INTRODUCTION TO PHP

PREPARED BY: ARNIKA PATEL

LECTURER

PARUL INST. OF ENGG. & TECH.-DS

INTRODUCTION TO PHP

- Php firstly called **Personal Home Page**
- It will then changed to **Hypertext Preprocessor**
- **Used to develop dynamic web page**
- **Extension of file .php**
- **Loosely typed language:** no need to define type of variable
- **Installation includes 1st to install web server, then install php, lastly install database MYSQL**

Php Start and End Tags

- Standard php tags:

  <?php /*….*/ ?>

- Script Tags:

  <script language="php">…</script>

- Short Tags:

  <? /*…*/ ?>

- ASP style Tags:

  <% /*….*/ %>

Comments in Php

- Single line Comment

```
// This is a single-line comment
# This is also a single-line comment
```

- Multi line Comment

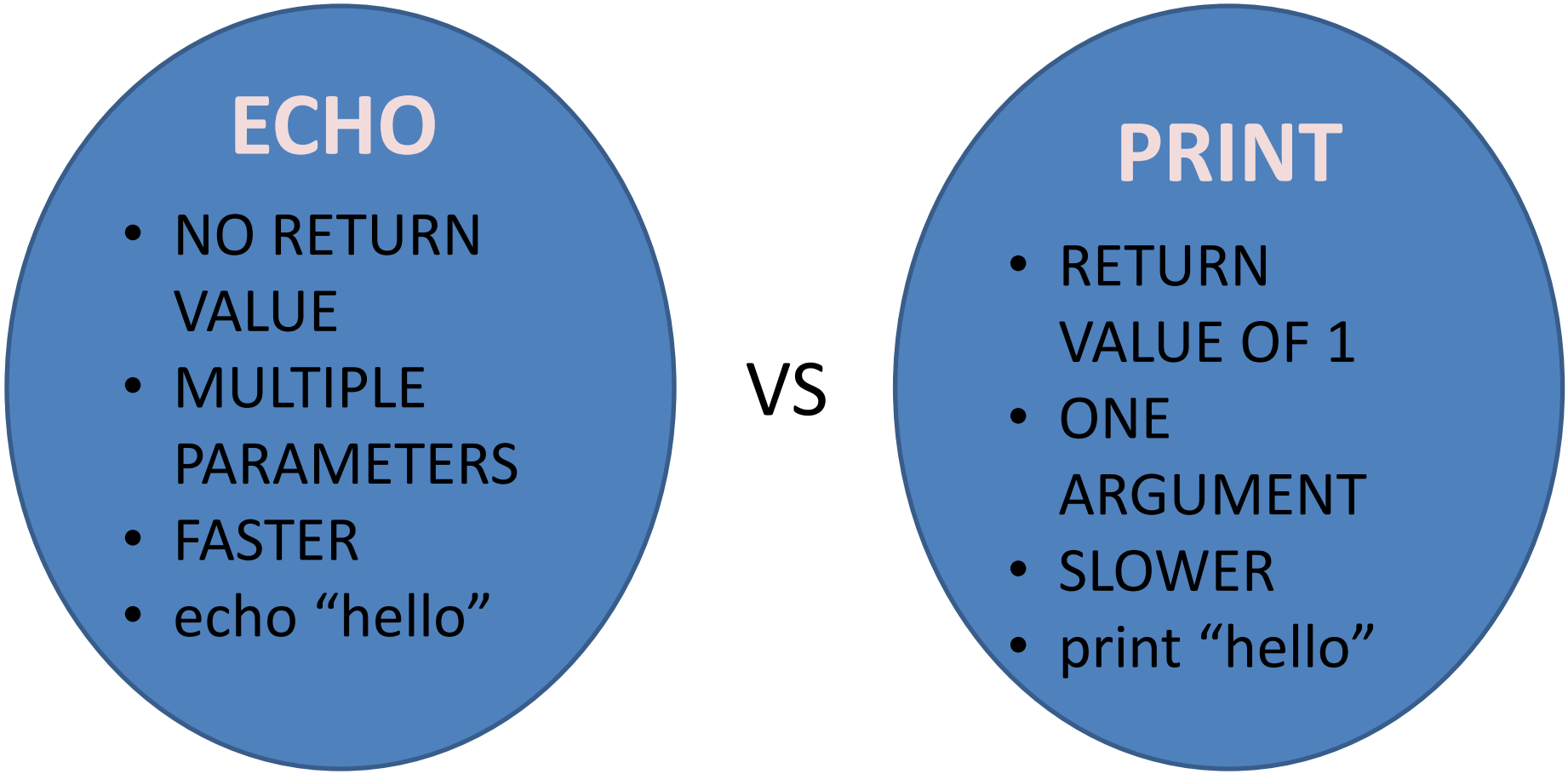
```
/*
        This is a multiple-lines comment block
        that spans over multiple
        lines
*/
```

Php Variable

- Rules:
  - **Starts with '$' sign**
  - **After '$' sign it must start with '_' or letter**
  - **Only alpha-numeric and underscore allowed**
  - **Case-sensitive**
- Example:

  $name, $_name valid variable names

Php Echo and Print Statement



Php DataTypes

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

Php Constant

- Can’t changed
- Syntax:

  define(*name, value, case-insensitive*);

- Example:

  define(“pie”,3.14);

Php Operators

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators

Php Operators

- Arithmetic operators
  - Addition
    $x + $y
  - Subtraction
    $x - $y
  - Multiplication
    $x * $y
  - Division
    $x / $y
  - Modulus
    $x % $y
  - Exponentiation
    $x ** $y

Php Operators

- Assignment operators

  $x = $y

  $x += $y ($x=$x+$y)

  $x *= $y ($x=$x*$y)

  $x /= $y ($x=$x/$y)

  $x %= $y ($x=$x%$y)

Php Operators

- Comparison operators
- **Equal ($x==$y)**
- **Identical ($x===$y)**
- **Not Equal ($x!=$y or $x<>$y)**
- **Not Identical ($x!==$y)**
- **Greater Than ($x>$y)**
- **Less Than ($x<$y)**
- **Greater Than Equal to ($x>=$y)**
- **Less Than Equal to ($x<=$y)**

Php Operators

- Increment/Decrement operators
- Pre-increment

  $++x

- Post-increment

  $x++

- Pre-decrement

  $--x

- Post-decrement

  $x--

Php Operators

- Logical operators
- AND

  $x and $y
- OR

  $x or $y
- XOR

  $x or $y
- &&

  $x && $y
- ||

  $x || $y
- !

  !$x

Php Operators

- String Operator
- Concatenation

  $x . $y

- Concatenation assignment

  $x .= $y

Php Flow Control Statements

- **if**
- **if…else**
- **if…else if…else**
- **switch**

Php Flow Control Statements

- **if**

  if (*condition*) {
  code;
  }

- **Example:**

```
<?php
$x=2;
If($x<1){echo “Hello World”;}
?>
```

Php Flow Control Statements

- **If…else**

```
if (condition) {
        code;
} else {
        code;
}
```

- **Example:**

```
<?php
$x=2;
If($x<1){echo “Hello World”;}
else{“condition false”;}
?>
```

Php Flow Control Statements

- **If…else if…else**

```
if (condition) {
        code;
} else if(condition){
        code;
}else{
        code;
}
```

- **Example:**

```
<?php
$x=2;
If($x<1){echo “1st condition”;}
else if($x>2){echo “condition second”;}
else {echo “condition false”;}
?>
```

Php Flow Control Statements

- **Switch**
- switch (*n*) {
  case *label1:*
  code if n=label1;
  break;
  case *label2:*
  code if n=label2;
  break;
  case *label3:*
  code if n=label3;
  break;
  ...
  default:
  code if n not match;
  }

Php Flow Control Statements

- **Example:**

```
<?php
$favFlower = “Rose";

switch ($favFlower) {
   case “rose":
      echo “Rose!";
      break;
   case “Lotus":
      echo “Lotus!";
      break;
   case “Lily":
      echo “Lily!";
      break;
   default:
      echo “No One!";
}
?>
```

Php Loops

- **While**
- **Do…While**
- **For**
- **Foreach**

Php Loops

- **While**

  while (*condition is true*) {
  code
  }

- **Example:**

```
while($x <= 8) {
   echo "$x <br>";
   $x++;
```

Php Loops

- **Do…While**

```
do {
        code;
} while (condition is true);
```

- **Example:**

```
do {
   echo "$x";
   $x++;
} while ($x <= 8);
```

Php Loops

- **For**

```
for (initialization;test;increment) {
        code;
}
```

- **Example:**

```
for ($x = 0; $x <= 5; $x++) {
        echo "$x <br>";
}
```

Php Loops

- **Foreach**

```
foreach ($array as $value) {
        code;
}
```

- **Example:**

```
$flowers=(“rose”,”lotus”,”lily”)
foreach ($flowers as $value) {
        echo "$value <br>";
}
```

Php Arrays

- **Indexed Arrays**

  $flowers=array(“lotus”, “lily”, “rose”);

- **Associative Arrays (“key”=> “value”)**

  $age = array(“om”=>“15”, “rim”=> “37”);

- **Multidimensional Arrays**

```
$colors = array
(
        array(“rose",45,8),
        array(“lily",12,5),
        array(“rose",5,3),
);
```

Php Functions

- **User Defined Function without argument**

```
function functionName() {
        code;
}
```

Rule: function name can start with a letter or underscore

function names are not case-sensitive

```
<?php
        function hello()
        {
                echo “hello world”;
        }
        hello();          //function calling
?>
```

Php Functions

- **Function with Argument**

```
<?php
function people ($name, $age) {
    echo "$name:$age <br>";
}
people(“mark”, “20”);
?>
```

Php Functions

- **Function with Default Argument Value**

```
<?php
        function age($a=20)
        {
                echo “age is $a”;
        }
        age(30);        //age=30
        age();          //age=20
?>
```

Php Functions

- **Function with Return Value**

```
<?php
        function add($x,$y)
        {
                $z=$x+$y
                return $z;
        }
        $ans=add(4,6);
        echo “addition is: ”.$ans;
?>
```