

1. Assigning variables:

```
<?php
$var1="Azizul";
$var2=21.567;
?>
```

2. Concatinating variables:

```
<?php
$var1="Azizul";
$var2=21567;
echo $var1.'s salary is'.$var2
?>
```

Output:

Azizuls salary is21567

3. == and === operator:

```
<?php
/* comparison operation */
$var1='1';
$var2=1;
if($var2===$var1)
{
    echo "ok";
}
else echo "Not ok";
```

Output:

C:\xampp\php\php.exe C:\xampp\htdocs\phppractise\index.php

Not ok

4. do while loop:

```
<?php
/* do while loop operation*/
$var1='1';
do
{
    echo $var1." hello<br> ";
    $var1++;
}
while ($var1<3);
```

Output:

1 hello

2 hello

5.for loop:

```
<?php
/* for loop operation*/

for($i=1;$i<21;$i++)
{
    echo $i."<br>";
}
```

Output:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

6.switch loop:

```
<?php
/* switch loop operation*/
```

```
$var="friday";
switch($var)
{
    case "friday":
    case "saturday":
        echo "It is weekend";
        break;
    default:
        echo "It is working day";
}
```

Output:

It is weekend

7.function operation in php:

```
<?php
/*welcome to function*/
function addResult($num1,$num2)
{
    $num3=$num1+$num2;
    return $num3;
}
echo addResult(10,5);
```

8.Global variable declaration in php:

```
<?php
/*welcome to function*/
$num1=10;$num2=11;
function addResult()
{
    global $num1,$num2;
    $num3=$num1+$num2;
    return $num3;
}
echo addResult();
```

Output:

21

9.string_word_count operation:

*print_r is used for full array displaying.echo cannot be used in this case.

```
<?php
$string="I am Azizul .";
$string word count=str word count($string,1,'. ');
print_r($string word count);
```

Output:

Array ([0] => I [1] => am [2] => Azizul [3] => .)

10.different string related function operaton:

```
<?php
$string="I am Azizul & hakim .";
$string_word_count=str_word_count($string,1,'. ');
$string_half=str_shuffle($string);
$substr=substr($string,0,6);
$string_reversed=strrev($string);
print_r($string_word_count);
echo $string_half;
echo strlen($string);
echo $string_reversed;
```

Output:

Array ([0] => I [1] => am [2] => Azizul [3] => hakim [4] => .) ai zzhim& k. ImulAa21.
mikah & luzizA ma I

11.similiar_text function:

```
<?php
$string="I am Azizul & hakim .";
$string2="I am Zahir & hasib";
similar_text($string,$string2,$result);
$string_word_count=str_word_count($string,1,'. ');
$string_half=str_shuffle($string);
$substr=substr($string,0,6);
$string_reversed=strrev($string);
echo $result."<br>";
print_r($string_word_count);
echo $string_half;
echo strlen($string);
echo $string_reversed;
```

Output:

61.53.....

12.trim function:

```
<?php
$string="&nbsp;I am Azizul & hakim .&nbsp;";
$trimmed_string=trim($string);
```

```
echo $trimmed_string;
```

Output:

```
I am Azizul & hakim .
```

13.array:

*print_r is used for full array displaying.echo cannot be used in this case.

```
<?php
$arr=array("apple","orange","lemon");
print_r($arr);
print_r($arr[2]);
echo $arr[2]
```

output:

Array ([0] => apple [1] => orange [2] => lemon) lemonlemon

14.Assigning array by string:

```
<?php
$arr=array("apple"=>200,"orange"=>215,"lemon"=>567);
echo $arr["apple"];
```

Output:

200

15. Multidimensional array:

```
<?php
$arr=array(0=>array("chicken","beef","mutton"),1=>array("tomato","carrot"));
echo $arr[0][1];
```

Output:

beef

16.foreach loop operation:

When you are using **for** loop you know how many times this loop will execute. Its' prefix

But while using **foreach** loop we don't have to think about the how many times this will execute. In case of **foreach** loop there is no extra memory required for the counters also. Because we don't have any counter in foreach loop.

Conclusion: **foreach** loop mostly we use for dynamic array. Because we don't know about counter. **for loop** always use when I know counter

Ref: <https://www.quora.com/What-is-the-difference-between-for-and-foreach-in-php>

16.foreach loop:

```
<?php
$arr=array("meat">array("chicken","beef","mutton"),"vegetable">array("tomato","carrot"));
foreach ($arr as $meat=>$veg)
{
    echo "<strong>".$meat."</strong><br>";
    foreach ($veg as $vegetable)
    {
        echo $vegetable."<br>";
    }
}
```

Output:

meat

chicken

beef

mutton

vegetable

tomato

carrot

17.strtolower function:

```
<?php
if(isset($_GET['user_name']) && !empty($_GET['user_name']))
{
    $user_name=$_GET['user_name'];
    $user_name_lc=strtolower($user_name);
    if($user_name_lc=='azizul')
    {
        echo "You are the best " . $user_name;
    }
}
?>
<form action="index.php" method="GET">
    Name:<input type="text" name="user_name"><br>
    <input type="submit" value="Submit">
</form>
```

Output:

You are the best aZiZul

Name:

18.strpos function:

```
<?php
$string="I am Azizul hakim.I am a student.I am feeling good today";
```

```
$find="am";  
$i=0;  
$find_length=strlen($find);  
while ($pos=strpos($string,$find,$i))  
{  
  
    echo $pos."<br>";  
    $i=$pos+$find_length;  
}
```

Output:

2

20

35

19-rest of the part are included in git commit with codes in the github.

Github project link: <https://github.com/Azizul31/phppractise>

Prepared by,
Azizul Hakim Chowdhury
CSE'14,CUET