**PHP**

**INTRO:**

Developer - **Rasmus Lerdorf** Year - **1994**

PHP –PHP Hypertext Preprocessor

**Hyper Text:**

One Page linked with another page (page within a page)

**Server side** scripting language (DP,DB,E site, Session Tracking)

Create **dynamic content** that interacts with **databases**

**Major DBs**:

MySQL , Oracle , Sybase , Informix , Microsoft SQL Server

**Major Protocols**:

POP3 , IMAP , LDAP .

PHP 4 support JAVA and **distributed object architectures** (COM and CORBA), making n-tier development a possibility for the first time.

Like us **C Syntax**.

**COMMON USE:**

Perform system function ( like **File** create ,delete ,open ,read ,write ,close )

Handle **Form** ( Get form data , store data to file ,return data)

Handle **DB** ( add ,delete ,modify)

**Restrict** some Page

**Encrypt** data

**CHARACTERISTICS OF PHP:**

**Simplicity:**

No need **Libraries**

No need **special Directories**

Engine run code after the first escape sequence () // any syntactic error occur

Does not displayed ,otherwise displayed well.

**Efficiency:**

Multiuser environment (WWW)

PHP 4: resource allocation mechanism

Object oriented programming

Session Management

Reference Counting

**Security:**

Provide security safeguards.

**System Level** :

cgi-bin folder - Admin place Restriction on the location .

Which user view , execute and view guard

Information.

**Application Level** :

**Data encryption** predefined function

Source **does not view** in a browser

Does not produce **browser support** related problem

**Platform Independent**

Access **External component** ( JAVA Beans ,Win32 COM)

**Flexibility:**

Open source Environment

**Familiarity:**

This is typically C or Pascal program (minimize learning curve)

**SYNTAX - OVERVIEW**

**Escaping to PHP:**

**PHP** Parsing Engine - Differentiate PHP code from other Elements

Four Ways:

**Canonical PHP tags**

**<?php...?>**

**Short-open (SGML-style) tags**

**<?...?>**

Two Things:

Enable short-tags configuration

Set short \_open\_ tag setting in in your php.ini file

**ASP-style tags** :

**<%...%>**

Set configuration option in php.ini

**HTML script tags:**

**<script language=”PHP”>…</script>**

**Commenting Code:**

**Comment:**

Some part of code ignored in a program for user referencing purpose

**Single Line Comment:**

<?php

# This is a comment

# This is the second line of the comment

// This is a comment too print "An example with single lines"; ?>

**Multi Line Comments:**

<?php

/\* This is a comment

this is the second line of the comment

this is a comment too print "An example with single \*/

?>

**Multi Line Printing :**

<? # First Example

print <<<END This uses the "here document" syntax to output multiple lines with $variable interpolation. Note that the here document terminator must appear on a line with just a semicolon no extra whitespace! END;

# Second Example

print "This spans multiple lines. The newlines will

be output as well";

?>

**PHP is whitespace insensitive**

$four = 2 + 2; // single spaces

$four <tab>=<tab2<tab>+<tab>2 ; // spaces and tabs

$four = 2+ 2; // multiple lines

**PHP is case sensitive**

<html>

<body>

<? $capital = 67;

print("Variable capital is $capital<br>"); // 67

print("Variable CaPiTaL is $CaPiTaL<br>"); ?> //

</body>

</html>

**Statements are expressions terminated by semicolons**

PHP is any expression that is followed by a **semicolon (;)**

$greeting = "Welcome to PHP!";

**Expressions are combinations of tokens**

numbers (3.14159), strings (.two.), variables ($two), constants (TRUE)

**Braces make blocks**

if (3 == 2 + 1)

{

print("Good - I haven't totally");

print("lost my mind.<br>");

}

**VARIABLE TYPES**

* dollar sign (**$**)
* value of a variable is the value of its most **recent assignment**
* Variables are assigned with the **= operator**, with the variable on the **left-hand** side
* Variables in PHP do not have **intrinsic types**
* Variables used before they are assigned have **default values**
* c**onverting types** from one to another when necessary
* PHP variables are **Perl-like**

**Eight data types:**

**Integers**: are **whole** numbers, **without a decimal** point, like 4195.

**Doubles**: are **floating-point** numbers, like 3.14159 or 49.1.

**Booleans**: either **true or false**.

**NULL**: is a special type that only has one value: NULL.

**Strings**: are **sequences of characters**, like 'PHP supports string operations.'

**Arrays**: are **named and indexed** collections of other values.

**Objects**: are **instances of programmer-defined classes,** which can package up both other kinds of values and functions that are specific to the class.

**Resources**: are special variables that hold references to resources external to PHP (**such as database connections**).

**Integer:**

$negative= -100;

$octal=012;

$hex=0x23;

echo $octal+$hex;

**Double:**

$many = 2.2888800;

$many\_2 = 2.2111200;

$few = $many + $many\_2;

print(.$many + $many\_2 = $few<br>.);

**Boolean:**

if (TRUE)

print("This will always print<br>");

else

print("This will never print<br>");

Number:

Exactly zero - False

String:

‘0’ + empty - False

NULL - False

**Null:**

$my\_var =NULL $my\_var=null

Evaluates to FALSE in a Boolean context.

returns FALSE when tested with IsSet() function

**String:**

<?

$variable = "name";

$literally = **'My $variable will not print!\\n'**; //My $variable will not print

print($literally);

$literally = **"My $variable will print!\\n"**; // My name will print

print($literally);

?>

**Escape-Sequence:**

\n is replaced by the **newline** character

\r is replaced by the **carriage-return** character

\t is replaced by the **tab** character

\$ is replaced by the **dollar sign** itself ($)

\" is replaced by a **single double-quote (")**

\\ is replaced by a single **backslash (\)**

**Variable Naming:**

must begin with a letter or underscore character.

consist of numbers, letters, underscores but cannot use + , - , % , ( , ) . & ,etc

**PHP – Variables:**

**Four scope types :**

**Local variables**

<?php

$x = 4;

function assignx () {

**$x = 0;**

print "\$x inside function is $x. ";

}

assignx();

print "\$x outside of function is $x. ";

?>

**Function parameters**

<?php

$sum=call('b','a');

**$sum2=call(10,40);**

print("$sum\n");

print($sum2);

**function call($value='',$value2)**

{

return $value+$value2;

}

?>

**Global variables**

<?php

$glo=10;

increment();

function increment()

{

**GLOBAL $glo**;

while($glo==15){

$glo++;

print($glo);

}

}

print($glo);

?>

**Static variables**

<?php

function keep\_track() {

**STATIC $count = 0**;

$count++;

print $count;

print " ";

}

keep\_track();

keep\_track();

keep\_track();

?>

**PHP ─ CONSTANTS**

* Always uppercase
* define() function and to retrieve the value of a constant
* Do not need to have a constant with a $
* Function constant() to read a constant's value(Change name Dynamically)
* Only scalar data (boolean, integer, float and string).

**constant() function**

<?php

**define("MINSIZE", 50);**

$var=10;

echo MINSIZE; //50

echo **constant("MINSIZE");** // same thing as the previous line //50

fun();

function fun(){

GLOBAL $var;

define("SIZE",MINSIZE+ $var);

print(SIZE); // 60

}

?>

**PHP Magic constants:**

<?php

print("optimus");

**print(\_\_LINE\_\_);** //print no of line 3

**print(\_\_FILE\_\_);** //print E:\Xampp\htdocs\Practice\Magic constatnts.php

print("arg");

fun();

function fun()

{

print("optimus");

}

**print(\_\_FUNCTION\_\_);**

class first{

function \_\_construct()

{

print("first classs");

}

}

**print(\_\_CLASS\_\_ );**

**print(\_\_METHOD\_\_);**

?>

**PHP ─ Operator Types**

**Arithmetic Operators**

<?php

$num1=20;

$num2=50;

print("<h1> Arithmetic operators and Assignment</h1> <br>");

echo "Addtion=",$num2+$num1."<br>";// print Addition=70

echo "Subtraction=",$num2-$num1."<br>";// print Subraction=30

echo "Mul=",$num2\*$num1."<br>";// print Mul=1000

echo "Div=",$num2/$num1."<br>";// print Div=2.5

echo "Mod=",$num2%$num1."<br>";// print Mod=10

echo "Inc=",$num2++."<br>";// print Inc=50

echo "Dec=",$num1--."<br>";// print Dec=20

?>

**Comparison Operators**

<?php

$num1=20;

$num2=50;

print("<h1> Comparision operators</h1> <br>");

echo "$num2==$num1 &nbsp&nbsp&nbsp;",$num2==$num1."<br>";// print $num2==$num1

echo "<br>";

echo "$num2!=$num1 &nbsp&nbsp&nbsp;",$num2!=$num1."<br>";// print $num2!=$num1 1

echo "<br>";

echo "$num2>$num1 &nbsp&nbsp&nbsp;",$num2>$num1."<br>";// print $num2>$num1 1

echo "<br>";

echo "$num2<$num1 &nbsp&nbsp&nbsp;",$num2<$num1."<br>";// print $num2<$num1

echo "<br>";

echo "$num2>=$num1 &nbsp&nbsp&nbsp;",$num2>=$num1."<br>";// print $num2>=$num1 1

echo "<br>";

echo "$num2<=$num1 &nbsp&nbsp&nbsp;",$num2<=$num1."<br>";// print $num2<=$num1

?>

**Logical (or Relational) Operators**

<?php

$num1=20;

$num2=50;

print("<h1> Logical operators</h1> <br>");

echo "$num1 and $num2",$num1 and $num2."<br>";// print $num1 and $num2 1

echo"<br>";

echo "$num1 or $num2",$num1 or $num2."<br>";// print $num1 or $num2

echo"<br>";

echo "$num1 && $num2",$num1 && $num2."<br>";// print $num1 && $num2 1

echo"<br>";

echo "$num1 || $num2",$num1 || $num2."<br>";// print $num1 || $num2 1

echo"<br>";

echo "!($num1 and $num2)",!($num1 and $num2)."<br>";// print !($num1 and $num2)

?>

**Assignment Operators**

**Conditional (or ternary) Operators**

<?php

$var1=40;

$var2=20;

($var1>$var2)?print("var1 big"):print("var2 big")

?>

**Category Operator Associativity**

**Unary ! ++ -- Right to left**

Multiplicative \* / % Left to right

Additive + - Left to right

Relational < <= > >= Left to right

Equality == != Left to right

Logical AND && Left to right

Logical OR || Left to right

**Conditional ? : Right to left**

**Assignment = += -= \*= /= %= Right to left**

**PHP DECISION MAKING**

**Example:**

<?php

$var1=20;

$var2=20;

**if ($var2>$var1)** {

print("var2 is bigger");

}

**else if($var1>$var2)**{

print("var1 is bigger");

}

**else**{

print("var1 and var2 same");

}

$var=date("D");

**switch ($var)** {

**case 'Sun'**:

# code...

print("Sunday");

break;

case 'Mon':

# code...

print("Monday");

break;

case 'Tue':

# code...

print("Tuesday");

break;

case 'Thu':

# code...

print("Thursday");

break;

case 'Fri':

# code...

print("Friday");

break;

case 'Sat':

# code...

print("Saturday");

break;

**default:**

# code...

print("nothing ");

break;

}?>

**LOOP TYPES**

**Example:**

<?php

**for ($i=0; $i <10 ; $i++)** {

# code...

if($i==5)

**break;**

print($i);

}

print("<br>");

$j=0;

inner:

**while ( $j<= 10)** {

# code...

if($j==9)

{ $j++;

**continue;**

}

print($j);

$j++;

}

print("<br>");

$k=0;

**do** {

# code...

print($k);

$k++;

} **while ($k<= 10);**

print("<br>");

$arr=array(1,23,43,343,232,1,12);

**foreach ($arr as $value)** {

# code...

print($value);

print("<br>");

}

?>

**ARRAYS**