

Unit - II

2.1 Introduction: What is PHP?

PHP ஒரு Open Source scripting languageஆகும். இதை dynamic web applicationகளை உருவாக்க பயன்படுத்தப்படுகிறது. PHPஇன் syntax C, Java and Perl languageகளை சார்ந்து இருக்கும். PHPயை Rasmus Lerdorf 1994ல் C programming Language மூலமாக உருவாக்கினார், இதனை PHP 2 என்று அழைத்தனர். 1998, PHP 3 மக்களின் பயன்பாட்டிற்காக வெளியிடப்பட்டது, இந்த version PHPஆனது பரவலாக அனைவராலும் அதிகமாக உபயோகப்படுத்தப்பட்ட PHP versionஆகும்.

PHPஇன் முழுமை PHP:Hypertext Preprocessorஆகும். PHP scriptஆனது serverல் executeஆகும் அதன் resultஐ வெறும் HTMLஆக clientக்கு அனுப்பும். PHP script “.php” எனும் extensionல் save செய்யப்படுகிறது. ஒரு PHP fileலில் HTML tags, CSS elements, JavaScriptகள் PHP Scriptஉடன் இடம்பெறலாம்.

Basic Syntax of PHP

PHP scriptஆனது PHP script tagஐ கொண்டு மூடப்பட்டிருக்கும்.

Syntax:

```
<?php
    php script
?>
```

Example:

```
<?php
    //Single Line Comment
    /*frist Program
```

explain

syntax of PHP*/

```
echo "Hello World";
```

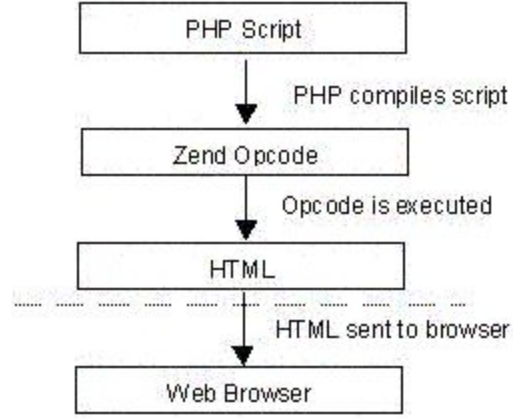
```
#echo "First Program";
```

?>

Programming In Web Environment

PHP web severல் இடம்பெற்றிருக்க வேண்டும் அப்பொழுதுதான் அதனால் ஒரு scriptஐ interpret செய்து அதற்கு தகுந்தாற்போல் HTML pageஐ உருவாக்கமுடியும். PHPன் மத்திய மையமாக இருப்பது அதன் language engineஆகும், இதன் பெயர் “zend”. PHP என்பது ஒரு முழுமையான அமைப்பை குறிக்கும். PHPல் Zend Engine compiler மற்றும் runtime engineஆக பயன்படுத்தப்படுகிறது. PHP Scriptகள் memoryல் ஏற்றப்பட்டு Zend Engine மூலமாக Zend Opcodeகளாக மாற்றப்படுகிறது. இந்த opcodeகள் execute செய்யும்பொழுது HTML உருவாக்கப்பட்டு client systemதிற்கு அனுப்பப்படுகிறது

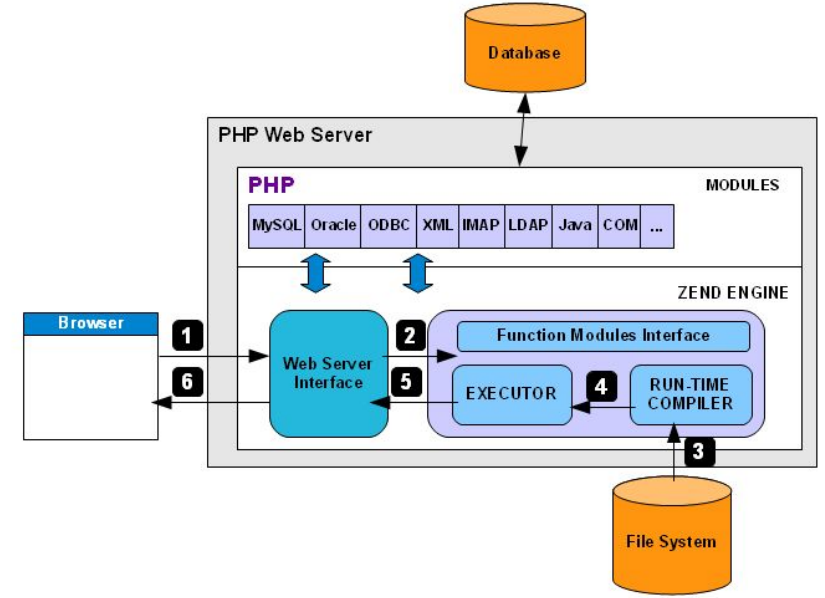
Zend engineன் பெயர் அதன் developerகளான Zeev and Aandi ஆகியோரின் பெயரிலிருந்து உருவாக்கப்பட்டது. இந்த engine நம்பகத்தன்மை, அதிக செயல்திறன் மற்றும் அதன் அம்சங்களை நீடிக்கப்படக்கூடிய வகையில் உருவாக்கப்பட்டது.



ஒரு web serverல் PHP scriptஐ execute செய்ய மூன்று கூறுகள் தேவை, அவை

1. ஒரு interpreter PHP Scriptஐ ஆய்வு செய்வதற்கு
2. PHPன் செயல்பாடுகளை செயல்படுத்த interpreterல் உள்ள ஒரு functionality section.
3. Web serverஉடன் தொடர்புகொள்வதற்காக ஒரு Interface section.

மேல்கூறப்பட்ட கூறுகளில் 1 மற்றும் 2ன் பகுதியை Zend Engine கையாளும். 2ஆம் கூறின் மூதிப்பகுதி மற்றும் 3ஐ PHP கையாளும்.



PHP Server Engine Architecture

Step 1). Browserல் இருந்து ஒரு scriptஐ execute செய்ய கோரிக்கை மேற்கொள்ளப்படுகிறது.

Step 2). இந்த கோரிக்கையை Web Server Interface முறைப்படுத்தி file systemதிற்கு அனுப்புகிறது.

Step 3). கோரிக்கை விடுக்கப்பட்ட script file systemதில் இருந்து runtime compilerல் compile செய்வதற்காக load செய்யப்படுகிறது. இதனை Zend engine opcodeஆக execution செய்வதற்கு ஏற்றாற்போல் parse செய்கிறது. Opcode உருவாக்கும் பொழுது அந்த script execute செய்வதற்கு தேவையான moduleகளை zend engine சேர்க்கிறது.

Step 4). Executor உருவாக்கப்பட்ட opcodeஐ execute செய்து HTML fileஐ உருவாக்குகிறது.

Step 5). உருவாக்கப்பட்ட HTML file web server interfaceக்கு அனுப்பப்படுகிறது.

Step 6). Web Server Interface இந்த HTML fileஐ browserக்கு அனுப்புகிறது.

Benifits of PHP

- No Execution in browser - serverல் script execute செய்யப்பட்டு அதன் resultஆன HTML page மட்டுமே browserக்கு அனுப்பப்படுகிறது.
- Code is Invisible - PHP Scriptஆனது serverல் மட்டுமே பார்க்க முடியும், browserல் பார்க்க முடியாது. Script serverல் execute அவதால் browserல் பார்க்க முடியாது.
- Browser Independent - PHPல் ஒரு script serverல் execute ஆனபிறகு resultஆக HTML page மட்டுமே browserக்கு அனுப்பப்படுகாதல் இது எந்த ஒரு பிரௌசரையும் சார்ந்து இருக்காது.
- No binary Code is generated - PHPல் binary code (exe) உருவாக்கப்படமாட்டாது. இதில் HTML file மட்டுமே outputஆக உருவாக்கப்படும்.
- JavaScript can be Embedded - ஒரு PHP scriptல் JavaScriptஐ நம்மால் சேர்க்கமுடியும். இதனால் PHP script executeஆவதில் எந்த ஒரு பாதிப்பும் வராது.

Drawbacks of PHP

- Need Server - ஒரு PHP scriptஐ execute செய்வதற்கு ஒரு server தேவைப்படுகிறது. server இல்லாமல் PHP scriptஐ execute செய்ய முடியாது.

- Need Server Resource - PHP ஒரு scripting language என்பதால் ஒரு script எத்தனை முறை browserல் இருந்து அழைக்கப்பட்டாலும் அத்தனை முறையும் execute செய்யப்படுகிறது. இதனால் serverன் resourceகள் அதிகமாக தேவைப்படுகிறது.

Common PHP Script Elements

PHP script elements என்பது PHP server execute செய்யும் elementகள் ஆகும். அதிகமாக உபயோகப்படுத்தப்படும் PHP script elementகள் கீழே கொடுக்கப்பட்டுள்ளது

- PHP script tag
- Comments
- Statements
- Escape Sequence

PHP Script Tag

இந்த tag PHP server interpret செய்யவேண்டிய PHP scriptகளை கொண்டிருக்கும்.

Syntax:

```
<?php  
.....  
.....  
?>
```

Comments

Comment என்பது ஒரு programmer எத்துப்பட்டிரும் program பற்றிய விவரத்தை தெரிவிப்பதற்கு பயன்படுகிறது. Comment tagல் கொடுக்கப்பட்ட text ஆனது PHP serverல் execute ஆகாது. PHP comment இரண்டு வகைப்படும்

1. Single Line Comment - இது ஒரு வரியில் comment எழுத பயன்படும். இதற்கு double slash (//) அல்லது hash (#) பயன்படுத்தப்படுகிறது.
2. Multiline Comment - இது பல வரியில் comment எழுத பயன்படும். இதற்கு (/*....*/) syntax பயன்படுத்தப்படுகிறது.

Statements

ஒரு PHP script tagஇன் உள்ளே எதனை PHP statement வேண்டுமானாலும் எழுதலாம். எழுதப்பட்ட அனைத்து statementம் semi-colon(:) கொண்டு முடிவடையவேண்டும். ஒரே ஒரு PHP statement மட்டுமே ஏறுகிறது என்றால் அதற்கு semi-colon கொடுக்கவேண்டுமென்று அவசியம் இல்லை. .

Example:

```
<?php echo("hai") ?> - semi-colon not necessary
```

```
<?php
echo("Hai");
echo("Welcome");
?>
```

Semicolon Must

Escape Sequence

Escape sequence என்பது PHP scriptஐ சார்ந்த special characterகளை echo statementல் உபயோகிக்க பயன்படுவதாகும். உதாரணத்திற்கு echo statementல் ஒரு stringஐ display செய்வதற்கு double

quotes (“) உபயோகப்படுத்தப்படுகிறது, இந்த double quotesஐ display செய்வதற்கு முதலில் PHP interpreterக்கு அதனை process செய்யாமல் display செய்யவேண்டும் என்பதை தெரியப்படுத்தவேண்டு. இதற்காக backslash (\) பயன்படுத்தப்படுகிறது இதனை escape sequence என்று அழைக்கப்படுகிறது. PHPயின் சில முக்கியமான escape sequence characterகள் கீழே கொடுக்கப்பட்டுள்ளது

- Double Quotes (“ ”)
- Single Quotes (‘ ’)
- New Line Character (\n)
- Tab Space (\t)
- Print \$ (\\$)
- Print Backslash (\\)

Example

```
<?php
#Escape Sequences
echo "\"Hello World\" \n\t First PHP Program";
?>
```

Using Variables

Variable PHP scriptல் மதிப்புகளை சேமிக்க பயன்படுத்தப்படுகிறது. PHP scriptல் ஒரு variableஐ உபயோகிக்க சில விதிமுறைகள் உள்ளன. இந்த விதிகள் naming rules என்று அழைக்கப்படுகிறது

Naming Rules

1. அனைத்து variable பெயரும் \$ symbolஐ கொண்டு ஆரம்பிக்கப்படவேண்டும்.

2. அடுத்து இருக்க வேண்டிய எழுத்து ஒரு alphabet(a-z)ஆக இருக்கவேண்டும்.
3. Variable பெயரில் இடைவெளி இருக்கக்கூடாது.
4. Variable பெயரில் சிறப்பு எழுத்தாக underscore(_) மட்டுமே பயன்படுத்த முடியும்.
5. Variable பெயர்கள் PHPல் case sensitive ஆகும், Upper caseல் எழுதப்பட்ட பெயரும் lower caseல் எழுதப்பட்ட பெயரும் வெவ்வேறு ஆகும்.
6. ஒரு variableன் data type அதில் சேமிக்கப்பட்டிருக்கும் மதிப்பை பொறுத்து இருக்கும்.
7. ஒரு variable அதன் பயன்பாட்டிற்கு முன்னதாக declare செய்யவேண்டியது இல்லை.

Example:

```
<?php
#PHP Variables
$aVal_ = 10;
//b = 20; //Wrong Assignment
echo "Variable a=".$aVal_;
//echo "Case Sensitive $aval";
$aVal_="Hello World";
echo "Recent Value: $aVal_";
?>
```

Constants

ஒரு மதிப்பை குறிப்பதற்காக பயன்படுத்தப்படும் பெயர் constant எனப்படும். ஒரு constantன் மதிப்பை PHP script execute ஆகும் பொழுது மற்றம் செய்ய இயலாது.

Syntax:

```
define("const_name",value);
```

Variable பெயர் மற்றும் constant பெயரை வேறுபடுத்துவதற்காக சில விதிகள் பயன்படுத்தப்படுகிறது.

Rules:

1. Constant பெயர் முழுவதுமாக Upper caseல் கொடுக்கப்படவேண்டும்.
2. Constant பெயர் alphabet(a-z) அல்லது underscore(_) மட்டுமே தொடங்க முடியும்.
3. ஒரு constantஐ உருவாக்க Define function பயன்படுத்தப்படுகிறது.
4. ஒரு constantல் உள்ள மதிப்பை உபயோகிப்பதற்கு அதன் பெயர் அல்லது constant function பயன்படுத்தப்படுகிறது.

Example:

```
<?php
define("PI",3.141);
echo "PI Value = ".PI;
echo "PI Value = ".constant("PI");
?>
```

Data Types

ஒரு programன் அடிப்படை கட்டமைப்பிற்கு Data Types பக்கபலமாக உள்ளது. ஒரு variableல் எந்த விதமான data சேமிக்கப்படவேண்டும் என்பதை Data Types வரையறுக்கிறது. அதேபோல் அந்த dataவை சேமிக்க தேவைப்படும் memoryஐ காட்டுகிறது. PHPல் உள்ள Data Typeகள் கீழே கொடுக்கப்பட்டுள்ளது.

- String - இது எழுத்துக்களின் கூட்டு ஆகும். String valueகள் double quotes (“ ”) அல்லது single quotes (‘ ’) உள்ளே கொடுக்கப்படும்.

Example:

```
<?php
$str = "Hello World";
$str1 = "Welcome To TNPT";
echo "<b>String Value:</b> ".$str;
echo $str1;
//echo "<h4>String</h4><br>$str";
?>
```

- Integer - இது முழுமையான எண்ணை சேமிக்க பயன்படுகிறது. ஒரு குறிப்பிட்ட மதிப்பை தாண்டும் பொழுது இது தானாக floatஆக மாறிவிடும்.

Example:

```
<?php
$intvar = 34;
echo "<br><b>Integer Value:</b> ".$intvar;
?>
```

- Float - இது பின்ன எண்கள் மற்றும் அதிக மதிப்புடைய முழு எண்களை சேமிக்க பயன்படுகிறது.

Example:

```
<?php
$fltvar = 1.243;
echo "<br><b>Float Value:</b> ".$fltvar;
?>
```

- Boolean - இது TRUE அல்லது FALSE மதிப்புகளை சேமிக்க பயன்படுகிறது. இதில் 0 FALSEஐயும், 1 TRUEவையும் குறிக்கும்.

Example:

```
<?php
$bool = TRUE;
echo "<br><b>Boolaen Value: </b>".$bool;
?>
```

- NULL - இது ஒரு தனிவகையான data typeஆகும் இது ஒரு variableலில் எந்தஒரு மதிப்பும் இல்லை என்பதை குறிக்கும். NULLன் மதிப்பும் zero(0) மதிப்பும் ஒன்றுஅல்ல.

Example:

```
<?php
$empty = NULL;
echo "Vlaue is ".$empty;
?>
```

- Arrays - இது ஒரே data typeல் பல மதிப்புகளை சேமிக்க பயன்படுகிறது. PHPல் ஒரு array உருவாக்க array() function பயன்படுத்தப்படுகிறது.

Example:

```
<?php
```

```
$numarray = array(1,2,3,4);
echo "2nd Array Element: ".$numarray[1];
?>
```

- Object - Objects என்பது userகாலால் வரையறுக்கப்பட்ட ஒரு data typeஆகும். இவற்றில் தனிப்பட்ட variable மற்றும் function கொடுக்கப்பட்டிருக்கும்.

Example:

```
<?php
class sample
{
function sample()
{
$this->val = 10;
}
}
$obj = new sample();
echo "Value in Object: ".$obj->val;
?>
```

- Resources - இது ஒரு சிறப்புமிக்க variableஆகும், இதில் ஒரு fileன் reference அல்லது ஒரு databaseன் tableஐ சேமிக்க முடியும்.

Operators

Operatorகள் variable மற்றும் மதிப்புகளில் மாற்றங்கள் செய்ய மற்றும் ஒப்பிட பயன்படுகிறது. PHP உள்ள Operatorகள்.

- Arithmetic Operators
- Assignment Operators
- Comparison Operators

- Increment/Decrement Operatos
- Logical Operators
- Conditional Operator

Arithmetic Operators

Arithmetic operatorகள் கணித செயல்பாடுகளை செய்ய பயன்படுகிறது.

Symbol	Operator Name	Description
+	Addition	add two values
-	Subtraction	subtract two values
*	Multiplication	multiplies two values
/	Division	divides two values
%	Modulus	return remainder of dividing two values

Assignment Operators

PHP scriptல் variableக்கு மதிப்பை நியமிக்க Assignment Operatorகள் பயன்படுத்தப்படுகிறது. Variable operatorன்(=) இடதுபுறத்தில் மதிப்பு operatorன் வலதுபுறத்தில் கொடுக்கப்பட்டிருக்கும்.

Symbol	Operator Name	Description
=	Equal	assign value from right to left side variable

+=	Add and	adds value on right to left and store in left side variable
*=	Multiply and	multiply value on right to left and store in left side variable
/=	Divide and	divide value on right to left and store in left side variable
%=	Modulus and	return remainder of value on right to left and store in left side variable

Comparison Operators

இரண்டு variableன் மதிப்புகளை ஒப்பிடுவதற்கு Comparison Operatorகள் பயன்படுத்தப்படுகிறது.

Symbol	Operator Name	Description
==	Double Equal	check if two values are same
!=	Not equal to	check if two values are not same
>	Greater than	check if left value is greater than right value
<	less than	check if left value is less than right value
<=	less than equal to	check if left value is less than or equal to right value
>=	greater than equal to	check if left value is greater than or equal to right value

Increment/Decrement Operators

D. NATARAJASIVAN/TNPT

Increment/Decrement operatorகள் ஒரு variableல் உள்ள மதிப்பை கூடவோ குறைக்கவோ பயன்படுகிறது. இந்த operator பயன்படுத்தப்படும் இடத்தை பொறுத்து pre அல்லது post என்று இரண்டு வகைப்படும்.

Symbol	Operator Name	Description
++var	Pre-Increment	Increments the value and then assigns the value
var++	Post-Increment	value is assigned and then it is incremented
--var	Pred-Decrement	Decrement the value and then assigns the value
var--	Post-Decrement	Value is assigned and then it is decremented

Logical Operators

Logical Operatorகள் conditional statementல் இரண்டு conditionகளை இணைத்து ஒரு முடிவை எடுப்பதற்கு பயன்படுகிறது.

Symbol	Operator Name	Description
and, &&	and operator	return true only if both the conditions are true
or,	or operator	returns true if any one of the two condition is true
!	not operator	invers the result of a condition
xor	exclusive or	return true only if both conditions are

	operator	not same
--	----------	----------

Conditional Operator

Conditional Operator ஒரு ternary operator ஆகும். இதில் மூன்று operandகள் இருக்கும்.

syntax

```
var = operand1?operand2:operand3;
```

operand1 - இது ஒரு conditionஆகும், இதுவே முதலில் execute செய்யப்படுகிறது.

operand2 - Condition TRUEஆக இருந்தால் இது execute செய்யப்படும்.

operand3 - Condition FALSEஆக இருந்தால் இது execute செய்யப்படும்.

example

```
<?php
$mark=65;
$res = ($mark>45)?"Pass":"Fail";
echo $res;
?>
```

Statements

PHP script வாக்கியங்களால் உருவாக்கப்பட்டது. Program வாக்கியம் ஒரு குறிப்பிட்ட செயலை செய்வதற்கு பயன்படுத்தப்படுகிறது.

PHPல் Statementகள் இரண்டு விதமாக வகைப்படுத்தப்படுகிறது

1. Conditional Statements
2. Looping Statements

Conditional Statements

Conditional Statementகள் கொடுக்கப்பட்டுள்ள நிபந்தனையை பொறுத்து program controlஐ மாற்றம் செய்கிறது. இதில் ஒரு condition சரிபார்க்கப்படுகிறது அந்த conditionன் விளைவு TRUEஆக இருந்தால் ஒரு குறிப்பிட்ட statementகளின் தொகுப்பு execute செய்யப்படுகிறது அல்லது அதன் விளைவு FALSEஆக இருந்தால் வேறு statementகளின் தொகுப்பு execute செய்யப்படுகிறது.

PHPல் மூன்று விதமான conditional statementகள் உள்ளன அவை

1. if...else statement
2. if...else if statement
3. switch statement

if...else statement

If...else ஒரு இருவழி condition statement ஆகும். இது conditionன் முடிவை பொறுத்து TRUE பகுதி அல்லது FALSE பகுதியை தேர்ந்தெடுக்கும்.

syntax:

```
if(condition)
{
    True Part
}
else
{
    False Part
}
```

example:

```
<?php
$mark = 39;
```

```

if($mark>50)
{
    echo "<B style='color: GREEN;'>Pass</B>";
}
else
{
    echo "<B style='color: RED;'>Fail</B>";
}
?>

```

if...else if statement

If...else if statement ஒன்றன் பின் ஒன்றாக பல conditionகளை சரிபார்க்க உபயோகிக்கப்படுகிறது. இதில் ஒரு condition FALSEஆக இருந்தால் else பகுதியில் உள்ள conditionன் முடிவை பார்க்கிறது.

syntax:

```

if(condition 1)
{
    True Part
}
else if(condition 2)
{
    True Part
}
else
{
    False Part
}

```

example:

```

<?php
$mark = 84;
if($mark>=80)
{
    echo "<b>A Grade</b>";
}
else if($mark>=70 and $mark<=79)
{
    echo "<b>B Grade</b>";
}
else if($mark>=60 && $mark<=69)
{
    echo "<b>C Grade</b>";
}
else if($mark>=50 and $mark<=59)
{
    echo "<b>D Grade</b>";
}
else
{
    echo "<b>Fail</b>";
}
?>

```

Switch Statement

Switch statement மதிப்பை பொறுத்து ஒருகுறிப்பிட்ட statementஐ execute செய்யும் இதனை multiway branching statement என்றும் அழைக்கப்படும். PHP switch statementல் எண் மற்றும் stringஐ மதிப்பாக பயன்படுத்த முடியும்.

syntax:

```

switch(option)

```

```

{
case value1:
    statement;
    break;
case value2:
    statement;
    break;
case valueN:
    statement;
    break;
default:
    statement;
    break;
}

```

Example:

```

<?php
$mark = 77;
$opt = (int)($mark/10);
switch($opt)
{
case 10:
    echo "<b>S Grade</b>";
    break;
case 9:
    echo "<b>A Grade</b>";
    break;
case 8:
    echo "<b>B Grade</b>";
    break;
case 7:
    echo "<b>C Grade</b>";
    break;
case 6:
    echo "<b>D Grade</b>";

```

```

break;
case 5:
    echo "<b>E Grade</b>";
    break;
default:
    echo "<b>Fail</b>";
    break;
}
?>

```

Looping Statement

Looping statement குறிப்பிட்ட statementகளின் தொகுப்பை மீண்டும் மீண்டும் execute செய்வதற்கு பயன்படுத்தப்படுகிறது. இது குறிப்பிடப்பட்டுள்ள condition TRUEஆக இருக்கும்வரை செயல்படும். PHPல் இரண்டு விதமான looping statementகள் இருக்கின்றது, அவை

1. Condition based loop
2. Iterator loop

Condition Based Loop

Condition based loopகளில் loopன் execution ஒரு conditionஐ பொறுத்தே இருக்கும். கொடுக்கப்பட்ட condition TRUEஆக இருந்தால் loop executeஆகும் இல்லையேல் loop execute ஆகாது. Condition கொடுக்கப்பட்டுள்ள இடத்தை பொறுத்து இதனை இரண்டு பிரிவுகளாக வகைப்படுத்தலாம், அவை

- Entry Contolled Loop
- Exit Controlled Loop

Entry Controlled Loop

இந்த வகையான loopல் condition loopன் ஆரம்பத்தில் கொடுக்கப்பட்டிருக்கும். கொடுக்கப்பட்ட condition TRUEஆக இருந்தால்

மட்டுமே loop statements executeஆகும்.. PHPல் while loop மற்றும் for loop இந்த வகையை சேர்ந்ததாகும்

while Loop

While loop கொடுக்கப்பட்டுள்ள condition TRUEஆக இருக்கும்வரை அதனுள் இருக்கும் statementகளை execute செய்யும். ஒரு loopன் executionஐ கட்டுப்படுத்த loop control variable பயன்படுத்தப்படுகிறது. இந்த variable while loop ஆரம்பிப்பதற்கு முன்னதாகவே கொடுக்கப்பட்டிருக்கும்.. இதில் Condition loopன் ஆரம்பத்தில் சரிபார்க்கப்படும், loop control variableன் மதிப்பை கூடவோ/குறைக்கவோ பயன்படுத்தும் statement loopன் உள்ளே கொடுக்கப்பட்டிருக்கும்.

syntax

```
initialization
while(condition)
{
    Loop body
    Increment/Decrement
}
```

example

```
<?php
$val = 3;
while($val>0)
{
    echo "<h$val>Level $val</h$val>";
    $val--;
}
?>
```

for loop

For loopல் initialization, condition மற்றும் increment/decrement பகுதிகள் ஒரே இடத்தில் கொடுக்கப்பட்டிருக்கும். இது ஒரு programஐ தெளிவாக பிழை இல்லாமல் எழுத பயன்படும்.

syntax

```
for(initialization;condition;increment/decrement)
{
    Loop Body
}
```

Example

```
<?php
$mul = 3;
$stable = "<table border='2'>";
for($i=1;$i<=5;$i++)
{
    $res = $mul * $i;
    $stable.="<tr><td>$mul x $i = </td><td> $res</td></tr>";
}
$stable.="</table>";
echo $stable;
?>
```

ஒரு for loopல் initialization, condition and increment/decrement பகுதிகள் இடம்பெற்றாமல் இருந்தால் அது ஒரு முடிவில்லா loop(infinite loop) என்று அழைக்கப்படும்

syntax

```
for(;;)
{
    Loop Body
}
```

Iterator Loop

PHPல் ஒரு பிரத்யேகமான loop பயன்படுத்தப்படுகிறது இதனை Iterator Loop ஆகும். இந்த loop ஒரு array அல்லது objectல் உள்ள அணைத்து மதிப்புகளையும் ஒன்றன்பின் ஒன்றாக எடுக்க பயன்படுத்தப்படுகிறது..

syntax

```
foreach(element as itme)
{
    Loop body
}
```

Example

```
<?php
$numarr = array(1,2,3,4,5,6,7,8,9,10);
$sum = 0;
foreach($numarr as $val)
{
    echo $val."<br>";
    $sum += $val;
}
echo "<br><b>Sum of the Array is:". $sum."</b>";
?>
```

Working with Arrays

Array என்பது ஒரு data structure ஆகும் இது ஒரே data typeல் பல மதிப்புகளை ஒரு variable பெயரில் சேமிக்க பயன்படுகிறது. PHPல் மூன்று வகையான array உபயோகிக்கப்படுகிறது, அவை

- Numeric array
- Associative array
- Multidimensional array

Numeric Array

PHPல் arrayஐ உபயோகிக்க பயன்படுத்தப்படும் indexன் மதிப்பை பொறுத்து இந்த array numeric array என்று அழைக்கப்படுகிறது. இந்த arrayல் எண்கள் indexஆக உபயோகப்படுத்தப்பட்டு அந்த arrayல் உள்ள மதிப்புகள் access செய்யப்படுகிறது. இதன் index மதிப்பு 0 ஆரம்பித்து size-1 முடிகிறது, இதில் size arrayன் size ஆகும்.

PHPல் ஒரு array இந்த மூன்று விதங்களில் உருவாக்கப்படுகிறது

1. using []
2. using index value
3. using array function

using []

இந்த முறையில் arrayன் அளவு syntaxல் கொடுக்கப்படமாட்டாது. இதில் arrayன் பெயர் மற்றும் அதில் சேமிக்கப்படவேண்டிய மதிப்புகள் square bracketன் உள்ளே கொடுக்கப்படும்.

syntax

```
$array_name = [values];
```

example

```
$rollno = [101,102,103];
```

using index value

இந்த முறையில் arrayல் மதிப்புகள் ஒன்றன்பின் ஒன்றாக சேர்க்கப்படுகிறது. இதில் arrayன் பெயர் மற்றும் index valueவை பயன்படுத்தி மதிப்புகள் சேமிக்கப்படுகிறது.

syntax

```
$array_name[index] = value;
```

example

```
$rollno[0] = 101;
$rollno[1] = 102;
```

```
$rollno[2] = 103;
```

using array function

இதுவே PHPல் array உருவாக்க பயன்படுத்தப்படும் பாதுகாப்பான முறையாகும். இதில் array() functionஐ பயன்படுத்தி array உருவாக்கப்படுகிறது.

syntax

```
$array_name = array(values);
```

example

```
$names = array("Anbu","Charan","Karthick");
```

Accessing Arrays

PHPல் ஒரு array மதிப்பை பயன்படுத்த இரண்டு முறைகள் உள்ளன அவை

1. Using index
2. Using Iterator Loop

using index

இந்த முறையில் எண் indexஐ பயன்படுத்தி arrayன் மதிப்பு உபயோகிக்கப்படுகிறது.

syntax

```
$variable = $arrayname[index];
```

using iterator loop

ஒரு arrayல் உள்ள மதிப்புகளை ஒன்றன்பின் ஒன்றாக உபயோகிப்பதற்கு, foreach iterator loop பயன்படுத்தப்படுகிறது. இந்த loop arrayல் உள்ள அனைத்து மதிப்புகளும் உபயோகிக்கும்வரை செயல்படும்.

syntax

```
foreach($array as $value)
{
    Loop Body
}
```

Numeric Array Example

```
<html>
```

D. NATARAJASIVAN/TNPT

```
<head>
```

```
<title>Arrays</title>
```

```
</head>
```

```
<body>
```

```
<h3>Numeric Arrays</h3>
```

```
<?php
```

```
//Using []
```

```
$marks = [89,92,73];
```

```
//Using Array Index
```

```
$marks[3] = 77;
```

```
$marks[4] = 85;
```

```
$marks[5] = 95;
```

```
//Using Array Function
```

```
$subj = array("OS","BEEE","C","Linux Lab","BEEE Lab","C  
Programming");
```

```
?>
```

```
<p>Accessing Using Index Value</p>
```

```
<table border="1">
```

```
<tr><th>Subject</th><th>Mark</th></tr>
```

```
<?php
```

```
for($i=0;$i<6;$i++)
```

```
{
```

```
    echo "<tr><td>".$subj[$i]."</td><td>".$marks[$i]."</td></tr>";
```

```
}
```

```
//Calculating Total
```

```
$tot = 0;
```

```
foreach($marks as $subj_mark)
```

```
{
```

```
    $tot+=$subj_mark;
```

```
}
```

```
echo "<tr><td>Total</td><td><b>".$tot."</b></td></tr>";
```

```
?>
```

```
</table>
```

```
</body>
```

</html>

Output

Numeric Arrays

Accessing Using Index Value

Subject	Mark
OS	89
BEEE	92
C	73
Linux Lab	77
BEEE Lab	85
C Programming	95
Total	511

Numeric Array Functions

PHPல் ஒரு arrayல் பயன்படுத்தக்கூடிய பல functionகள் உள்ளன. இந்த functionகள் ஒரு குறிப்பிட்ட செயலை செய்யும். சில முக்கியமான array functionகள் கீழே கொடுக்கப்பட்டுள்ளன

Basic Array Functions

- count - ஒரு arrayல் உள்ள மதிப்புகள் எத்தனை என்று எண்ணி சொல்லும்
count(\$array_name);
- unset - Arrayல் ஒரு மதிப்பை நீக்க பயன்படும். இதில் index value உபயோகிக்கப்படுகிறது.
unset(\$array[index]);
- reset - இது array pointerஐ arrayன் முதல் உறுப்புக்கு கொண்டுசெல்லும்.
reset(\$array);
- end - இது array pointerஐ arrayன் கடைசி உறுப்புக்கு கொண்டுசெல்லும்
end(\$array);

Additional Array Functions

- array_push - ஒரு arrayன் முடிவில் மதிப்பை சேர்ப்பதற்கு பயன்படும்
array_push(\$array,value);
- array_pop - ஒரு arrayன் முடிவில் இருக்கும் மதிப்பை நீக்குவதற்கு பயன்படும்
array_pop(\$array);
- array_unshift - ஒரு arrayல் ஆரம்பத்தில் மதிப்பை சேர்ப்பதற்கு பயன்படுகிறது
array_unshift(\$array,value);
- array_shift - ஒரு arrayன் ஆரம்பத்தில் இருந்து மதிப்பை நீக்குவதற்கு பயன்படுகிறது
array_shift(\$array);
- array_merge - இரண்டு arrayகளை ஒன்றுசேர்ப்பதற்கு இது பயன்படுகிறது
array_merge(\$array1,\$array2);
- sort - ஒரு arrayஐ ஏறு வரிசையில் வரிசைப்படுத்துவதற்கு பயன்படுகிறது
sort(\$array);
- rsort - ஒரு arrayஐ இறங்கு வரிசையில் வரிசைப்படுத்துவதற்கு பயன்படுகிறது
rsort(\$array);

Associative Array

இந்த array numeric array போன்றே இயங்கும் இதில் உள்ள ஒரே வித்தியாசம் இதில் numeric index பதிலாக string index பயன்படுத்தப்படுகிறது. இது key value pair association என்று அழைக்கப்படுகிறது.

PHPல் ஒரு array இந்த மூன்று விதங்களில் உருவாக்கப்படுகிறது

1. using []
2. using index value
3. using array function

using []

இந்த முறையில் arrayன் அளவு syntaxல் கொடுக்கப்படமாட்டாது. இதில் arrayன் பெயர் மற்றும் அதில் சேமிக்கப்படவேண்டிய மதிப்புகள் “key=>value” pair முறையில் square bracketன் உள்ளே கொடுக்கப்படும்.

syntax

```
$array = ["key1"=>"value", "key2"=>"value", "key3"=>"value"];
```

Example

```
$cse_stud = ["01"=>"Anbu", "02"=>"Bala", "03"=>"Guna"];
```

using index value

இந்த முறையில் arrayல் மதிப்புகள் ஒன்றன்பின் ஒன்றாக சேர்க்கப்படுகிறது. இதில் arrayன் பெயர் மற்றும் “string” index valueவை பயன்படுத்தி மதிப்புகள் சேமிக்கப்படுகிறது.

syntax

```
$array_name["index"] = value;
```

example

```
$cse_stud["01"] = "Anbu";  
$cse_stud["02"] = "Bala";  
$cse_stud["03"] = "Guna";
```

using array function

இதுவே PHPல் array உருவாக்க பயன்படுத்தப்படும் பாதுகாப்பான முறையாகும். இதில் array() functionஐ பயன்படுத்தி array உருவாக்கப்படுகிறது.

syntax

```
$array = array("Key1"=>"Value", "Key2"=>"Value");
```

Example

```
$cse_stud = array("01"=>"Anbu", "02"=>"Bala", "03"=>"Guna");
```

Accessing Arrays

PHPல் ஒரு array மதிப்பை பயன்படுத்த இரண்டு முறைகள் உள்ளன அவை

1. Using string index
2. Using Iterator Loop

using index

இந்த முறையில் string indexஐ பயன்படுத்தி arrayன் மதிப்பு உபயோகிக்கப்படுகிறது.

syntax

```
$variable = $arrayname["string_index"];
```

using iterator loop

ஒரு arrayல் உள்ள மதிப்புகளை ஒன்றன்பின் ஒன்றாக உபயோகிப்பதற்கு, foreach iterator loop பயன்படுத்தப்படுகிறது. இந்த loop arrayல் உள்ள அனைத்து மதிப்புகளும் உபயோகிக்கும்வரை செயல்படும்.

syntax

```
foreach($array as $key=>$value)  
{  
    Loop Body  
}
```

Associative Array Example

```
<html>  
<head>  
    <title>Arrays</title>
```



```

</head>
<body>
  <h2>Associative Arrays</h2>
  <?php
    //Using []
    $studDCE = ["01"=>"Anbu","02"=>"Bala"];

    //Using Array Index
    $studDEEEE["01"] = "Arun";
    $studDEEEE["02"] = "Mani";

    //Using Array Function
    $dept = array("101"=>"DCE","102"=>"DEEE");
  ?>
  <h2>Student Details</h2>
  <?php
    echo "<h3>".$dept["101"]."</h3>";
    foreach($studDCE as $roll=>$name)//Using Iterator Loop
    {
        echo "<br>Roll: ".$roll;
        echo "<br>Name: ".$name;
    }
    echo "<h3>".$dept["102"]."</h3>";
    foreach($studDEEEE as $roll=>$name)
    {
        echo "<br>Roll: ".$roll;
        echo "<br>Name: ".$name;
    }
  ?>
</body>
</html>

```

Output

Associative Arrays

Student Details

DCE

Roll: 01
Name: Anbu
Roll: 02
Name: Bala

DEEE

Roll: 01
Name: Arun
Roll: 02
Name: Mani

Associative Array Functions

PHPல் ஒரு arrayல் பயன்படுத்தக்கூடிய பல functionகள் உள்ளன. இந்த functionகள் ஒரு குறிப்பிட்ட செயலை செய்யும். மூன்று முக்கியமான associative array functionகள் கீழே கொடுக்கப்பட்டுள்ளன

- `array_key_exists` - இந்த function ஒரு arrayல் நாம் குறிப்பிட்டிருக்கும் string index ஏற்கனவே இடம்பெற்று இருக்கிறதா என்று சரிபார்க்கும். அந்த string index இடம்பெற்றிருந்தால் TRUE இல்லை என்றால் FALSE result ஆக தரும்.
`array_key_exists("index",$array);`
- `array_keys` - இந்த function arrayல் உள்ள key மதிப்புகளை மட்டும் பிரித்து எடுக்க பயன்படுகிறது.
`array_keys($array);`
- `array_values` - இந்த function arrayல் உள்ள valueகளை மட்டும் பிரித்து எடுக்க பயன்படுகிறது.

`array_values($array);`

Multidimensional Array

Multidimensional array என்பது ஒரு arrayன் உள்ளே ஒவ்வொரு indexலும் மற்றொரு array இடம்பெற்றிருக்கும். பொதுவாக multidimensional array என்பது arrayகளின் array ஆகும். ஒரு multidimensional arrayஐ உள்ள மதிப்பை பயன்படுத்த ஒன்றுக்கும் மேற்பட்ட index value பயன்படுத்தப்படுகிறது. Multidimensional array உருவாக்க சிறந்த முறை array() functionஐ பயன்படுத்துவதாகும்.

Two dimensional array ஒரு எளிய two dimensional array ஆகும்.

syntax:

```
$array = array( array(elements),array(elements));
```

example:

```
$array = array(array(1,2,3),array(4,5,6));
```

Accessing Arrays

இந்த arrayஐ பயன்படுத்துவதற்கு இரண்டு முறைகள் உள்ளன

1. Using index
2. Using Iterator Loop

using index

இதில் இரண்டு index valueகளை கொண்டு அந்த array பயன்படுத்தப்படுகிறது. இந்த இரண்டு indexகளும் row மற்றும் column என்று அழைக்கப்படுகிறது.

syntax

```
$variable = $arrayname[row][column];
```

using iterator loop

ஒரு arrayல் உள்ள மதிப்புகளை ஒன்றன்பின் ஒன்றாக உபயோகிப்பதற்கு, foreach iterator loop பயன்படுத்தப்படுகிறது. இந்த loop arrayல் உள்ள அனைத்து மதிப்புகளும் உபயோகிக்கும்வரை செயல்படும். ஒரு multidimensional arrayல் உள்ள மதிப்புகளை பயன்படுத்த nested foreach loop பயன்படுத்தப்படுகிறது.

syntax

```
foreach($array as $row)
{
    foreach($row as $val)
    {
        Loop Body
    }
}
```

Example

```
<head>
  <title>Arrays</title>
</head>
<body>
  <h2>Multi Dimensional Array</h2>
  <?php
    //Using array() function
    $ary = array(array(1,2,3,4),array(5,6,7,8));
    $val = $ary[1][1];
    echo "<br> 1Row 2Col Value: $val";
  ?>
  <p>
    <b>Using Nested Foreach Loop</b><br>
    <?php
      foreach($ary as $row)
      {
```

```

foreach($row as $val)
{
    echo " $val";
}
echo "<br>";
}
?>
</p>
</body>
</html>

```

Output

Multi Dimensional Array

1Row 2Col Value: 6

Using Nested Foreach Loop

```

1 2 3 4
5 6 7 8

```

Using Functions

ஒரு function என்பது statementகளின் தொகுப்பு இதை ஒரு programல் எத்தனை முறை வேண்டுமானாலும் அழைக்கலாம். Functionல் மதிப்புகளை அனுப்புவதற்கு parameterகள் பயன்படுத்தப்படுகிறது. PHPல் ஒரு function execute செய்வதற்கு அதன் பெயர் உபயோகிக்கப்படுகிறது.

PHPல் ஒரு functionஐ சரியாக execute செய்வதற்கு இரண்டு பகுதிகள் தேவைப்படும், அவை

1. Creating a PHP function
2. Calling a PHP function

Creating a PHP function

PHPல் ஒரு execute செய்வதற்கு முன்னர் அதனை உருவாக்கவேண்டும் . ஒரு function உருவாக்க நான்கு பகுதிகள் தேவைப்படுகிறது.

1. சரியான function பெயர்
2. function பெயர் முன்னதாக “function” keyword கொடுக்கப்படவேண்டும்
3. function பெயருக்கு பிறகு parameter list கொடுக்கப்படவேண்டும்
4. Function statement தொகுப்பு curly braces இடையே கொடுக்கப்பட்டிருக்கும்.

syntax

```

function Function_Name(argument list)
{
    Function Code block;
}

```

Calling a PHP function

ஒரு function உருவாக்கப்பட்டபிறகு அதனை execute செய்வதற்கு அந்த function பெயர் மற்றும் அதில் உள்ள parameter listஐ கொண்டு அழைக்கப்படுகிறது.

syntax

```

Function_Name(argument list);

```

Example

```

<html>
<head>
    <title>Function</title>
</head>

```

```

<body>
  <h2>Simple Example</h2>
  <?php
    //Creating function
    function display()
    {
      echo "Welcome";
    }
  ?>
  <h4><?php display(); ?></h4>
  <h3><?php display(); ?></h3>
</body>
</html>

```

Output

Simple Example

Welcome

Welcome

Function Types

ஒரு functionல் உபயோகப்படுத்தப்படும் parameterகள் மற்றும் அந்த function திரும்ப தரும் மதிப்பை பொறுத்து ஐந்து வகைப்படும், அவை

1. No Parameter
2. With Parameter
3. Call by Reference
4. Default Value
5. Return Value

No Parameter

இந்த வகையான functionல் parameterகள் கொடுக்கப்பட்டிருக்காது.

syntax

```

function function_name()
{
  Code
}
function call
function_name();

```

With Parameter

இந்த வகையான functionல் parameterகள் கொடுக்கப்பட்டிருக்கும்.

syntax

```

function function_name($para1, $para2)
{
  Code
}
function call
function_name($para1, $para2);

```

Example

```

<?php
$a = 10;
$b = 20;
//Function with Arguments
function swap($x, $y)
{
  $temp = $x;
  $x = $y;
  $y = $temp;
  echo "<label>After Swapping</label>";
  echo "<br>a = ".$x;

```

```
echo "<br>b = ".$y;
}
```

```
//Calling Function
swap($a,$b);
```

?>

Call by reference

இந்த வகையான functionல் மதிப்புகள் functionக்கு referenceஆக அனுப்பப்படுகிறது. ஒரு மதிப்பை functionக்கு referenceஆக அனுப்பும்பொழுது அந்த மதிப்புக்கு functionல் செய்யப்படும் மாற்றங்கள் அந்த functionஐ அழைக்கப்பட்ட இடத்தில் பிரதிபலிக்கும். ஒரு parameter functionக்கு referenceஆக அனுப்ப ampersand (&) அந்த variable பெயருடன் சேர்க்கப்படுகிறது.

syntax

```
function function_name(&$para1, &$para2)
{
    Code
}
```

function call

```
function_name($para1, $para2);
```

example

```
<?php
```

```
$a = 10;
$b = 20;
echo "<label>Before Swapping</label>";
echo "<br>a = ".$a;
echo "<br>b = ".$b;
swap($a,$b);
echo "<br><label>After Swapping</label>";
echo "<br>a = ".$a;
echo "<br>b = ".$b;
```

```
//Call by reference
function swap(&$x, &$y)
{
    $temp = $x;
    $x = $y;
    $y = $temp;
}
```

?>

Default Value

இந்த வகையான functionல், parameterகளுக்கு defaultஆன மதிப்பு கொடுக்கப்பட்டிருக்கும். இந்த functionஐ அழைக்கும் பொழுது parameterகளுக்கு மதிப்பு கொடுக்கப்படவில்லை என்றால் defaultஆக கொடுக்கப்பட்ட மதிப்பு எடுத்துக்கொள்ளப்படும்.

syntax

```
function function_name($para1 = value)
{
    Code
}
```

function call

```
function_name($para1);
```

(or)

```
function_name();//இந்த function call default ஆக
```

கொடுக்கப்பட்டிருக்கும் மதிப்பை எடுத்துக்கொள்ளும்

example

```
<?php
```

```
//Default Value
function AddImg($img = "noimg.jpg")
{
    echo '';
```

```

}
AddImg("01.jpg");
AddImg("02.png");
AddImg();
?>

```

Return Value

ஒரு functionல் இருந்து மதிப்பை திரும்ப பெறுவதற்கு **return** statement பயன்படுத்தப்படுகிறது. இந்த functionல் return statementஐ அடைந்தவுடன் function செயல்பாடு நிறுத்தப்பட்டு, மதிப்பை function call செய்யப்பட்ட இடத்திற்கு திரும்ப அனுப்புகிறது. இந்த முறையில் ஒரு மதிப்பை மட்டுமே திரும்ப அனுப்ப முடியும்.

syntax

```

function function_name(parameter List)
{
    Code
    return $val;
}
function call
$variable = function_name(parameter list);

```

Example

```

<?php
function fun($x,$y,$z)
{
    $val = 0;
    while($x && $y && $z)
    {
        $x--;$y--;$z--;
        $val++;
    }
    return $val;
}

```

```

}
$res = fun(10,5,8);
echo "<br>Result: ".$res;
?>

```

OOP

Object Oriented Programming என்பது நிஜ உலகில் உள்ள செயல்பாட்டை மாத்திரையகக்கொண்டு software உருவாகும் அணுகுமுறையாகும். PHP object oriented முறைகள் PHP5ல் அறிமுகப்படுத்தப்பட்டது, இது மிகவும் கடினமான மற்றும் மறுபயன்பாடு செய்யக்கூடிய web applicationகளை உருவாக்க பயன்படுகிறது. PHPல் உள்ள மிகமுக்கியமான Object Oriented conceptகள்

- Class
- Member Variable
- Member Function
- Object
- Inheritance

Class

Class என்பது userல் வரையறுக்கப்பட்ட data type ஆகும், இதில் variable மற்றும் function இடம்பெற்றிருக்கும். ஒரு class என்பது ஒரு class என்பது அதைப்போலவே பிரதிகள்(object) உருவாக்கப்படுகிறது.

Member Variable

இந்த variableகள் ஒரு classன் உள்ளே கொடுக்கப்பட்டிருக்கும். இந்த variableகள் private அல்லது protectedஆக கொடுக்கப்பட்டிருந்தால் அதனை வேறுஎங்கும் access செய்ய முடியாது.

Member Function

இந்த functionகள் ஒரு classன் உள்ளே வரையறுக்கப்பட்டிருக்கும், இது ஒரு objectல் உள்ள மதிப்பை access செய்ய பயன்படுகிறது.

Object

Object என்பது ஒரு classன் பிரதியாகும். ஒரு classஐ பயன்படுத்தி ஒன்றிற்கும் மேற்பட்ட objectகளை உருவாக்கலாம்.

Syntax

```
class class_name
{
    var $variable1;
    var $variable2;
    function function_name(parameter list)
    {
        Function Code
    }
}
```

Object creation

```
$obj = new class_name;
      (or)
$obj = new class_name();
```

Member variables மற்றும் member functionகளை ஒரு classக்கு வெளியே access செய்வதற்கு access operator (->) பயன்படுகிறது.

syntax

```
$obj->variable;
$obj->function_name();
```

Member variables மற்றும் member functionகளை ஒரு classக்கு உள்ளே access செய்வதற்கு \$this keyword மற்றும் access operator (->) பயன்படுகிறது.

syntax

```
$this->variable;
$this->function_name();
```

Example

```
<?php
class student
{
    var $roll;
    var $name;
    function set_data($r,$n)
    {
        $this->roll = $r;
        $this->name = $n;
    }
    function get_data()
    {
        echo "<br>Roll No: ".$this->roll;
        echo "<br>Name: ".$this->name;
    }
}
```

```
$stu1 = new student;
$stu2 = new student();
echo "<br> Two Objects created";
```

```
$stu1->set_data(101,"Anbu");
$stu2->set_data(102,"Bala");
```

```
$stu1->get_data();
```

```
$stu2->get_data();  
?>
```

Output

```
Two Objects created  
Roll No: 101  
Name: Anbu  
Roll No: 102  
Name: Bala
```

Inheritance

ஒரு classல் உள்ள member variables மற்றும் member functionகளை வேறு ஒரு classல் பயன்படுத்துவது inheritance ஆகும். Base class என்பது அதில் உள்ள member variables மாற்று member functionகளை வேறு ஒரு classல் பயன்படுத்த அனுமதிக்கிறது. பகிரப்பட்ட member variables மாற்று member functionகளை உபயோகிக்கும் class derived class எனப்படுகிறது. Derived classல் அதற்கு சொந்தமான function மற்றும் variableகளும் இருக்கும். PHPல் இதை அமல்படுத்த “extends” keyword பயன்படுத்தப்படுகிறது.

syntax

```
class base  
{  
    var $basevariable;  
    function base_function(parameter_list)  
    {  
        Function Code  
    }  
}  
class derived extends base  
{  
    var $derivedvariable;
```

```
function derived_function(parameter_list)  
{  
    Function Code  
}  
}
```

Example

```
<?php  
//Base Class  
class student  
{  
    var $roll;  
    var $name;  
    function set_data($r,$n)  
    {  
        $this->roll = $r;  
        $this->name = $n;  
    }  
    function get_data()  
    {  
        echo "<br>Roll No: ".$this->roll;  
        echo "<br>Name: ".$this->name;  
    }  
}  
//Derived Class  
class alumni extends student  
{  
    var $year_pass;  
    function set_yearpass($yr)  
    {  
        $this->year_pass = $yr;  
    }  
    function get_yearpass()  
    {
```



```

        echo "<br>Year of Pass: ".$this->year_pass;
    }
}

$stu1 = new student;
$stu2 = new alumni;
echo "<br> Two Objects created";

$stu1->set_data(101,"Anbu");
$stu2->set_data(102,"Bala");
$stu2->set_yearpass(2017);

$stu1->get_data();
$stu2->get_data();
$stu2->get_yearpass();
?>

```

மேலே கொடுக்கப்பட்ட உதாரணத்தில் base class “student” கொடுக்கப்பட்ட விவரங்கள் derived class “alumni” பயன்படுத்தமுடியும்.

Access Modifiers

Object oriented programmingல் access modifierகள் member variable மற்றும் member function எந்த இடத்தில் access செய்யலாம் என்பதை குறிக்கும்.

PHPல் மூன்று Access Modifierகள் உள்ளன, அவை

1. Public
2. Private
3. Protected

Public

PHPல் ஒரு classன் memberகளுக்கு இதுவே default access specifier ஆகும். Public memberஐ scriptல் எங்கிருந்து வேண்டுமானாலும் access செய்யலாம்.

Example

```

<?php
class base
{
    var $x;
    function set_data($a)
    {
        $this->x = $a;
    }
    function get_data()
    {
        echo "<br>Value in X: ".$this->x;
    }
}
$obj = new base;
$obj->x = 10;
$obj->get_data();
?>

```

மேலே கொடுக்கப்பட்டுள்ள உதாரணத்தில் Member variable x மற்றும் member function get_data publicஆக கொடுக்கப்பட்டுள்ளது, இதனால் இவற்றை classக்கு வெளியே objectன் பெயரைகொண்டு access செய்யமுடியும்.

Private

ஒரு private memberஐ அது declare செய்யப்பட்டிருக்கும் classன் உள்ளே மட்டுமே access செய்யமுடியும், அந்த classக்கு வெளியே access செய்யமுடியாது.

Example

```

<?php
class base
{
    private $x;
}

```

```

function set_data($a)
{
    $this->x = $a;
}
function get_data()
{
    echo "<br>Value in X: ".$this->x;
}
}
$obj = new base;
//$obj->x = 20; // This access is not allowed
$obj->set_data(20);
$obj->get_data();
?>

```

மேலே கொடுக்கப்பட்டுள்ள உதாரணத்தில் classன் member variable privateஆக declare செய்யப்பட்டுள்ளது. இந்த scriptல் அந்த variableஐ classன் objectஐ கொண்டு access செய்யமுடியாது. இதனை classன் உள்ளே மட்டுமே access செய்யமுடியும்.

Protected

ஒரு protected member அது declare செய்யப்பட்டிருக்கும் class மற்றும் அதனை inherit செய்யும் classல் மட்டுமே access செய்யமுடியும். இதனை தவிர வேறுஎங்கும் access செய்யமுடியாது.

Example

```

<?php

class base
{
    protected $x;
}

class derived extends base
{

```

```

var $y;
function set_data($a,$b)
{
    $this->x = $a;
    $this->y = $b;
}
function get_data()
{
    echo "<br>Value in X: ".$this->x;
    echo "<br>Value in Y: ".$this->y;
}
}
$obj = new derived;
//$obj->x = 20;
$obj->set_data(20,10);
$obj->get_data();
?>

```

மேலே கொடுக்கப்பட்டுள்ள உதாரணத்தில் base classல் உள்ள member variable x privateஆக declare செய்யப்பட்டிருக்கிறது, இதனை அந்த class மற்றும் அதனை inherit செய்யும் classல் பயன்படுத்தமுடியும். இதன்படி class derivedல் variable xஐ பயன்படுத்தமுடியும்.

Constructor

Constructor என்பது ஒரு சிறப்புமிக்க functionஆகும். இந்த function ஒரு classக்கு objectஐ உருவாக்கும்பொழுது தன்னிச்சையாக அழைக்கப்படும். இந்த function ஒருமுறை மட்டுமே execute செய்யமுடியும். இந்த function ஒரு programக்கு தேவையான வளத்தை ஒதுக்குவதற்கு பயன்படுகிறது.

syntax

```

function __construct(Parameter_List)
{

```

Constructor Code

```
}
```

Destructor

Destructor function PHPல் ஒரு objectன் கடைசி உபயோகத்திற்கு பிறகு அழைக்கப்படுகிறது. இந்த function ஒரு programக்கு ஒதுக்கப்பட்ட வளத்தை நீக்குவதற்கு பயன்படுகிறது.

Syntax

```
function __destruct()  
{  
    Destructor Code  
}
```

Example

```
<?php  
class employee  
{  
    var $name;  
    var $dept;  
    function __construct($n,$d)  
    {  
        echo "<br>Calling Constructor<br>";  
        $this->name = $n;  
        $this->dept = $d;  
    }  
    function display()  
    {  
        echo "<br>Employee Name: ".$this->name;  
        echo "<br>Department: ".$this->dept;  
    }  
    function __destruct()  
    {
```

```
        echo "<br><br>Calling Destructor";  
    }  
}
```

```
}
```

```
}
```

```
$obj = new employee("Sajid","HR");
```

```
$obj->display();
```

```
?>
```

Output

Calling Constructor

Employee Name: Sajid
Department: HR

Calling Destructor

String Manipulation

PHPல் stringகளை கையாள இந்த functionகள் பயன்படுத்தப்படுகிறது. PHPல் stringஆனது case sensitiveஆக இருக்கிறது. PHPல் stringகளை கையாள பல functionகள் உள்ளன, அவற்றில் சில முக்கியமான functionகள் கீழே கொடுக்கப்பட்டுள்ளது.

- Length
- Reverse
- SubString
- String Search
- String Replace
- Case Change
- Comparison
- Explode/Implode

Length

கொடுக்கப்பட்டுள்ள stringன் அளவை கணக்கிட இந்த function பயன்படுகிறது. இந்த function ஒரே ஒரு parameter மட்டுமே இருக்கும், அது string ஆகும் இந்த function அந்த stringன் அளவை திரும்பத்தரும்

syntax

```
$length = strlen($string);
```

Reverse

இந்த function ஒரு stringஐ தலைகீழாக மற்ற பயன்படுகிறது. இது ஒரு stringஐ parameterஆக கொண்டு அதனை தலைகீழாக மாற்றி திரும்பத்தரும்.

syntax

```
$reverse = strrev($string);
```

SubString

இந்த function ஒரு stringன் பகுதியை மட்டும் பிரித்து எடுக்க பயன்படுகிறது. இந்த functionல் மொத்தம் மூன்று parameterகள் கொடுக்கப்படும், முதலாவது string, பிரித்து எடுக்க வேண்டிய stringன் ஆரம்பம், பிரித்து எடுக்கவேண்டிய stringன் அளவு. அளவு கொடுக்கப்படவில்லை என்றால் அசல் stringன் இறுதிவரை பிரித்துஎடுக்கும்.

Syntax

```
$substring = substr($string,Start,Lenght);
```

String Search

PHPல் stringல் தேடுதல் செய்வதற்கு இரண்டு functionகள் உள்ளன, அவை

1. strstr
2. strpos

strstr function ஒரு stringல் மற்றொரு string இருக்கிறதா என்று தேடுகிறது. தேடப்படும் string இருந்தால், எந்த இடத்தில் இருக்கிறதோ

அந்த இடத்திலிருந்து இறுதிவரை திரும்ப தருகிறது. இதில் before parameter TRUEஆக இருந்தால் தேடப்படும் string இருக்கும் இடத்திற்கு முன்னர் இருக்கும் string திரும்ப தருகிறது.

Syntax

```
$search = strstr($string,$search,before);
```

strpos function ஒரு stringல் இன்னொரு string இருக்கிறதா என்று தேடுகிறது, string இருந்தால் அது எந்த இடத்தில் இருக்கிறதோ அந்த எண் index மதிப்பு திரும்ப தருகிறது.

Syntax

```
$position = strpos($string,$search);
```

String Replace

இந்த function ஒரு string மற்றொரு stringல் இருந்தால் அதற்கு பதிலாக வேறு ஒரு stringஐ மாற்றியமைக்க பயன்படுகிறது. இதில் கூடுதலாக count parameter இடம்பெற்றிருக்கும் இது எத்தனைமுறை அந்த string மாற்றிஅமைக்கப்பட்டது என்பதை கணக்கிடும்..

Syntax

```
str_replace($search,$replace,$string,$count);
```

Case Change

PHPல் stringன் caseஐ மாற்றியமைக்க தனிப்பட்ட function உள்ளது. மொத்தமாக நன்கு case மற்றும் functionகள் உள்ளன, அவை

1. Uppercase
2. Lowercase
3. First Letter Uppercase
4. Title Case

Uppercase function ஒரு stringல் உள்ள அனைத்து எழுத்துக்களையும் uppercaseஆக மாற்றிஅமைகிறது.

Syntax

```
strtoupper($string);
```

Lowercase function ஒரு stringல் உள்ள அனைத்து எழுத்துக்களையும் lowercaseஆக மாற்றி அமைகிறது.

Syntax

```
strtolower($string);
```

First Letter Uppercase function ஒரு stringல் உள்ள முதல் எழுத்தை மட்டும் uppercaseஆக மாற்றி அமைகிறது.

Syntax

```
ucfirst($string);
```

Title Case function ஒரு stringல் உள்ள அனைத்து வார்த்தைகளின் முதல் எழுத்துக்களை uppercaseஆக மாற்றி அமைகிறது.

syntax

```
ucwords($string);
```

Comparison

PHPல் இரண்டு stringஐ ஒப்பிட இரண்டு functionகள் உள்ளன, அவை

1. strcmp
2. similar_text

strcmp function இரண்டு stringஐ ஒப்பிட பயன்படுகிறது மற்றும் இது ஒப்பீட்டின் விளைவாக ஒரு எண் மதிப்பை திரும்ப தருகிறது.

Syntac

```
strcmp($string1,$string2)
```

0 - if both the string are equal

<0 - if string1 is smaller than string2

>0 - if string1 is larger than string2

Regular Expression

PHPல் regular expression ஒரு பொதுவான மாதிரியை உருவாக்க பயன்படுகிறது, இதன்மூலம் pattern matching செய்யப்படுகிறது. இதில் பல operatorகளை கொண்டு பொதுவான மாதிரி உருவாக்கப்படுகிறது, இந்த operatorகளை meta character என்று அழைப்போம். ஒவ்வொரு meta characterக்கும் தனி அர்த்தம் உண்டு. சில முக்கியமான meta characterகள் மற்றும் அதன் பொருள் கீழே கொடுக்கப்பட்டுள்ளது

Meta Character	Meaning
/	use for begining and ending of a pattern
^	Matching at the begining
\$	Matching at the end
.	Check only one element
[Char List]	Single Character in the list
[^Char List]	Single Character not in the list
elm*	0 or many times
elm+	1 or maby times
elm?	0 or 1 time
{}	Exact number of Count
\	Check for meta character

PHPல் pattern matchingஐ செய்ய preg_match function பயன்படுத்தப்படுகிறது. இந்த function pattern match ஆனால் 1 திரும்ப தரும் இல்லையேல் 0 திரும்ப தரும்.

Syntax

```
preg_match("Pattern",$string);
```

Some example of using pattern matchin in PHP is given below

Check Only One Element

Example

```
$str = "a";  
$res = (preg_match("/^.$"/,$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Valid

Check Character in a List

Example

```
$str = "G";  
$res = (preg_match("/[abcd]$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Invalid

Check Elements 0/many, 1/many

Example

```
$str = "11";  
$res = (preg_match("/^10*1$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Valid

Example

```
$str = "11";  
$res = (preg_match("/^10+1$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Invalid

Check Elements 0/1, count

Example

```
$str = "1001";  
$res = (preg_match("/^10?1$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Invalid

Example

```
$str = "10001";  
$res = (preg_match("/^10{3}1$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern Match Valid

Check Meta Character

Example

```
$str = "1021.01";  
$res = (preg_match("/^[1-9][0-9]+\.[0-9]{2}$/",$str)?"Valid":"Invalid");  
echo "Pattern Match $res";
```

Output

Pattern match Invalid

2.2 File and Directory Handling

PHPல் பல file மற்றும் directory கையாளும் functionகள் உள்ளன. இந்த functionகள் serverல் file சம்பந்தமான செயல்கள் மற்றும் directory சம்பந்தமான செயல்களை செய்யப்பயன்படும்.

File Handling

PHPல் fileஐ கையாள இந்த நன்கு செயல்கள் உபயோகிக்கப்படுகிறது.

1. Open File
2. Read File
3. Write File
4. Close File

Open File

PHPல் ஒரு fileஐ திறக்க fopen function பயன்படுத்தப்படுகிறது. இதில் இரண்டு parameterகள் உள்ளன, முதல் parameter fileன் பெயர் மற்றும் இரண்டாவது parameter fileஐ திறக்கும் முறைஆகும். ஒரு file சரியாக திறக்கப்பட்டால் அதன் file handleஐ இந்த function திரும்ப தரும்.

syntax

```
$file_handle = fopen(filename,File Opening mode);
```

File Opening Mode

There are six different file opening modes available in PHP. Each mode has its own property.

File Opening Mode	Description
r	fileஐ படிக்க மட்டும் திறக்கும்.
w	Fileஐ எழுதுவதற்காக மட்டும் திறக்கும், இதில் fileல் உள்ள அனைத்தும் அழிக்கப்படும். இதில் கொடுக்கப்பட்ட file இல்லை என்றால் ஒரு புது file உருவாக்கப்படுகிறது.
a	Fileஐ எழுதுவதற்காக திறந்து file pointerஐ fileன் இறுதியில் வைக்கும். Fileல் உள்ள அனைத்தும் அழியாமல் இருக்கும். இதில் கொடுக்கப்பட்ட file இல்லை என்றால் ஒரு புது file உருவாக்கப்படுகிறது.
r+	Fileஐ படிக்க மற்றும் எழுத திறக்கும்.
w+	Fileஐ எழுத மற்றும் படிக்க திறக்கும். இதில் fileல் உள்ள அனைத்தும் அழிக்கப்படும். இதில் கொடுக்கப்பட்ட file இல்லை என்றால் ஒரு புது file உருவாக்கப்படுகிறது.
a+	Fileஐ படிக்க மற்றும் எழுத திறக்கும், file pointerஐ fileன் இறுதியில் வைக்கும். இதில் கொடுக்கப்பட்ட file இல்லை என்றால் ஒரு புது file உருவாக்கப்படுகிறது.

example

```
$f_handle = fopen("test.txt","r");
```

Read file

ஒரு fileஐ படிக்க fread function PHPல் பயன்படுத்தப்படுகிறது. இதில் இரண்டு parameterகள் கொடுக்கப்படும், file handle மற்றும் fileன் size. File handle fopen functionல் இருந்து பெறப்படுகிறது மற்றும் fileன் size filesize function மூலம் பெறப்படுகிறது.

syntax

```
$size = filesize($filename);
$text = fread($file_handle,$size);
```

example

```
$size = filesize("text.txt");
$text = fread($file_handle,$size);
```

File Write

PHPல் ஒரு fileல் எழுதுவதற்கு fwrite function பயன்படுத்தப்படுகிறது. இந்த function இரண்டு parameterகளை கொண்டிருக்கும், file handle மற்றும் fileல் எழுதவேண்டிய text.

syntax

```
fwrite($file_handle,text);
```

example

```
fwrite($file_handle,"Welcome");
```

File Close

PHPல் fclose function ஒரு fileஐ மூட பயன்படுத்தப்படுகிறது. இதில் ஒரு parameter மட்டுமே இருக்கும் அது file handle. திறக்கப்பட்ட file சரியாக மூடப்படவேண்டும், அப்படி செய்தால் அந்த fileல் உள்ள அனைத்தும் முழுமையாக சேமிக்கப்படும்.

Syntax

```
fclose($file_handle);
```

Example

```
fclose($file_handle);
```

File Handling Complete Example

```
<?php
$text = "File Handling Introduction";
```

```
$fhandle = fopen("test.txt","w");
fwrite($fhandle,$text);
fclose($fhandle);
```

```
$fhandle = fopen("test.txt","r");
$size = filesize("test.txt");
$text = fread($fhandle,$size);
echo "File Content";
echo "<br>$text";
```

?>

Directory Handling

Fileகளை கொண்டிருக்கும் கொள்கலன் directory எனப்படும். PHPல் directory சம்பந்தமான function மற்றும் classகள் உள்ளன. இந்த functionகள் serverல் உள்ள directoryஐ உபயோகிக்க பயன்படுகிறது.

Directory Handling Functions

PHPல் directory handlingக்கு மூன்று functionகள் உள்ளன. ஒரு directoryஐ திறக்க opendir function பயன்படுத்தப்படுகிறது. Directoryல் உள்ளவற்றை படிப்பதற்கு readdir function பயன்படுத்தப்படுகிறது, இந்த function ஒரு loopல் கொடுக்கும்பொழுது அந்த directoryல் உள்ளவற்றை ஒவ்வொன்றாக படிக்க பயன்படுகிறது. திறக்கப்பட்ட directoryஐ மூட closedir function பயன்படுத்தப்படுகிறது.

Syntax

```
$dir_handle = opendir(path);
$file_name = readdir($dir_handle);
closedir($dir_handle);
```

Example

```
<?php
$dh = opendir("images");
while(($file=readdir($dh))!=FALSE)
{
```



```
    echo "<br>$file";
}
closedir($dh);
?>
```

Dir Class

dir() function ஒரு objectஐ சார்ந்த functionஆகும் இது directory classக்கு ஒரு பிரதியை உருவாகும். இந்த function ஒரு directoryன் pathஐ parameterஆக பெற்றுக்கொண்டு ஒரு objectஐ திரும்ப தருகிறது. PHPல் read() function மூலமாக அந்த objectல் உள்ளவை படிக்கப்படுகிறது. திறக்கப்பட்ட directoryஐ மூட close() function பயன்படுத்தப்படுகிறது.

Syntax

```
$obj = new dir(Directory path);
$obj->read();
$obj->close();
```

Example

```
<?php
    $obj = dir("images");
    while(($file=$obj->read())!=FALSE)
    {
        echo "<br>$file";
    }
    $obj->close();
?>
```

Including Files

PHPல் ஒரு fileல் உள்ளவற்றை மற்றொரு fileல் பயன்படுத்தலாம். இந்த செயல்முறை serverல் script execute செய்வதற்கு முன்னதாக செயல்படுத்தப்படுகிறது. இதனை செய்வதற்கு இரண்டு functionகள் உபயோகிக்கப்படுகிறது, அவை

- include() function
- require() function

include() functionல் கொடுக்கப்பட்டுள்ள fileல் உள்ள அனைத்தையும் அந்த functionஐ பயன்படுத்திய fileல் copy செய்கிறது. இந்த செயலில் ஏதேனும் பிழை இருந்தால் எச்சரிக்கை மட்டும் கொடுத்துவிட்டு PHP script execute ஆகும்.

require() functionம் இதே செயலை செய்யும். இதில் உள்ள வித்தியாசம் என்னவென்றால், fileஐ copy செய்யும்பொழுது பிழை இருந்தால் இந்த function fatal errorஐ உருவாகும் மற்றும் script executionஐ நிறுத்தும்.

Syntax

```
include(Include File);
require(Include File);
```

Example

menu.php

```
<a href="eg30.php">Patter Matching</a><br>
<a href="eg31.php">File Handling</a><br>
<a href="eg32.php">Directory Handling</a>
```

include.php

```
<html>
<head>
    <title>File Inclusion</title>
</head>
<body>
    <h3>File Inclusion</h3>
    <?php
        include("menu.php");
    ?>
    <h3>This Page Include Menu.php</h3>
```

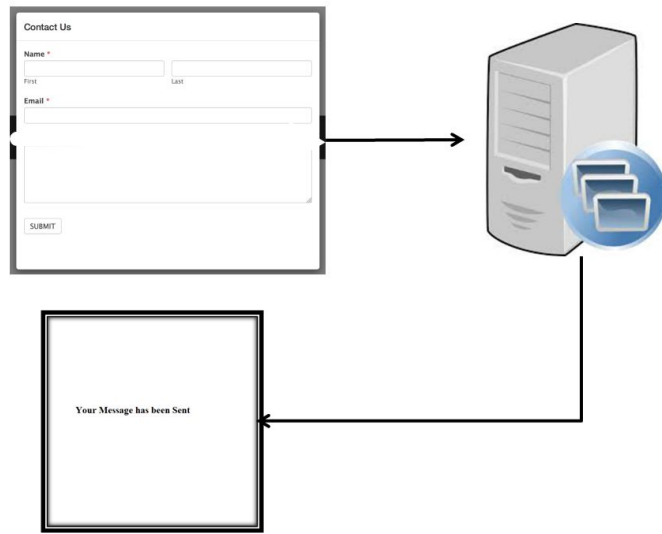
```

<!-- <?php
    include("test.php");
?>
<h3>Testing require</h3-->
</body>
</html>

```

2.3 Working with Forms

Web சார்ந்த programmingல் form என்பது user பெறப்பட்ட மதிப்புகளை serverக்கு அனுப்ப பயன்படுகிறது. Formஐ பயன்படுத்தாமல் ஒரு userல் websiteல் உள்ளிட்ட மதிப்புகள் serverக்கு அனுப்ப முடியாது. Form ஒரு HTML elementஆகும்.



Form Process Flow

Processing Forms

HTML form tagல் தனிப்பட்ட attributeகள் உள்ளன இவை மதிப்புகளை சேகரிக்கவும் அதை serverக்கு அனுப்பவும் பயன்படுகிறது.

D. NATARAJASIVAN/TNPT

Syntax

```

<form action="Script_URL" method="POST/GET">
    HTML input elements to collect Data with name attribute
    Submit button to send data to "Script_URL"
</form>

```

Example

```

<form action="<?php echo $_SERVER["PHP_SELF"] ?>"
method="POST">
    <input type="text" name="sname"/>
    <input type="submit" value="Send name"/>
</form>

```

Form tag இரண்டு முக்கியமான attributeகள் உள்ளன, action மற்றும் method. Action Attribute formல் சேகரித்த மதிப்புகளை எந்த fileக்கு அனுப்ப வேண்டும் என்பதை தெரிவிக்கும். இது serverல் இருக்கும் வேறு ஒரு script fileஐயோ அல்லது இதே fileஐயோ குறிக்கும். அதே fileஐ குறிக்க “\$_SERVER['PHP_SELF']” action attributeல் பயன்படுத்தப்படுகிறது.

இரண்டாவது attribute method attribute ஆகும். இது சேகரித்த மதிப்புகள் எந்த methodன் மூலமாக அனுப்பவேண்டும் என்பதை குறிக்கும். இது URL encoding முறையை பயன்படுத்தி மதிப்புகளை அனுப்புகிறது. URL encoding name, value pair (name=value) முறையில் செயல்படுகிறது, இதில் வெவ்வேறு மதிப்புகளை அனுப்ப ambersent(&) symbol பயன்படுத்தப்படுகிறது. இதில் GET அல்லது POST முறையில் மதிப்புகளை அனுப்பலாம்.

GET

இந்த முறையில் மதிப்புகள் url encoding மூலமாக action attributeல் குறிப்பிடப்பட்டுள்ள script pageஉடன் question mark (?) கொண்டு சேர்க்கப்பட்டிருக்கும்.

Example

```
localhost/test.php?rollno=101&last=mani
```

இங்கே கொடுக்கப்பட்டுள்ள உதாரணத்தில் rollno மற்றும் name மதிப்புகள் test.php எனும் script pageக்கு URL encoding முறையில் அனுப்பப்படுகிறது. இந்த முறையில் 1024 எழுத்துக்களை மட்டுமே அனுப்ப முடியும், அதுமட்டும் இல்லாமல் அனுப்பப்படும் மதிப்பு browserன் address barல் அனைவரும் பார்க்கும் வகையில் தெரியும். இதனால் இந்த முறையை username/passwordகளை அனுப்ப பயன்படுத்துவது இல்லை. PHPல் GET முறையில் அனுப்பப்பட்ட மதிப்புகளை \$_GET associative arrayன் மூலமாக பெறப்படுகிறது. இந்த arrayல் மதிப்பின் பெயர் indexஆக பயன்படுத்தப்படுகிறது.

Example

```
$roll_no = $_GET["rollno"];  
$stud_name = $_GET["name"];
```

POST

இந்த முறையில் சேகரிக்கப்பட்ட மதிப்புகள் HTTP header மூலமாக அனுப்பப்படுகிறது. இது URL encodingஐ பயன்படுத்தி மதிப்புகளை header மூலக அனுப்புகிறது இது QUERY_STRING என்று அழைக்கப்படுகிறது. இதில் உள்ள முக்கியமான நன்மை என்னவென்றால் இதில் மதிப்புகளை அனுப்புவதில் எந்த ஒரு கட்டுப்பாடும் கிடையாது, அதுபோல அனுப்பப்படும் மதிப்பு யாருக்கும் பார்க்கும்படி இருக்காது மேலும் இதில் binary மதிப்பையும் அனுப்ப முடியும். PHPல் POST முறையில் அனுப்பப்பட்ட மதிப்பு \$_POST

associative arrayஐ கொண்டு பெறப்படுகிறது. இந்த arrayல் மதிப்பின் பெயர் indexஆக பயன்படுத்தப்படுகிறது.

Example

```
$roll_no = $_POST["rollno"];  
$stud_name = $_POST["name"];
```

Form Validation

Form validation ஒரு formல் userல் கொடுக்கப்பட்ட மடிப்புகளை சரிபார்ப்பதாகும். இதன் மூலம் சரியான மதிப்பு மட்டுமே அனுப்பப்படுகிறது என்றும், மேலும் எந்த ஒரு inputம் காலியாக இல்லாமல் இருக்கிறதா என்பது உறுதிசெய்யப்படுகிறது. PHPல் validation serverல் செய்யப்படுகிறது. சேகரித்த மதிப்பு serverக்கு அனுப்பப்படுகிறது, serverல் அந்த மதிப்புகள் சரியானவைதான் என்று சரிபார்க்கப்படுகிறது. மதிப்பு சரியானதாக இல்லையென்றால் server அதனை webpageக்கு தெரியப்படுத்துகிறது.

PHP server side validationஐ செயல்படுத்த இந்த வழிமுறை பயன்படுகிறது.

- Clientல் இருந்து serverக்கு மதிப்பு அனுப்பப்படுகிறது என்பதை சரிபார்த்தல். இதனை isset() function செய்கிறது.
- அனுப்பப்பட்ட தகவலில் மதிப்பு இருக்கிறதா என்பதை சரிபார்த்தல். இதற்கு empty() function பயன்படுகிறது.
- மதிப்பு தவறாக இருந்தால் ஒரு error message input tagக்கு ஒதுக்கப்படுகிறது. இதனை செயல்படுத்த ஒருவரு input tagஉடன் ஒரு span element இணைக்கப்படுகிறது. இதனை முன்னிலைப்படுத்த span elementக்கு CSSல் error class பயன்படுத்தப்படுகிறது.

Example

```

<html>
<style>
    .error
    {
        color:#FF0000;
    }
</style>
<?php
$nameerr = ""; // define variables and set to empty values
if (isset($_POST["username"]))
{
    if (empty($_POST["username"]))
    {
        $nameerr = "User Name is required";
    }
    else
    {
        $usname = $_POST["username"];
    }
}
?>
<body>
<?php
if(empty($nameerr))
{
    echo "<h2>User Details</h2>";
    echo "<br>User Name: ".$usname;
}
} //End of If ISSET
else
{
?>
<form method = "post" action = "<?php echo
$_SERVER["PHP_SELF"];?>">
<label>User Name:</label>

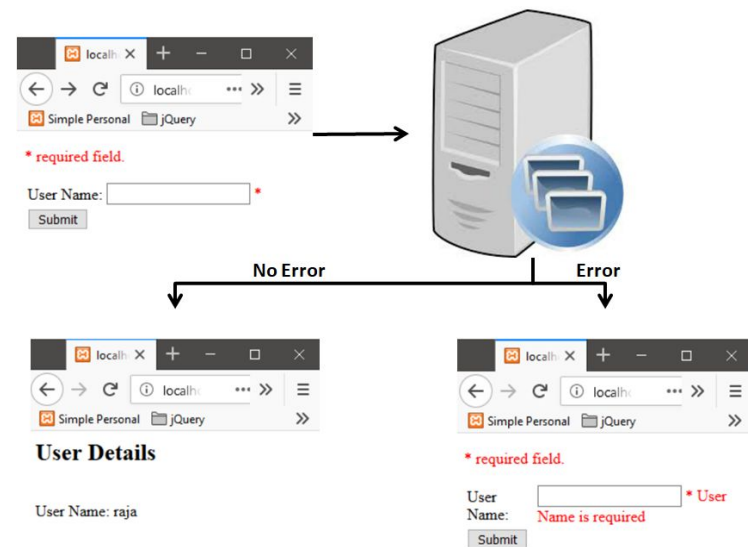
```

```

<input type = "text" name = "username">
<span class = "error">* <?php echo $nameerr;?></span><br>
<input type = "submit" name = "submit" value = "Submit">
</form>
<?php
} //End of Else ISSET
?>
</body>
</html>

```

Output



Introduction to advanced PHP Concepts

PHPல் உள்ள Advanced concepts

- Error Handling
- Cookies

- Session

Error Handling

Program executeஆகும் பொழுது உருவாகும் தவறுகளை திறம்பட நிர்வகிக்க Error handling பயன்படுகிறது. PHPல் error handling இரண்டு முறையில் செயல்படுத்தப்படுகிறது, die() function மற்றும் try...catch block.

die() function

இந்த function ஒரு error messageஐ தெரிவித்துவிட்டு program executeஆவதை நிறுத்துகிறது. இதுவே PHP உள்ள எளிமையான error handling optionஆகும்.

syntax

```
die("Error Message");
```

Example

```
<?php
$mark = -10;
if($mark<0)
    die("Invalid Mark");
if($mark>50)
    echo "Pass";
else
    echo "Fail";
echo "Marks Displayed";
?>
```

இந்த உதாரணத்தில் markன் மதிப்பு negativeஆக கொடுப்பது தவறாகும். இதனை சரிபார்க்க ஒரு if statement பயன்படுத்தப்படுகிறது இதில் markன் மதிப்பு 0 விட குறைவாக இருந்தால் die() function

அழைக்கப்படுகிறது. die() function அழைத்தவுடன் அது ஒரு error messageஐ தெரிவித்துவிட்டு script executionஐ நிறுத்துகிறது

try...catch

இந்த முறையில் error handling செய்வதை exception handling என்று அழைப்பர். இதுவே error handling செய்வதற்கு சிறந்தமுறையாகும், இதில் ஒரு error ஏற்பட்டால் அதற்கான error message தெரிவிக்கப்படுகிறது மேலும் script execution நிறுத்தப்படாது.

syntax

```
try
{
    throw new Exception(msg);
}
catch(Exception $e)
{
    Display error Message
}
```

Example

```
<?php
$mark = -10;
try
{
    if($mark<0)
        throw new Exception("Invalid Mark");
    if($mark>50)
        echo "Pass";
}
```

```

else
    echo "Fail";
}
catch(Exception $ex)
{
    echo $ex->getMessage();
    echo " Error: ".$ex->getMessage();
}
echo "<br>Marks Displayed";
?>

```

இந்த உதாரணத்தில் scriptல் error வந்தாலும் முழுமையாக execute ஆகும்.

Cookies

Cookies என்பது clientல் சேமிக்கப்படும் text fileஆகும். இவை கண்காணிப்பிற்காக பயன்படுத்தப்படுகிறது. பொதுவாக இது ஒரு userஐ அடையாளம்காண பயன்படுகிறது. ஒரு cookieல் மூன்று functionகளை செய்யலாம்.

- Create cookie
- Access cookie
- Delete cookie

Create cookie

ஒரு cookieஐ உருவாக்க setcookie function பயன்படுகிறது. இந்த function அதில் கொடுக்கப்பட்டுள்ள parameterகளை பொறுத்து ஒரு cookieஐ உருவாக்கி clientன் computerல் சேமிக்கிறது.

syntax

setcookie(name,value,expiry,path,domain,security);

- name - இதுவே cookieஐ access செய்ய பயன்படும் variableன் பெயர் ஆகும்

- value - இதுவே cookieல் சேமிக்கவேண்டிய மதிப்பாகும். இதனை access செய்வதற்கு name parameter பயன்படுகிறது.
- expiry - இதுவே cookieன் வாழ்நாளை குறிக்க பயன்படும். இதில் நேரத்தின் மதிப்பு நொடிகளில் தெரிவிக்கப்படுகிறது.
- path - இது ஒரு cookie எந்த webpageல் access செய்யமுடியும் என்பதை தெரிவிக்கும். இதில் “/” உபயோகப்படுத்தினால் அந்த cookie மொத்த websiteலும் access செய்யமுடியும்.
- domain - இது ஒரு cookie எந்த domainல் கிடைக்கப்பெறும் என்பதை தெரிவிக்கும்.
- security - இந்த parameter 0 அல்லது 1 மதிப்பை பெரும். இதில் 0 HTTP மற்றும் 1 HTTPS குறிக்கும்.

Access Cookie

PHPல் ஒரு cookieஐ access செய்வதற்கு \$_COOKIE associative array பயன்படுத்தப்படுகிறது. இதில் cookieஐ உருவாக்கும்பொழுது பயன்படுத்திய name parameter மதிப்பை indexஆக இருக்கும்.

syntax

\$_COOKIE[“name”];

Delete Cookie

ஒரு cookieஐ அழிக்க setcookie() functionல் எதிர்மறை expiry மதிப்பு பயன்படுத்தப்படுகிறது.

syntax

```
setcookie("name","",time()-10,"/","",0);
```

Example

setcookie.html

```
<html>
<body>
<form method="POST" action="cookie.php">
<input type="text"name="username"/>
<input type="submit" value="set cookie"/>
</form>
</body>
</html>
```

cookie.php

```
<?php
if(isset($_POST["username"]))
{
$username = $_POST["username"];
setcookie("username",$username, time()+30,"/","",0);
}
?>
<html>
<body>
<p>User Name stored in Cookie for 30 seconds</p>
</body>
</html>
```

getcookie.php

```
<html>
<body>
<?php
$val = $_COOKIE["username"];
?>
```

```
<p>User Name in Cookie is "<?php echo $val ?>"</p>
</body>
</html>
```

இந்த உதாரணத்தில் setcookie.html pageல் user name பெறப்பட்டு cookie.php scriptக்கு அனுப்பப்படுகிறது. இந்த PHP script 30 வினாடிகளில் அழியக்கூடிய ஒரு cookieஐ உருவாக்கி client computerல் சேமிக்கிறது. getcookie.php page cookieஐ அதன் பெயரை கொண்டு access செய்து அதனை வெளிக்காட்டுகிறது. getcookie.php page 30 வினாடிகளுக்கு பிறகு execute செய்தால் cookie அழிந்திருக்கு அதனால் user nameஐ access செய்யமுடியாது.

Session

பல web pageகளில் பயன்படுத்தக்கூடிய விதத்தில் ஒரு மதிப்பை சேமிப்பது sessionஆகும். session serverல் உள்ள தற்காலிக directoryல் fileஐ உருவாக்குகிறது. இந்த fileல் session variableகள் மற்றும் அதன் மதிப்புகள் சேமிக்கப்படுகிறது. Sessionல் இரண்டு functionகள் செயல்படுத்தப்படுகிறது

- Create session
- Destroy session

Create Session

ஒரு sessionஐ உருவாக்க முதலில் PHPல் session_start() function பயன்படுத்தப்படுகிறது. இது serverல் ஒரு sessionஐ ஆரம்பிக்கிறது, இதில் மதிப்புகளை சேமிக்கமுடியும். ஆரம்பிக்கப்பட்ட sessionல் மதிப்புகளை சேமிக்க \$_SESSION associative array பயன்படுத்தப்படுகிறது. இதில் உள்ள indexன் பெயரே அதில் சேமிக்கப்பட்டிருக்கும் மதிப்பை அடையாளம்காண பயன்படுகிறது

Syntax

```
session_start();
$_SESSION["Name"] = Value;
```

Destroy Session

ஒரு serverல் உள்ள sessionஐ அழிக்க இரண்டு முறைகள் உள்ளன. முதலாவது session_destroy() function மூலமாக அணைத்து session variableஐயும் அழிப்பது. இரண்டாவது unset() function மூலமாக ஒரு குறிப்பிட்ட session variableஐ மட்டும் அழிப்பது.

Syntax

```
session_destroy();
unset($_SESSION["Name"]);
```

Example

setsession.html

```
<html>
<body>
<form method="POST" action="session.php">
<label>Name: </label>
<input type="text" name="studname"/><br>
<label>Roll No: </label>
<input type="text" name="rollno"/>
<input type="submit" value="set Session"/>
</form>
</body>
</html>
```

session.php

```
<?php
if(isset($_POST["studname"]))
{
    $stuname = $_POST["studname"];
```

```
$rollno = $_POST["rollno"];
session_start();
$_SESSION["studname"] = $stuname;
$_SESSION["rollno"] = $rollno;
}
?>
<html>
<body>
<p>Student Name and Roll No stored in session</p>
</body>
</html>
```

getsession.php

```
<html>
<body>
<?php
    session_start();
    $studname = $_SESSION["studname"];
    $rollno = $_SESSION["rollno"];
    unset($_SESSION["studname"]);
    session_destroy();
?>
<p>Student Name is "<?php echo $studname ?>" Roll No is
<?php echo $rollno ?> </p>
</body>
</html>
```

இந்த உதாரணத்தில் setsession.html student name மற்றும் roll numberஐ பெற்று session.php scriptக்கு அனுப்புகிறது. session.php scriptல் ஒரு session உருவாக்கப்பட்டு அதில் student name மற்றும் roll number சேமிக்கப்படுகிறது. getsession.php pageன் மூலமாக sessionல் சேமிக்கப்பட்ட student name மற்றும் roll number access செய்யப்படுகிறது.

Session variableகள் படித்தபிறகு unset() மற்றும் session_destroy() functionகால் மூலமாக அழிக்கப்படுகிறது.

Review Questions

Part-A

1. What is PHP?
2. What is a Variable in PHP?
3. What is a constant in PHP?
4. What is a conditional Statement in PHP?
5. What is a looping statement in PHP?
6. What is an array in PHP?
7. Write the syntax of function in PHP.
8. Define class in PHP.
9. Write the syntax of any two string manipulation functions in PHP.
10. What are the file reading modes available in PHP?
11. What is a form?
12. What is a cookie?
13. What is a session?

Part-B

1. Write the benefits of PHP.
2. Explain the usage of variables in PHP.
3. Write about the data types available in PHP.
4. Explain if...else statement in PHP with an example.
5. Explain switch statement in PHP with an example.

6. Write about Associative array in PHP.
7. Explain function with return value in PHP.
8. What are the access specifiers available in PHP? Explain.
9. Write a PHP script to read and write a file.
10. How form is processed in PHP? Explain with an example.

Part-C

1. Explain in detail about PHP Programming in web environment.
2. What are the common PHP script elements? Explain in detail.
3. Explain in detail about Operators in PHP.
4. What are the looping statements available in PHP? Explain with example.
5. What are the numeric array functions available in PHP?
6. Explain in detail about Function with parameter and function with default values in PHP.
7. Explain in detail about inheritance in PHP.
8. Explain in detail about Regular Expression in PHP.
9. Explain in detail about Including Files in PHP.
10. Write in detail about form validation in PHP.
11. Elaborate advanced concepts in PHP.