



Introduction to PHP

- *PHP (Hypertext Preprocessor) is a popular open-source server-side scripting language.*
- *It is widely used for web development to create dynamic web pages and applications.*
- *PHP scripts are executed on the server, and the result is sent to the client's browser as plain HTML.*



Sanjeev Kumar
@prosanjeev



Sanjeev Kumar
@prosanjeev

👉 Keep Swiping...

Basic Syntax

- *PHP code is embedded in HTML using <?php ... ?> tags.*
- *Example:*



```
<!DOCTYPE html>
<html>
<body>
    <h1><?php echo "Hello, World!"; ?></h1>
</body>
</html>
```



Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Variables

- *Variables in PHP are declared with a \$ sign, followed by the variable name.*
- *Example:*

```
$name = "John";  
$age = 30;  
$isStudent = true;
```

```
<?php  
$greeting = "Hello, World!";  
echo $greeting;  
?>
```



Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Data Types

- Common data types: String, Integer, Float, Boolean, Array, Object, NULL.

```
<?php
$string = "Hello";
$integer = 123;
$float = 12.34;
$boolean = true;
$array = array("Apple", "Banana", "Cherry");
$object = (object) ['property' => 'value'];
>null = NULL;
?>
```

Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

Strings and String Functions

- *Strings are sequences of characters, enclosed in quotes.*
- *Concatenate strings using the `.` operator.*

```
● ● ●  
<?php  
$firstName = "John";  
$lastName = "Doe";  
$fullName = $firstName . " " . $lastName;  
echo $fullName;  
?>
```

- *Common string functions: `strlen()`, `strtolower()`, `strtoupper()`, `substr()`.*

```
● ● ●  
  
$str = "Hello, World!";  
echo strlen($str); // Output: 13  
echo strtoupper($str); // Output: HELLO, WORLD!
```

Sanjeev Kumar
@prosanjeev

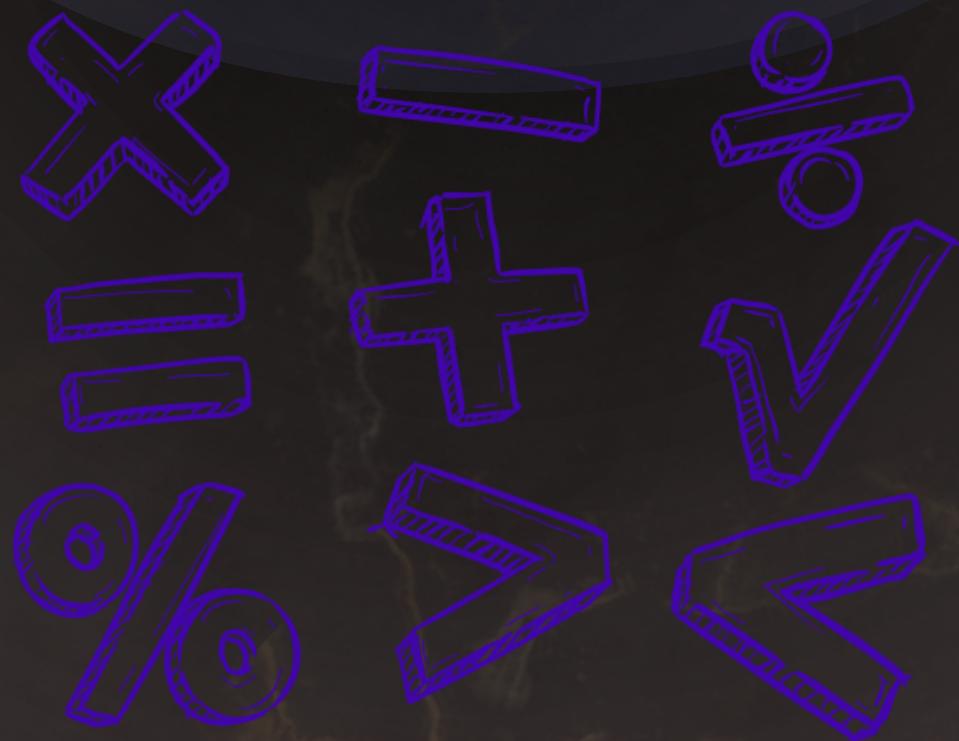
➔ Keep Swiping...

Operators

- PHP supports arithmetic (+, -, *, /), assignment (=), comparison (==, !=, >, <), and logical operators (&&, ||, !).



```
$str = "Hello, World!";
echo strlen($str); // Output: 13
echo strtoupper($str); // Output: HELLO, WORLD!
```



Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Control Structures

- *If-Else:*

```
● ● ●  
if ($age > 18) {  
    echo "Adult";  
} else {  
    echo "Not an adult";  
}
```

- *Switch:*

```
● ● ●  
$color = "red";  
switch ($color) {  
    case "red":  
        echo "Color is red";  
        break;  
    case "blue":  
        echo "Color is blue";  
        break;  
    default:  
        echo "Color not recognized";  
}
```

Sanjeev Kumar
@prosanjeev

👉 Keep Swiping...

Loops

- *For Loop:*

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

- *While Loop:*

```
$i = 0;  
while ($i < 10) {  
    echo $i;  
    $i++;  
}
```

- *Foreach Loop:*

```
$colors = array("red", "green", "blue");  
foreach ($colors as $color) {  
    echo $color;  
}
```



Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Functions

- *Functions in PHP are defined using the **function** keyword.*



```
<?php
function greet($name) {
    return "Hello, " . $name;
}

echo greet("John"); // Outputs: Hello, John
?>
```

f(x)

Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

Arrays

- PHP supports both indexed and associative arrays.

-Indexed array:



```
$fruits = array("apple", "banana", "orange");
echo $fruits[0]; // Output: apple
```

-Associative array:



```
$age = array("John" => 30, "Jane" => 25);
echo $age["John"]; // Output: 30
```

Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

Superglobals

- *Superglobals are predefined variables in PHP, accessible from anywhere in the script.*
- *Examples: `$_GET`, `$_POST`, `$_SESSION`, `$_COOKIE`, `$_SERVER`, `$_FILES`.*

-Example usage:

```
// $_GET example: Assuming the URL is example.com?name=John
echo $_GET['name']; // Outputs: John
```

```
// $_SERVER example: Outputs the server name
echo $_SERVER['SERVER_NAME']; // e.g., localhost
```

```
// $_COOKIE example: Assuming a cookie named 'user' is set
echo $_COOKIE['user']; // Outputs the value of the 'user' cookie
```

Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Form Handling

- *Forms are used to collect user input and send it to the server.*

-Example form:

```
● ● ●  
  
<form method="post" action="process.php">  
    Name: <input type="text" name="name">  
    <input type="submit">  
</form>
```

-Handling form data in PHP:

```
● ● ●  
  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = $_POST['name'];  
    echo "Hello, " . $name;  
}
```

Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

File Handling

- PHP provides functions for creating, reading, writing, and deleting files.

-Example of reading a file:



```
$file = fopen("example.txt", "r");
echo fread($file, filesize("example.txt"));
fclose($file);
```

-Example of write a file:



```
$file = fopen("example.txt", "w");
fwrite($file, "This is the content to be written.");
fclose($file);
```



Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

Sessions and Cookies

- Sessions store user information across multiple pages.



```
session_start();
$_SESSION['username'] = "JohnDoe";
echo $_SESSION['username'];
```

- Cookies are used to store small amounts of data on the client's machine.



```
setcookie("user", "JohnDoe", time() + (86400 * 30), "/");
echo $_COOKIE['user'];
```

Sanjeev Kumar
@prosanjeev

➔ Keep Swiping...

Error Handling

- *Error handling in PHP can be managed using try-catch blocks.*

-Example:

```
<?php
try {
    $file = fopen("example.txt", "r");
    if (!$file) {
        throw new Exception("File could not be opened.");
    }
    echo fread($file, filesize("example.txt"));
    fclose($file);
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

Sanjeev Kumar
@prosanjeev

→ Keep Swiping...

Database Interaction

- PHP can connect to databases using extensions like MySQLi or PDO.

-Example:

```
$conn = new mysqli($servername, $username, $password, $dbname);
$result = $conn->query("SELECT * FROM users");
while ($row = $result->fetch_assoc()) {
    echo $row['username'];
}
$conn->close();
```

Security Considerations

- Sanitize and validate user input to prevent SQL injection.
- Use prepared statements for database queries.
- Hash passwords using functions like password_hash().



Sanjeev Kumar
@prosanjeev

That's it.

Conclusion

- PHP is a powerful language for server-side web development.
- It integrates seamlessly with HTML and databases.
- Understanding PHP basics and best practices is crucial for building dynamic and secure web applications.

