



PHP PROGRAMMING

UNIT-I

A.BASHEER AHAMED
MSc.,MPhil.,SET.,MBA.,PGDCA.,PGDDM.,
Jamal Mohamed College
Department of Information Technology
Trichy

PHP INTRODUCTION

- PHP stands for Hypertext Preprocessor
- It is a server side scripting
- It can run in client machine
- It is a powerful tool for creating a dynamic web page
- It is a open source scripting language
- It executes on web server(ex Apache web server)

Contd..

- LAMP: Linux Apache My SQL –PHP
- PHP is ready to use with CMS(Content Management System) such as word press,Joomla,Drupal)
- PHP is ready to use e-commerce application(Amazon, Azanta,Zencart,Opencart,drupal)
- PHP is reused to use framework.

Lexical Structure

Lexical Structure

Lexical means rules to follow the usage of languages. Lexical Structure of PHP represents Tokens

Token: Smallest individual unit in a program. The following are the tokens in PHP

- Comment
- Whitespace
- Semicolon
- Variables
- Constants
- Literals
- Delimiters
- Operators
- Keyword

Contd..

Comment: There are two types of comment statements in PHP

Single Line Comment (//)

Multiline Comment (/*.....*/)

Whitespace: It is used to separate tokens.

Semicolons: Each statement in PHP should end with semicolon. It is the terminator

Variables

Variables: Variables are used to store the value

General Form

\$ <Name of variable>

Example : \$a

Contd..

- **Rules for Naming Variables**
 - It can be alphanumeric
 - It should not start with numbers
 - Special symbols are not allowed except underscore
 - White space are not allowed
 - Case sensitive

Contd..

Invalid variable names

(ABC

2_ABC

A b C

A B c

<1>

a_b_c#

Constants

- It is a variable whose value cannot be changed

Syntax: `define (“<Const Name>”,<values>)`

Example: `define (“MAX”,30)`

Literals

- A literal is any notation representing a value in PHP Script

```
$a=12      // Integer
```

```
$b="ABA"   // String
```

```
$c=12.5    //double
```

Delimiters

- It is used to separate boundaries between regions

Example:

{ }

<?php...?>

[]

()

•

Data Types

PHP supports total of 8 data type to define variables.

The following are the eight data types

Integer

Double

Boolean

Null

String

Arrays

Objects

Resources

Note:

Array – An index collection value

Objects-Instance of user defines class which we contains function and data

Resource: Code reference to resource external to PHP.

Expressions in PHP

- It is an important building block of PHP
- It combines with operators and operands to form a result.

Example

`x=a+b`

`X=a++`

`C=b/90`

`i++`

Operators

Operators: It combines with operators and operands to form a result.

The following are the operators in PHP

- Assignment Operators
- Assignment Operators
- Comparison Operators
- Increment/Decrement Operator
- Logical Operator
- String Operator
- Array Operator

Arithmetic Operators

Assume $x=15$ $y=5$

Operators	Expression	Result
+	$x+y$	20
-	$x-y$	10
*	$x*y$	75
/	x/y	3
%	$x\%y$	0

Assignment Operators

Assume $x=15, y=5$

Operators	Expression	Result
=	$x=y$	$x=5$
+=	$x+=y$	$x=x+y=20$
-=	$x-=y$	$x=x-y=10$
=	$x=y$	$x=x*y=75$
/=	$x/=y$	$x=x+/y=23$

Comparison Operators

Assume $x=15, y=5$

Operators	Expression	Result
<code>==</code>	<code>\$x==\$y</code>	True
<code>===</code>	<code>\$x===\$y</code> (Checks the data type)	True
<code>!=</code>	<code>\$x!=\$y</code>	True
<code><></code>	<code>\$x<>\$y</code>	True
<code>!==</code>	<code>\$x!==\$y</code> (Checks the data type)	True
<code>></code>	<code>\$x>\$y</code>	True
<code><</code>	<code>\$x<\$y</code>	False
<code><=</code>	<code>\$x<=\$y</code>	False
<code>>=</code>	<code>\$x>=\$y</code>	True

Logical Operators and String Operators

Assume $x=10, y=5$

Operators	Expression	Result
and (&&)	$x > y$ and $y > x$	False
Or()	$x > y$ or $y > x$	True
.	$x.y$	105
.=	$x.=y$	105

Array Operators

Operators	Explanation	Operators
+	To concatenate	+
==	Equal	==
===	Equal data type	===
!=	Not equal	!=
<>	Not equal	<>

Flow Control Statements

- If
- If..else
- else if
- switch
- while
- do..while
- for
- for each

If statement

```
<?php
echo "Program for voter eligibility"
$a=60;
if($a>=18)
{
    echo "Eligible for vote";
}
?>
```

Program using if else

```
<?php
echo "Program for voter eligibility"
$a=60;
if($a>=18)
{
    echo "Eligible for vote";
}
else
{
    echo "Not eligible for vote";
}
?>
```

Program using if else if

```
.  
<?php  
$a=3;  
$b=3;  
If($a>$b)  
{  
echo "a is big";  
}  
If($a==$b)  
{  
echo "Both are same";  
}  
else  
{  
echo "b is big";  
}  
?>
```

Program using switch

```
<?php
echo " Program for switch";
$x=3;
switch($x)
{
case 1: echo "IIIBSc IT A Students"
        break;
case 2:  echo "IIIBSc IT B Students;
        break;
case 3: echo "IIIBSc IT C Students

default: echo "Enter choice between 1-3";
?>
```


Program using while loop

```
<? php
$n=10,$i=0;
while ($i<10)
{
    echo "$i" + "<br>"
}
?>
```

Program using do while

```
<? php
$n=10,$i=0;
do
{
    echo "$i" + "<br>"
    $i++;
} while ($i<10)

?>
```

Program using for loop

```
<?php
for ($i=1;$i<10;$i++)
{
    echo "$i";
}
```

Program using for each

for each statement used only with Arrays

```
<?php
    $fruits=array('Apple','orange','Grapes');
    for each($fruits as fruit)
    {
        echo $fruit;
        echo "<br>"
    }
?>
```

Terminating loop entity

```
<?php
$i=1;
$fruits=array('Apple','Orange','Grapes');
for each($fruits as $fruit)
{
    If($i==2)
    {
        echo $fruit
        break;
    }
    i++;
}
```

Functions

Functions

- Set of codes will be placed in function to perform specific task.

Syntax

```
function <function name>()  
{  
    <Set of codes>  
}
```

Example1

```
<?php  
function Bat()  
{  
    echo "Batting Star Virendar Sehwan"  
}  
Bat();  
?>
```

Example 2

```
<?php
function Bat()
{
    echo "Batting Star Virendar Sehwaq"
}
function Bowl()
{
    echo "Bowling Star Rashedkhan"
}

Bat();
Bowl()
?>
```


Example3

```
<?php
function counter()
{
static $count=0;
return $count++;
}
for ($i=1;$i<5;$i++)
{
    Print counter();
}
```

Operator Precedence

The order in which operation and expression evaluated depends on their relative precedence.

^				
*	/	%		
+	-			
<	<=	>	>=	
=	==	===	>=	
&&				
and	or			