## AJAX (Asynchronous JavaScript and XML) क्या है?

AJAX एक तकनीक है जो वेब पेजों को बिना पेज को रीफ्रेश किए डेटा लोड करने की अनुमति देती है। इसका उपयोग मुख्य रूप से वेब पेजों के इंटरएक्शन को तेज और बेहतर बनाने के लिए किया जाता है। AJAX के माध्यम से, आप पृष्ठ को पूरी तरह से लोड किए बिना, केवल उस पेज के कुछ हिस्सों को अपडेट कर सकते हैं, जिससे उपयोगकर्ता अनुभव (UX) बेहतर होता है।

AJAX **Asynchronous** होता है, जिसका मतलब है कि जब वेब पेज डेटा लोड कर रहा होता है, तब उपयोगकर्ता इंटरफेस पर अन्य गतिविधियाँ बिना रुकावट के चल सकती हैं।

## AJAX कैसे काम करता है?

1. **JavaScript** के द्वारा एक HTTP request भेजी जाती है (सामान्यतः XMLHttpRequest ऑब्जेक्ट का उपयोग किया जाता है)।
2. यह request सर्वर पर भेजी जाती है, जहाँ से डेटा प्राप्त होता है।
3. सर्वर से डेटा मिलने के बाद, पेज के उस हिस्से को बिना पूरे पेज को रीफ्रेश किए अपडेट कर दिया जाता है।

## AJAX के लाभ

- **तेज़ लोडिंग:** पेज के सिर्फ एक हिस्से को लोड किया जाता है, जिससे पेज का लोडिंग समय कम हो जाता है।
- **बेहतर यूज़र इंटरफ़ेस:** पेज के बिना पूरी तरह से लोड किए डेटा अपडेट होता है, जिससे इंटरएक्टिव अनुभव मिलता है।
- **संसाधनों का कम उपयोग:** पेज को बार-बार लोड नहीं किया जाता, जिससे सर्वर और क्लाइंट संसाधनों का कुशल उपयोग होता है।

## AJAX Events (AJAX इवेंट्स)

AJAX के कार्य को नियंत्रित करने के लिए कुछ प्रमुख इवेंट्स होते हैं। ये इवेंट्स AJAX रिक्वेस्ट के विभिन्न चरणों में होते हैं, जैसे रिक्वेस्ट भेजना, सर्वर से डेटा प्राप्त करना, और डेटा प्राप्त होने के बाद के परिणामों को हैंडल करना।

### 1 onreadystatechange

यह सबसे सामान्य और महत्वपूर्ण इवेंट होता है। यह इवेंट हर बार तब ट्रिगर होता है जब HTTP request का स्टेटस बदलता है। जब request का स्टेटस बदलता है, तो यह इवेंट रिक्वेस्ट की स्टेटस को ट्रैक करता है और उसके अनुसार कुछ कार्य करता है।

### स्टेटस कोड:

- 0 → रिक्वेस्ट नहीं भेजी गई है।
- 1 → सर्वर से कनेक्शन स्थापित हुआ है।
- 2 → सर्वर से उत्तर प्राप्त हुआ है।
- 3 → सर्वर से उत्तर प्राप्त होते हुए।
- 4 → सर्वर ने जवाब भेज दिया है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementById("result").innerHTML = this.responseText;
    }
};
xhttp.open("GET", "example.txt", true);
xhttp.send();
```

## 2 onload

यह इवेंट तब ट्रिगर होता है जब AJAX रिक्वेस्ट सफलतापूर्वक पूरी हो जाती है और सर्वर से डेटा प्राप्त हो जाता है। यह **readyState 4** और **status 200** के बाद काम करता है। यह इवेंट मुख्य रूप से सर्वर से डेटा लोड होने के बाद के कार्यों को हैंडल करता है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();
xhttp.onload = function() {
    if (this.status == 200) {
        document.getElementById("result").innerHTML = this.responseText;
    }
};
xhttp.open("GET", "data.txt", true);
xhttp.send();
```

## 3 onerror

यह इवेंट तब ट्रिगर होता है जब AJAX रिक्वेस्ट में कोई समस्या होती है (जैसे सर्वर डाउन हो, या नेटवर्क समस्या हो)। यह इवेंट एरर को हैंडल करने के लिए उपयोगी होता है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();
xhttp.onerror = function() {
    alert("Error occurred while sending the request.");
};
xhttp.open("GET", "nonexistent-file.txt", true);
xhttp.send();
```

## 4 onprogress

यह इवेंट तब ट्रिगर होता है जब डेटा सर्वर से लोड हो रहा होता है, जिससे आप लोडिंग प्रोग्रेस (progress) को ट्रैक कर सकते हैं। यह इवेंट खासकर तब उपयोगी होता है जब आप फाइल अपलोड या डाउनलोड करते हैं और आपको लोडिंग स्टेटस दिखानी होती है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();
xhttp.onprogress = function(event) {
    if (event.lengthComputable) {
        var percent = (event.loaded / event.total) * 100;
        console.log("Progress: " + percent + "%");
    }
};
xhttp.open("GET", "largefile.txt", true);
xhttp.send();
```

## 5 onabort

यह इवेंट तब ट्रिगर होता है जब यूज़र या कोड द्वारा AJAX रिक्वेस्ट को रोक दिया जाता है। जब रिक्वेस्ट का प्रोसेस रुक जाता है तो यह इवेंट एक्टिव होता है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();
xhttp.onabort = function() {
```



```
    alert("Request was aborted.");  
};  
xhttp.open("GET", "example.txt", true);  
xhttp.send();  
xhttp.abort(); // Request is aborted
```

## 6 ontimeout

यह इवेंट तब ट्रिगर होता है जब AJAX रिक्वेस्ट टाइमआउट हो जाता है, यानी अगर सर्वर से समय सीमा (timeout) के अंदर कोई प्रतिक्रिया नहीं मिलती है। यह इवेंट एक समस्या को सूचित करने के लिए उपयोगी होता है।

### उदाहरण:

```
var xhttp = new XMLHttpRequest();  
xhttp.ontimeout = function() {  
    alert("Request timed out.");  
};  
xhttp.open("GET", "example.txt", true);  
xhttp.timeout = 5000; // Timeout after 5 seconds  
xhttp.send();
```

## Q.11 What is static keyword ? Explain with example.

## PHP में static कीवर्ड क्या है?

static कीवर्ड का उपयोग PHP में वेरिएबल्स, फंक्शन्स और क्लास प्रॉपर्टीज को स्थिर (static) बनाने के लिए किया जाता है। इसका मुख्य उद्देश्य यह है कि एक बार static वेरिएबल की वैल्यू सेट होने के बाद, उसकी वैल्यू उसी स्थिति में बनी रहती है, भले ही वह वेरिएबल किसी फंक्शन के अंदर डिक्लेर किया गया हो।

### static का उपयोग कहाँ किया जाता है?

1. Static Variables in Functions
2. Static Methods in Classes
3. Static Properties in Classes

आइए इनको विस्तार से समझते हैं।

### 1 Static Variables in Functions (फंक्शन्स में स्टैटिक वेरिएबल्स)

जब हम किसी वेरिएबल को static घोषित करते हैं, तो उसका मान किसी भी फंक्शन कॉल के बाद भी बनाए रहता है। सामान्यतः फंक्शन के अंदर डिक्लेर किए गए वेरिएबल हर बार फंक्शन कॉल के साथ रीसेट हो जाते हैं, लेकिन static वेरिएबल्स का मान हमेशा बचा रहता है।

#### Example:

```
<?php
```

```
function counter() {
    static $count = 0; // Static Variable
    $count++;
    echo $count . "\n";
}

counter(); // Output: 1
counter(); // Output: 2
counter(); // Output: 3
?>
```

### समझाइए:

- ऊपर दिए गए उदाहरण में, \$count को static के रूप में डिक्लेर किया गया है।
- प्रत्येक बार जब हम counter() फंक्शन को कॉल करते हैं, तो \$count का मान बढ़ता रहता है, लेकिन वह रीसेट नहीं होता, जैसा कि सामान्य वेरिएबल के साथ होता है।
- इसलिए, आउटपुट में हर बार \$count का मान बढ़ता है, और यह वैल्यू प्रत्येक फंक्शन कॉल के दौरान बनाए रहती है।

## 2 Static Methods in Classes (क्लास में स्टैटिक मेथड्स)

static मेथड्स को एक क्लास के किसी विशेष ऑब्जेक्ट के बिना भी कॉल किया जा सकता है। ये मेथड्स क्लास के ऑब्जेक्ट से संबंधित नहीं होते, बल्कि क्लास के स्तर पर होते हैं। इन्हें सीधे क्लास से कॉल किया जाता है।

### Example:

```
<?php
class MyClass {
    public static $message = "Hello, World!"; // Static Property

    public static function greet() { // Static Method
        echo "Greetings from the static method!";
    }
}

// Static method and property can be accessed without creating an object
echo MyClass::$message . "\n"; // Output: Hello, World!
MyClass::greet(); // Output: Greetings from the static method!
?>
```

### समझाइए:

- static मेथड्स और प्रॉपर्टीज़ क्लास के ऑब्जेक्ट से स्वतंत्र होते हैं।
- MyClass::\$message और MyClass::greet() को हम क्लास के नाम से सीधे एक्सेस कर सकते हैं, बिना क्लास का ऑब्जेक्ट बनाए।
- यदि हम इन्हें बिना static के बनाते, तो हमें पहले क्लास का ऑब्जेक्ट बनाना पड़ता।

## 3 Static Properties in Classes (क्लास में स्टैटिक प्रॉपर्टीज़)

static प्रॉपर्टीज़ क्लास के एक निश्चित उदाहरण से संबंधित नहीं होती हैं। वे क्लास के स्तर पर होती हैं और सभी ऑब्जेक्ट्स में साझा होती हैं। जब आप किसी static प्रॉपर्टी को बदलते हैं, तो वह प्रॉपर्टी सभी ऑब्जेक्ट्स में बदली जाती है।

### Example:

```
<?php
class Counter {
    public static $count = 0; // Static Property

    public function increment() {
        self::$count++; // Access static property using 'self'
    }
}

// Creating objects
$obj1 = new Counter();
$obj1->increment();
$obj1->increment();

$obj2 = new Counter();
$obj2->increment();

// Displaying the static property
echo Counter::$count; // Output: 3
?>
```

### समझाइए:

- static प्रॉपर्टी Counter::\$count सभी ऑब्जेक्ट्स द्वारा साझा की जाती है।
- जब हम increment() मेथड को कॉल करते हैं, तो static प्रॉपर्टी का मान बढ़ता है, और यह मान सभी ऑब्जेक्ट्स में एक जैसा होता है।
- यहाँ self::\$count का उपयोग करके हम static प्रॉपर्टी को एक्सेस कर रहे हैं।

### static के फायदे:

1. **ऑब्जेक्ट-निर्भरता से स्वतंत्र:** static प्रॉपर्टीज़ और मेथड्स क्लास के ऑब्जेक्ट से स्वतंत्र होते हैं, जिससे हम बिना ऑब्जेक्ट बनाए इनका उपयोग कर सकते हैं।
2. **वैल्यू बनाए रखना:** फंक्शन में static वेरिएबल्स के उपयोग से हम उस वेरिएबल की वैल्यू को हर कॉल के बाद बनाए रख सकते हैं।
3. **मेमोरी उपयोग में सुधार:** static प्रॉपर्टीज़ क्लास के ऑब्जेक्ट्स के बीच साझा होती हैं, इसलिए मेमोरी का उपयोग कम होता है।

## Q.12 Write Various decision making statement.

PHP में निर्णय लेने के स्टेटमेंट्स (Decision Making Statements) का उपयोग प्रोग्राम में विभिन्न शर्तों के आधार पर निर्णय लेने के लिए किया जाता है। PHP में मुख्य रूप से चार प्रकार के निर्णय लेने वाले स्टेटमेंट्स होते हैं:

1. **if statement**
2. **if-else statement**

### 3. else if statement

### 4. switch statement

आइए इन स्टेटमेंट्स के बारे में विस्तार से समझते हैं और उदाहरणों के साथ देखते हैं:

#### 1 if statement (if स्टेटमेंट)

if स्टेटमेंट का उपयोग तब किया जाता है जब हम किसी शर्त की जाँच करते हैं और यदि वह शर्त सही (true) होती है तो एक निश्चित कोड ब्लॉक को निष्पादित करते हैं।

##### Example:

```
<?php
$x = 10;

if ($x > 5) {
    echo "x is greater than 5";
}
?>
```

##### समझाइए:

- यहाँ पर यदि \$x की वैल्यू 5 से बड़ी होती है (जो कि है), तो "x is greater than 5" प्रिंट होगा।

#### 2 if-else statement (if-else स्टेटमेंट)

if-else स्टेटमेंट का उपयोग तब किया जाता है जब हमें दो स्थितियों में से एक का चुनाव करना होता है। यदि शर्त सही होती है, तो if ब्लॉक चलता है, और यदि शर्त गलत होती है, तो else ब्लॉक निष्पादित होता है।

##### Example:

```
<?php
$x = 3;

if ($x > 5) {
    echo "x is greater than 5";
} else {
    echo "x is not greater than 5";
}
?>
```

##### समझाइए:

- यहाँ पर शर्त \$x > 5 गलत है, इसलिए "x is not greater than 5" प्रिंट होगा।

#### 3 else if statement (else if स्टेटमेंट)

else if स्टेटमेंट का उपयोग तब किया जाता है जब हमें कई शर्तों की जाँच करनी होती है। अगर पहली शर्त गलत होती है, तो दूसरी शर्त को जाँचने के लिए else if का उपयोग किया जाता है।

##### Example:

```
<?php
$x = 10;
```

```

if ($x > 15) {
    echo "x is greater than 15";
} elseif ($x > 5) {
    echo "x is greater than 5 but less than or equal to 15";
} else {
    echo "x is less than or equal to 5";
}
?>

```

#### समझाइए:

- यहाँ पर पहली शर्त गलत है (\$x > 15), लेकिन दूसरी शर्त सही है (\$x > 5), तो "x is greater than 5 but less than or equal to 15" प्रिंट होगा।

## 4 switch statement (switch स्टेटमेंट)

switch स्टेटमेंट का उपयोग तब किया जाता है जब हमें एक ही वैरिएबल के विभिन्न संभावित मानों के आधार पर निर्णय लेना हो। यह एक अधिक सुसंगत और बेहतर तरीका है, जब हम कई विकल्पों की जाँच कर रहे हों।

#### Example:

```

<?php
$day = 3;

switch ($day) {
    case 1:
        echo "Today is Monday";
        break;
    case 2:
        echo "Today is Tuesday";
        break;
    case 3:
        echo "Today is Wednesday";
        break;
    case 4:
        echo "Today is Thursday";
        break;
    case 5:
        echo "Today is Friday";
        break;
    case 6:
        echo "Today is Saturday";
        break;
    case 7:
        echo "Today is Sunday";
        break;
    default:
        echo "Invalid day";
}
?>

```

#### समझाइए:

- यहाँ पर \$day की वैल्यू 3 है, तो "Today is Wednesday" प्रिंट होगा।
- switch स्टेटमेंट में हम केसों का मिलान करते हैं और हर केस के बाद break स्टेटमेंट का उपयोग करते हैं ताकि हम उस केस के बाद बाकी के केसों को न चेक करें।

## Q.13 Write note on wordpress.

**WordPress** एक ओपन-सोर्स कंटेंट मैनेजमेंट सिस्टम (CMS) है, जो वेबसाइट्स और ब्लॉग्स बनाने के लिए सबसे लोकप्रिय प्लेटफॉर्म में से एक है। इसे PHP और MySQL का उपयोग करके विकसित किया गया है और यह उपयोगकर्ताओं को बिना किसी कोडिंग ज्ञान के वेबसाइट बनाने और उसे प्रबंधित करने की सुविधा देता है। WordPress को इसके उपयोगकर्ता-मित्र इंटरफ़ेस, अनुकूलन की क्षमता, और विस्तृत प्लगइन और थीम समर्थन के कारण पसंद किया जाता है।

### WordPress के मुख्य विशेषताएँ (Features)

#### 1. ओपन सोर्स सॉफ्टवेयर:

- WordPress पूरी तरह से निःशुल्क है और इसका कोड ओपन सोर्स है। इसका मतलब है कि कोई भी उपयोगकर्ता इसे डाउनलोड, उपयोग, और कस्टमाइज़ कर सकता है।

#### 2. कस्टमाइजेशन की सुविधा:

- WordPress उपयोगकर्ताओं को बहुत सारी **थीम्स** और **प्लगइन्स** प्रदान करता है, जिनकी मदद से वे अपनी वेबसाइट को डिज़ाइन और कार्यात्मक रूप से कस्टमाइज़ कर सकते हैं।

#### 3. इजी टू यूज़ इंटरफ़ेस:

- WordPress का डैशबोर्ड यूज़र-फ्रेंडली होता है और इसके माध्यम से वेबसाइट के सभी पहलुओं को आसानी से एडिट और मैनेज किया जा सकता है।

#### 4. प्लगइन्स और एक्सटेंशन्स:

- WordPress में लाखों प्लगइन्स होते हैं जो साइट की कार्यक्षमता को बढ़ाते हैं, जैसे SEO, सिक््योरिटी, सोशल मीडिया इंटीग्रेशन, और अधिक।

#### 5. SEO फ्रेंडली:

- WordPress SEO को ध्यान में रखते हुए डिज़ाइन किया गया है। इसमें SEO प्लगइन्स जैसे कि **Yoast SEO** या **All in One SEO Pack** आसानी से इंस्टॉल किए जा सकते हैं, जो वेबसाइट की रैंकिंग को बेहतर बनाने में मदद करते हैं।

#### 6. मोबाइल रिस्पॉन्सिव:

- अधिकांश WordPress थीम्स मोबाइल और टैबलेट पर भी अच्छी तरह से काम करती हैं, जिससे वेबसाइट का उपयोग कहीं से भी किया जा सकता है।

#### 7. मल्टी-यूज़र सपोर्ट:

- WordPress एक मल्टी-यूज़र प्लेटफॉर्म है, जिससे कई लोग एक वेबसाइट पर विभिन्न प्रकार की भूमिकाओं (जैसे एडमिन, एडिटर, लेखक, और उपयोगकर्ता) के साथ काम कर सकते हैं।

#### 8. मल्टीसाइट नेटवर्क:

- WordPress मल्टीसाइट फीचर का समर्थन करता है, जिसका मतलब है कि आप एक ही इंस्टॉलेशन के भीतर कई वेबसाइट्स बना सकते हैं और उन्हें प्रबंधित कर सकते हैं।

## 9. सुरक्षा:

- WordPress की सुरक्षा को ध्यान में रखते हुए कई सुरक्षा प्लगइन्स उपलब्ध हैं जैसे **Wordfence** और **Sucuri**। इसके अलावा, नियमित रूप से अपडेट किए जाने वाले वर्डप्रेस संस्करण और सुरक्षा पैच वेबसाइट को सुरक्षित रखने में मदद करते हैं।

## WordPress की स्थापना (Installation)

WordPress को स्थापित करना बहुत आसान है, और इसके लिए निम्नलिखित कदमों का पालन किया जा सकता है:

1. **वेब होस्टिंग और डोमेन खरीदें:** सबसे पहले, आपको एक वेब होस्टिंग सेवा और एक डोमेन नाम की आवश्यकता होगी।
2. **WordPress डाउनलोड करें:** WordPress की आधिकारिक वेबसाइट ([wordpress.org](https://wordpress.org)) से नवीनतम संस्करण डाउनलोड करें।
3. **इंस्टॉलेशन:** डाउनलोड की गई फाइल को अपने वेब सर्वर पर अपलोड करें और इंस्टॉलेशन प्रक्रिया का पालन करें। अधिकांश होस्टिंग प्रदाता **One-click installation** की सुविधा भी देते हैं।
4. **थीम और प्लगइन्स इंस्टॉल करें:** वेबसाइट के डिज़ाइन और कार्यक्षमता को बढ़ाने के लिए एक उपयुक्त थीम और आवश्यक प्लगइन्स इंस्टॉल करें।
5. **कस्टमाइजेशन और कंटेंट जोड़ें:** वेबसाइट को अपनी जरूरतों के अनुसार कस्टमाइज करें और पोस्ट, पेज, और अन्य कंटेंट जोड़ें।

## WordPress का उपयोग कहाँ किया जाता है?

WordPress का उपयोग विभिन्न प्रकार की वेबसाइट्स बनाने के लिए किया जा सकता है, जैसे:

- **ब्लॉग्स:** व्यक्तिगत या पेशेवर ब्लॉग्स को बनाने के लिए।
- **कॉर्पोरेट वेबसाइट्स:** छोटे से लेकर बड़े बिजनेस के लिए वेबसाइट्स।
- **ई-कॉमर्स साइट्स:** WooCommerce जैसे प्लगइन्स के साथ ऑनलाइन स्टोर बनाना।
- **फोरम:** विभिन्न विषयों पर चर्चा के लिए फोरम वेबसाइट्स।
- **पोर्टफोलियो वेबसाइट्स:** व्यक्तिगत या पेशेवर पोर्टफोलियो वेबसाइट्स।

## WordPress के फायदे (Advantages of WordPress)

1. **उपयोग में आसानी:** WordPress का इंटरफ़ेस बहुत सरल है और इसे सीखने में ज्यादा समय नहीं लगता।
2. **खुला स्रोत और मुफ्त:** WordPress एक ओपन-सोर्स प्लेटफॉर्म है और इसका उपयोग बिल्कुल मुफ्त है।
3. **कस्टमाइजेशन:** आपके पास वेबसाइट के हर पहलू को कस्टमाइज़ करने की पूरी स्वतंत्रता है।
4. **SEO फ्रेंडली:** WordPress पर वेबसाइट्स को SEO फ्रेंडली बनाने के लिए कई टूल्स और प्लगइन्स हैं।

5. **समर्थन और समुदाय:** WordPress का एक बड़ा और सक्रिय समुदाय है, जिससे आपको समस्या के समाधान के लिए बहुत सारे संसाधन और समर्थन मिल सकते हैं।
6. **सुरक्षा:** नियमित अपडेट और सुरक्षा प्लगइन्स WordPress साइट्स को सुरक्षित रखने में मदद करते हैं।

### WordPress के नुकसान (Disadvantages of WordPress)

1. **स्पीड:** अगर बहुत सारे प्लगइन्स का उपयोग किया जाता है, तो वेबसाइट की गति में कमी आ सकती है।
2. **सुरक्षा खतरे:** चूंकि WordPress बहुत लोकप्रिय है, यह हैकर्स का एक प्रमुख निशाना बनता है। नियमित रूप से अपडेट और सुरक्षा उपायों की आवश्यकता होती है।
3. **थीम्स और प्लगइन्स की गुणवत्ता:** सभी थीम्स और प्लगइन्स उच्च गुणवत्ता के नहीं होते, जिससे कभी-कभी प्रदर्शन या सुरक्षा समस्याएं उत्पन्न हो सकती हैं।

## Q.14 Write a PHP code to display date and time.

PHP में वर्तमान तिथि और समय को प्रदर्शित करने के लिए हम `date()` फंक्शन का उपयोग करते हैं। यह फंक्शन तिथि और समय को विभिन्न प्रारूपों में दिखाने की अनुमति देता है।

यहाँ एक साधारण PHP कोड है जो वर्तमान तिथि और समय को प्रदर्शित करता है:

```
<?php
// वर्तमान तिथि और समय को प्रदर्शित करें
echo "Current Date and Time: " . date("Y-m-d H:i:s");
?>
```

### समझाइए:

- `date("Y-m-d H:i:s")` में:
  - Y वर्ष को 4 अंकों में दिखाता है (जैसे 2025)।
  - m महीने को 2 अंकों में दिखाता है (01 से 12)।
  - d दिन को 2 अंकों में दिखाता है (01 से 31)।
  - H घंटे को 24-घंटे के प्रारूप में दिखाता है (00 से 23)।
  - i मिनट को 2 अंकों में दिखाता है (00 से 59)।
  - s सेकंड को 2 अंकों में दिखाता है (00 से 59)।

### आउटपुट:

Current Date and Time: 2025-04-26 14:30:45



यह कोड वर्तमान तिथि और समय को Y-m-d H:i:s प्रारूप में प्रदर्शित करेगा। आप इसमें किसी भी प्रकार का कस्टम प्रारूप भी सेट कर सकते हैं, जैसे d/m/Y या l, F j, Y आदि, जो आपके आवश्यकतानुसार तिथि और समय को प्रदर्शित करेंगे।

अगर आप इसे किसी विशेष समय क्षेत्र में सेट करना चाहते हैं, तो आप `date_default_timezone_set()` फंक्शन का उपयोग कर सकते हैं।

उदाहरण:

```
<?php
// समय क्षेत्र सेट करें
date_default_timezone_set("Asia/Kolkata");

// वर्तमान तिथि और समय को प्रदर्शित करें
echo "Current Date and Time: " . date("Y-m-d H:i:s");
?>
```

यह कोड भारतीय समय क्षेत्र (IST) में वर्तमान तिथि और समय प्रदर्शित करेगा।

## Q.15 What is MYSQL ? write steps for connecting to MYSQL .

### MySQL क्या है?

MySQL एक ओपन-सोर्स रिलेशनल डेटाबेस मैनेजमेंट सिस्टम (RDBMS) है, जो डेटा को संरचित रूप में स्टोर करने और प्रबंधित करने के लिए प्रयोग में लाया जाता है। यह SQL (Structured Query Language) का उपयोग करके डेटा के साथ संवाद करता है। MySQL का उपयोग बहुत सारी वेबसाइटों और एप्लिकेशनों में किया जाता है, क्योंकि यह तेज़, विश्वसनीय, और आसानी से उपलब्ध है।

MySQL डेटाबेस का उपयोग डेटा को स्टोर, रीट्रीव, अपडेट और डिलीट करने के लिए किया जाता है। इसका उपयोग विभिन्न एप्लिकेशन जैसे वेब एप्लिकेशन, ई-कॉमर्स साइट्स, ब्लॉग्स, आदि में बड़े पैमाने पर किया जाता है।

### MySQL से कनेक्ट करने के चरण (Steps for Connecting to MySQL)

PHP के माध्यम से MySQL से कनेक्ट करने के लिए आपको निम्नलिखित स्टेप्स का पालन करना होगा:

#### 1. MySQL सर्वर सेटअप:

सबसे पहले आपको अपने सिस्टम में MySQL इंस्टॉल करना होगा (या अगर आप लोकल सर्वर (जैसे XAMPP या WAMP) का उपयोग कर रहे हैं तो वह पहले से इंस्टॉल हो सकता है)।

#### 2. MySQL कनेक्शन को सेट करें:

PHP में MySQL से कनेक्ट करने के लिए, `mysqli` या `PDO` (PHP Data Objects) का उपयोग किया जाता है। नीचे हम `mysqli` का उपयोग करके कनेक्शन बनाने के चरणों पर चर्चा करेंगे।

## 1. MySQLi (MySQL Improved) का उपयोग करके कनेक्ट करना

### Step 1: PHP स्क्रिप्ट में MySQLi का कनेक्शन स्थापित करें

```
<?php
// MySQL सर्वर, यूज़रनेम, पासवर्ड और डेटाबेस का नाम
$servername = "localhost"; // MySQL सर्वर (आमतौर पर localhost)
$username = "root"; // MySQL यूज़रनेम
$password = ""; // MySQL पासवर्ड (आमतौर पर XAMPP में खाली होता है)
$dbname = "testdb"; // आपका डेटाबेस नाम

// MySQLi कनेक्शन
$conn = new mysqli($servername, $username, $password, $dbname);

// कनेक्शन की जाँच करें
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

### समझाइए:

- \$servername में हम MySQL सर्वर का पता देते हैं। लोकलहोस्ट (localhost) आमतौर पर सर्वर का पता होता है।
- \$username और \$password MySQL डेटाबेस से कनेक्ट होने के लिए उपयोगकर्ता नाम और पासवर्ड होते हैं।
- \$dbname वह डेटाबेस है जिससे आप कनेक्ट होना चाहते हैं।
- new mysqli(\$servername, \$username, \$password, \$dbname) द्वारा कनेक्शन स्थापित किया जाता है।

यदि कनेक्शन सफल होता है तो "Connected successfully" प्रदर्शित होगा। यदि कनेक्शन में कोई समस्या आती है तो connect\_error द्वारा त्रुटि संदेश दिखाई देगा।

## 2. PDO (PHP Data Objects) का उपयोग करके कनेक्ट करना

PDO एक और तरीका है MySQL से कनेक्ट करने के लिए। यह अधिक सुरक्षित और लचीला तरीका है, क्योंकि यह विभिन्न डेटाबेस के साथ काम कर सकता है, न केवल MySQL के साथ।

### Step 1: PDO का उपयोग करके कनेक्शन स्थापित करें

```
<?php
$servername = "localhost"; // MySQL सर्वर
$username = "root"; // MySQL यूज़रनेम
$password = ""; // MySQL पासवर्ड
$dbname = "testdb"; // डेटाबेस का नाम

try {
    // PDO कनेक्शन बनाएं
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
    $password);

    // कनेक्शन को सेट करें
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "Connected successfully";
}
```

```

}
catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>

```

### समझाइए:

- new PDO("mysql:host=\$servername;dbname=\$dbname", \$username, \$password) से हम MySQL सर्वर से कनेक्ट करते हैं।
- setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION) द्वारा हम एरर मोड को सेट करते हैं ताकि अगर कोई समस्या हो तो हमें एक exception (त्रुटि) प्राप्त हो।
- यदि कनेक्शन सफल होता है तो "Connected successfully" दिखाई देगा, अन्यथा कनेक्शन से संबंधित त्रुटि का संदेश प्रदर्शित होगा।

## MySQL कनेक्शन से डेटा प्राप्त करना

जब आप MySQL से कनेक्ट हो जाते हैं, तो आप SQL क्वेरी का उपयोग करके डेटा प्राप्त कर सकते हैं। उदाहरण के लिए, एक सरल SELECT क्वेरी:

```

<?php
// कनेक्शन (माइएसक्यूएल का उदाहरण)
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "testdb";

// कनेक्ट करें
$conn = new mysqli($servername, $username, $password, $dbname);

// कनेक्शन की जाँच करें
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// SELECT क्वेरी
$sql = "SELECT id, name, email FROM users";
$result = $conn->query($sql);

// डेटा को प्रदर्शित करें
if ($result->num_rows > 0) {
    // आउटपुट डेटा
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["name"]. " - Email: " .
        $row["email"]. "<br>";
    }
} else {
    echo "0 results";
}

$conn->close(); // कनेक्शन को बंद करें
?>

```

यह कोड users टेबल से id, name और email कॉलम को SELECT करता है और उसे प्रदर्शित करता है।

