Complete Php Course Pdf Files Ducat 2020 (Create by Kapil Sir)

Writing by - Saurabh shukla (7 may 2020)

****

PHP Course 2020 Ducat Noida

PHP : PHP stands for hypertext preprocessor. It is an open source server side scripting language which is used for developing dynamic websites. It was developed by Rasmus Lerdrof around 1994.

Extension : .php

Advantage

* open source(source code can change)
* multiplatform(win,linux,mac)
* support many databases (e.g mysql, oracle etc)
* support many web servers (e.g. apache, iis etc)
* easy to use, light weight, fast processing
* support oops features

php
xampp (Cross)
wamp (windows)
lamp (linux)
mamp (mac)

xampp download from apachefriends.org
install

xampp control panel (start apache and mysql) save file

```
 c://xampp/htdocs/folder/filename.php
```

e.g.

```
 c://xampp/htdocs/06pm/first.php
```

code

```
-----
-----
-----
```

Run (browser)

http://localhost/06pm/first.php

or

localhost/06pm/first.php

```
----------------------------------------------------
```

Tags(Delimiters)

1 standard tag

```
 <?php

 ?>
```

2 short tag

```
 <?

 ?>
```

3 Asp tag

```
 <%

 %>
```

for output on screen

echo

* fast processing
* it can execute multiple statement

e.g.

```
echo "","","";
echo  "Hello","Ducat";
```

print

* slow processing
* it can execute only single statement

e.g.

```
print "";
print  "Hello";
```

concatination (.)

```
echo   "Hello"."Ducat";
print   "Hello"."Ducat";
```

new line

<br> (webpage)

\n (file)

```
echo "Hello <br> PHP";
print "Hello <br> PHP";
echo "<h1>India</h1>";
```

comments

```
 //   single line comments
 #    single line comments


 /*

     multiple line comments

 */
```

variable : it is a temporary memory location which used for hold the information. the value of variable can vry(change) at run time.

Rules for declare variable

* start from $ sign

* continue with a-zA-Z_

* can combination a-zA-Z0-9

e.g.

```
 $a;
 $D;
 $5;     wrong
 $_A;
 $a5;
```

$5a;   wrong

```
$a = 5;
echo $a;
```

```
$b = "India";
echo $b;
```

Data Types : in php data types allocate automatically at run time according to variable value.

int, string, double, array, object , resource, boolean, null

check data type

gettype(var name)   show only data type

var_dump(var name)   print, show data type and value

e.g.

```
$a = 5;
echo gettype($a);
var_dump($a);
```

e.g

```
$x = 5;
echo "value of x is ".$x;
```

e.g.

```
$a = 5;
$b = 3;
echo "value of a is ".$a." and value of b is ".$b;
```

e.g.

```
$z = 8;

echo "value of z = ".$z;     (value of z is 8)
echo "value of z = $z";       (value of z is 8)
echo 'value of z = $z';       (value of z is $z)
```

Constant : it is a temporary memory location which used for hold the information. the value of constant can not vry(change) at run time.

```
defined("constant name","value");
const constantname = value;
```

Rules for define constant name

- * always start from a-zA-Z_
- * can use a-zA-Z0-9

e.g

```
define("HOST","Localhost");
or
const HOST = "Localhost";

echo HOST;
```

e.g.

```
  define("A","India");
  or
  const A = "India";

  echo A;
```

Operators : It is a special symbol whose meanings are predefined. They work according to their predefined meaning.

Type of Operators

1 Arithmetic Operator

+	Addition
-	Substraction
*	Multiplication
/	Division
%	Modulus

e.g.

```
$a = 10;
$b = 3;

$c = $a+$b;      // output 13
echo $c;

$c = $a/$b;      // output 3

$c = $a%$b;      // output 1
```

2 Assignment Operator

It used for shift the value from right side to left side

```
=
+=
-=
*=
/=
%=
```

e.g.

```
$a = 5;
$b = 2;

$a = $a+$b
  or
$a += $b;


$a = $a-$b;
$a -= $b;
```

3 Comparision Operator

==	match only value
===	match value and data type
!=	match only value
!==	match value and data type

```
$a = 5;
$b = "5";

$a == $b;
$a === $b;
```

4 Increment Operator

++ Increment By 1
-- Decrement By 1

```
$a = 5;
$a++;
echo $a;   // Output 6
```

```
$a = 5;
++$a;
echo $a;   // Output 6
```

```
$a = 5;
echo $a++; // Output 5
echo $a;   // Output 6
```

```
$a = 5;
echo ++$a; // Output 6
echo $a;   // Output 6
```

```
$a = 5;
$a++;
echo $a;   // Output 6
echo --$a; // Output 5
echo $a;   // Output 5
```

5 Relational Operator

> Greater than
>= Greater than equal too
< Less than
<= Less than equal too

e.g.

```
$a = 5;
$b = 5;

$a >= $b;
```

6 Logical Operator

&& AND
|| OR
! NOT

($a>$b && $a>$c) Both Condition True

($a>$b || $a>$c) At Least One Condition True

7 Conditional Operator

Ternary Operator
Question Colon Operator

syntax

(condition ? "True Statement" : "False Statement")

e.g.

```
$a = 5;
$b = 3;
```

```
echo ($a>$b ? "A is Greater" : "B is Greater");
```

Conditional Statement

1 if

2 if else

3 if else if

4 switch case

Q wap to store a 3 digit number in a variable, check that number is palindrome or not ?

```
$no = 134;
```

Q wap to store 3 digit number in a variable, check that is it armstrong or not ?

```
$no = 153
```

Loop : Loop is used for execute same statement multiple times.

Types

1 for

syntax

```
for(initialize; condtion; inc/dec)
{
  code
}
```

e.g

```
for($i=1; $i<=5; $i++)
{
  echo "Hello";
}
```

e.g

```
for($i=5; $i>=1; $i--)
{
  echo "Hello";
}
```

2 while

syntax

```
initialization;

while(condition)
{
  code
  inc/dec
}
```

e.g.

```
  $i=1;

  while($i<=5)
  {
    echo "Hello";
    $i++;
  }
```

3 do while

syntax

```
  intitialization

  do
  {
    code

    inc/dec
  }while(condition)
```

e.g.

Q Wap to print factorial of a given number.

```
 $no = 5;
```

Q. wap to print fibonacci series upto 10 numbers.

0 1 1 2 3 5 8 ....

Q. wap to check that given number is palindrome or not.

$no = 121;

Q. wap to check that given number is armstrong or not ?

$no = 153;

Jump Statement

1 break

2 continue

3 exit

```
*****
****
***
**
*
```

```
*****
****
 ***
  **
   *
```

```
1
12
123
1234
12345
```

```
  1
 22
 333
4444
55555
```

```
abcd
abcd
abcd
```

```
01 03 05 07
09 11 13 15
17 19 21 23
25 27 29 31
```

```
02 04 06 08
10 12 14 16
18 20 22 24
26 28 30 32
```

Q write a program to print prime number upto 100;

Q write a program to print table till given number.

$no = 4;

1........

2........

3........

4........

```
  *
 * *
 * * *
* * * *
* * * * *
```

```
  *
 ***
 *****
*******
*********
```

```
*********
*******
 *****
 ***
  *
```

```
*
*
*
*
* * * * *
```

```
*****
```

```
 *
 *
 *
 *
```

```
*  *
*  *
****
*  *
*  *
```

```
*   *
 * *
  *
 * *
*   *
```

```
****
  *
****
  *
****
```

(i,j)
(1,1) (1,2) (1,3) (1,4) (1,5)

```
* * * * *
```

(2,1) (2,2) (2,3) (2,4) (2,5)

```
* * * * *
```

```
(3,1) (3,2) (3,3) (3,4) (3,5)
* * * * *

(4,1) (4,2) (4,3) (4,4) (4,5)
* * * * *

(5,1) (5,2) (5,3) (5,4) (5,5)
* * * * *


===================================================================
=======
```

```
01 03 05 07
03 05 07 09
05 07 09 11
07 09 11 13
```

```
   1
  2 2
 3 4 3
4 5 6 4
5 7 8 9 5
```

```
   1
  2 2
 3 4 3
4 5 6 4
5 6 7 8 5
```

n=3

```
{*}
 {{*}}
 {{{*}}}
```

n=4

```
{*}
 {{*}}
 {{{*}}}
   {{{{*}}}}
```

n=3

```
1
22
22
333
333
333
```

n=4

```
1
22
22
333
333
333
4444
4444
4444
4444
```

===

Array : Array is a collection of element, which used for store multiple value in single location

type of array
1 index array (numbered array)
2 associative array
3 multidimensional array

1 Index array
In this array index(keys) of array always in numeric value and it start from 0

syntax

```
  array name = array(value,value,value,value);

        or

  array name = [value,value,value,value];
```

e.g.

```
  $arr = array(45,"red",82.6);

      or

  $arr = [45,"red",82.6];

    or

  $arr[0] = 45;
  $arr[1] = "red";
  $arr[2] = 82.6;



  echo $arr[0];
```

size of array

```
  sizeof(arrayname)
  count(arrayname)

  echo sizeof($arr);
```

print_r(arrayname); print array index and values

2 Associative Array

In this array index(keys) of array can alpha or numeric or alphanumeric

syntax

```
  arrayname = array('index'=>'value','index'=>'value')

     or

  arrayname = ['index'=>'value','index'=>'value'];
```

e.g.

```
  $arr = array('a'=>'red','8'=>'95.6','c3'=>'99');

     or

  $arr = ['a'=>'red','8'=>'95.6','c3'=>'99'];

     or
```

```
$arr['a'] = 'red';
$arr[8] = 95.6;
$arr['c3'] = 99;
```

3 Multidimensional array

Array within array is called multidimensional array

```
$arr = array(array(4,5,6),array(7,8,9));
```

or

```
$arr = [[4,5,6],[7,8,9]];
```

```
echo $arr[0][1];
```

```
$arr =
['India'=>['capital'=>'Delhi','currency'=>'Ruppee'],'USA'=>['capital'=>'Washington','currency'=>'Dollar'],'Pakistan'=>['capital'=>'Islamabad','currency'=>'Ruppee']];
```

or

```
$arr['India']['capital'] = "Delhi";
```

File Handling : It is a process in which we can store and read information in files and folders.

```
scandir("directory name")  // scan directory and store all names in array
```

```
mkdir("directory name")      // make directory
rmdir("directory name")      // remove directory
is_dir("directory name")   // check directory exist or not
```

```
touch("filename.ext")     // create filename
unlink("filename.ext")    // remove filename
file_exists("filename.ext")    // check file exist or not
```

```
fopen("filename.ext","mode");
mode : w,r,a;
```

Write

```
$fo = fopen("abc.txt","w");

fwrite($fo,"Data");    // write data in file
fputs($fo,"Data"); // write data in file
```

Read

```
$fo = fopen("abc.txt","r");

fread($fo,3);     // read file until given number of characters
filesize("filename.txt")   // count number of characters in file
fread($fo,filesize("abc.txt"))    // read whole file
```

```
fgets($fo)   // read line by line
fgetc($fo)   // read char by char
```

```
while(!feof($fo))
{
  echo fgetc($fo);
  echo "<br>";
}
```

```
file_put_contents("xyz.txt","Delhi\nNoida");   // create file if not exist, open file in w model and write data
```

```
file_get_contents("xyz.txt");  // open file in r mode and read whole file
```

```
fclose($fo)      // close file
```

State Management : It used for move information from one location to another location.

1 Session   (Server Side)

2 Query String (Client Side)
3 Hidden Fields (Client Side)
4 Cookies (Client Side)

1 Session : It work Server Side so It is more Secure, It used for track the information untill we found session environment, by default session time out is 24 minutes.

*** start**

```
 session_start();
```

*** create**

```
 $_SESSION['session name'];
 e.g.
 $_SESSION['user'] = 'abc@gmail.com';
```

*** Read**

```
 $_SESSION['session name'];
 e.g.
 echo $_SESSION['user'];
```

*** Destroy**

```
 session_destroy();   // destroy all session

 unset($_SESSION['session name'])  // destroy particular session
 e.g.
   unset($_SESSION['user']);
```

2 Query String : It used for move information from one page to another page by url

Send

```
   <a href="abc.php?obj=123&&obj2=321">Click</a>
      or
   header("location:abc.php?obj=123&&obj2=321");
```

Receive

```
   $_GET[];
   $_REQUEST[];


   echo $_GET['obj'];
   echo $_GET['obj2'];
```

3 Hidden Field : It is a form input field which used for store the information. It does not display on web page but we can access the information of hidden field

```
<input type="hidden" name="" value="">
```

4 Cookies : cookies are the text files which store on browser. it used for store small information. The maximum size of theses files 4kb.
20 cookies can create on system by single website.
300 cookies can store on browser

Type of Cookies
  Persistence (Permanent)
  Non Persistence(Temporary)

* create cookie

```
setcookie("cookie name","value")        Non Persistence
setcookie("cookie name","value","time")      Persistence

e.g.
setcookie("useremail","abc@gmail.com")    Non Persistence
setcookie("useremail","abc@gmail.com",time()+60*60)    Persistence
```

* Read

```
$_COOKIE['cookie name'];

e.g.
echo $_ COOKIE['useremail'];
```

* Destroy

```
setcookie("useremail","",time()-60*60)
```

File uploading (single and multiple files uploaded)

File Uploading : it is a process to upload file on server. when we upload file on server, first it goes in temporary folder. By default upto 2Mb file can upload.

```
<form method="post" enctype="multipart/form-data">
            Upload : <input type="file" name="att">
            <input type="submit" name="sub">
</form>
```

There are five super global array varibales which used for read file information

```
1 File name                              :           $_FILES['fieldname']['name'];
```

2 Temporary Name of File : $_FILES['fieldname']['tmp_name'];
3 Size Of File : $_FILES['fieldname']['size'];
4 Type Of File : $_FILES['fieldname']['type'];
5 Error In File : $_FILES['fieldname']['error'];

```
move_uploaded_file(temporary name, "path/".filename);
```

e.g.

```
move_uploaded_file($_FILES['fieldname']['tmp_name'],"foldername/".$_FILES['fieldname
'][''name']);
```

Note : when you will code then make sure that you have create destination folder for file (images) otherwise it will show error

----

With Examples : single upload files and multiple upload files :

Code is thats :

Single upload files ⇒

```
<?php

extract($_POST);

if(isset($sub))
{
	$fn = $_FILES['att']['name'];
	$tmp = $_FILES['att']['tmp_name'];

	$arr = explode('.',$fn);
```

```
        $ext = end($arr);
        $ext = strtolower($ext);

        if($ext=="jpg" || $ext=="jpeg")
        {
                $fnn = rand().'.'.$ext;
                //$fnn = uniqid().'.'.$ext;
                //$fnn = rand().$fn;

                if(move_uploaded_file($tmp,"images/".$fnn))
                {
                        echo "File Upload Successfully";
                }
        }
        else
        {
                echo "Only jpg or jpeg file support";
        }
}


?>
<html>
        <head>
                <title>File Uploading</title>
        </head>

        <body>
                <form method="post" enctype="multipart/form-data">

                        Upload : <input type="file" name="att" required>
                        <br>
                        <input type="submit" name="sub" value="Submit">

                </form>
        </body>
</html>

------------------------------------------------------------------------------------------------
```

Multiple uploaded files ⇒

```
<?php

extract($_POST);

if(isset($sub))
{
     $fn = $_FILES['att']['name'];
     $tmp = $_FILES['att']['tmp_name'];

     //echo "<pre>";
     //print_r($fn);
     //print_r($tmp);
     //echo "</pre>";

     $i = 0;
     $count = 0;

     foreach($tmp as $t)
     {
          if(move_uploaded_file($t,"images/".$fn[$i]))
          {
               $count++;
          }
          $i++;
     }

     if($count > 0)
     {
          echo "File Upload Successfully";
     }
}
```

```
?>
<html>
     <head>
          <title>File Uploading</title>
     </head>

     <body>
          <form method="post"
enctype="multipart/form-data">

               Upload : <input type="file" name="att[]"
multiple required>
               <br>
               <input type="submit" name="sub"
value="Submit">

          </form>
     </body>
</html>
```

Database in PHP Course ⇒

Database : Database is a collection of data where we can store information in oragnised way. and we can retreive the information easier and faster.

Mysql : it is an open source database. which used for store the data in oragnised way.

Data Types :

tinyint, int, bigint, float, decimal (numeric values)

char, varchar, text, medium text ,long text (string)

date, time, timestamp (date and time)

blob, medium blob, long blob (files)

Constraints (Restrictions of table fields)

1 primary key

* unique record (no duplicate record)
* only 1 can use in a table
* can not null

2 unique key

* unique record (no duplicate record)
* More than 1 can use in a table
* can null

3 not null

4 default

```
create table employee (eid int(11) primary key, name varchar(255) not null, email varchar(255) unique key, status char(1) default 'd');
create table employee (eid int(11) primary key
auto_increment, name varchar(255) not null, email
varchar(255) unique key, status char(1) default 'd');
```

5 Foreign Key

```
     create table account(id int(11), salary int(11) not
null, foreign key(id) references employee (eid));
```

6 Composite Key

```
     create table product(cat varchar(255), subcat
varchar(255), primary key(cat,subcat));
```

```
hostname = "localhost";
username = "root";
password = "";
```

Database connectivity with cmd (command prompt)

* open cmd

* D: (if xampp in d drive)

* cd.. (one step back)   or   cd\ (on root)

```
* cd xampp\mysql\bin

* mysql.exe -u root;
```

DDL (Data Definition Language)

Create, Alter, Drop

```
create database dbname;
e.g create database 1230pm;
```

```
show databases (show name of all databases)
```

```
use dbname;
eg.  use 1230pm;
```

```
create table.............................................
```

```
show tables; (show all tables name);
```

```
desc tablename;
desc employee; (description)
```

Alter

add column

```
*    alter table employee add address varchar(255) not null;          // insert in last

*    alter table employee add address varchar(255) not null after email;   // insert after email
```

Change data type of column

```
*    alter table employee modify address char(50) not null;
```

rename column name

```
*    alter table tablename change oldname newname datatype;

*    alter table employee change name fname varchar(255);
```

rename table

```
* rename table oldname to newname;
* rename table employee to students;
```

Drop

drop column

```
*    alter table employee drop address;
```

drop table

```
*    drop table employee;
```

drop database

*   drop database 1230pm;

****

DML (Data Manipulation Language)

insert, update, delete

Insert

*   insert into employee values (1,'abc',28000);

*   insert into employee (eid,fname,salary) values (1,'abc',28000);

*   insert into employee set eid=1, 'fname'='abc', salary=28000;

*   insert into employee (eid,fname,salary) values (1,'abc',28000),(2,'xyz',45000);

Update

*   update employee set salary=35000 where eid=1;

Delete

*   delete from employee;   // all record delete

```
*    truncate table employee;      // all record delete
*    delete from employee where eid=2;  // delete
particular record
```

Select

```
select * from employee;

select eid, fname from employee;

select * from employee where eid=2;

select * from employee where eid in(4,8,9);

select * from employee where salary>25000;

select * from employee where salary between 25000 and
40000;

select * from employee where eid>5 and salary>25000;

select * from employee where eid>5 or salary>25000;

select * from employee where name like 'a%';      // first
letter a in name

select * from employee where name like '_a%';          //
second letter a in name

select * from employee where name like '%a%';          //
any where a in name
```

```
select * from employee where name like '%a';      // last letter a in name

select * from employee order by salary;      // by default ascending order

select * from employee order by salary desc;      // descending order
```

Annual Salary

```
select eid, fname, salary*12 from employee;
```

Alias

```
select eid,fname, salary*12 as annual_salary from employee;
```

limit

```
Select * from employee limit 3;    // get starting 3 records  (1 to 3)
select * from employee limit 3,5;  // get 5 record after starting 3 records (4 to 8)
```

Aggregate Function

```
max()
min()
count()
```

```
upper()
lower()
sum()
concat()
```

```
select  max(salary) from employee;
select  min(salary) from employee;
select count(name) as total_employee from employee;
select eid, upper(name) from employee;
select sum(salary) from employee;
select concat(fname,'',lname) as full_name from employee;
```

autoincrement

```
     int, primary key,
```

enum

```
create table users(id int(11) primary key, name
varchar(255) not null, gender enum('male','female') not
null);
```

```
insert into users (id,name,gender) values (1,'abc','male');
insert into users (id,name,gender) values (1,'abc',1);
```

set

```
create table employee(id int(11) primary key, name
varchar(255) not null, city set('delhi','noida') not null);
```

```
insert into employee (id,name,gender) values
(1,'abc','delhi');
insert into employee (id,name,gender) values
(2,'xyz','noida');
```

```
insert into employee (id,name,gender) values
(3,'amit','delhi,noida');
```

Sql Query joining and all ⇒

Joins : A Join enables you to retrieve records from two (or more) logically related tables in a single result set.

1 .Inner Join

Select employee1.id, employee1.name, employee2.email from employee1 inner join employee2 on employee1.id = employee2.id;

Select employee1.id, employee1.name, employee2.email from employee1 inner join employee2 using(id);

Select e1.id, e1.name, e2.email from employee1 e1 inner join employee2 e2 on e1.id = e2.id;

Select e1.id, e1.name, e2.email from employee1 e1 inner join employee2 e2 using(id);

2. Left Join (all records of left table and common record of right table)

Select e1.id, e1.name, e2.email from employee1 e1 left join employee2 e2 on e1.id = e2.id;

3 Left Join(all records that is only in left table)

Select e1.id, e1.name from employee1 e1 left join employee2 e2 on e1.id = e2.id where e2.email is null;

4 Right Join (all records of Right table and common record of Left table)

```
Select e1.id, e1.name, e2.email from employee1 e1 right join employee2 e2 on e1.id =
e2.id;
```

5 Right Join (all records that is only in Right table)

```
Select e2.id,  e2.email from employee1 e1 right join employee2 e2 on e1.id = e2.id where
e2.name is null;
```

6 Outer Join (Show all record of both tables except common records)

```
Select e1.id, e1.name, e2.email from employee1 e1 left join employee2 e2 on e1.id = e2.id
where e2.id is null union Select e2.id, e1.name, e2.email from employee1 e1 right join
employee2 e2 on e1.id = e2.id where e1.id is null;
```

7 Full Outer Join (Show All Record of Both Tables)

```
Select e1.id, e1.name, e2.email from employee1 e1 left join employee2 e2 on e1.id = e2.id
union select e2.id, e1.name, e2.email from employee1 e1 right join employee2 e2 on
e1.id = e2.id ;
```

UNION

```
Select id, name from hr union select id,  name from acc;
```

UNION ALL

```
Select id, name from hr union all select id,  name from acc;
```

Distinct (Remove Duplicates)

```
select * from acc;
select city from acc;
select distinct city from acc;
```

Highest Salary and subquery (Query inside Query)

```
select distinct salary as 4th_heigest_salary from order by salary desc limit 3,1;
```

select max(salary) as 2nd_heigest_salary from user where salary <(select max(salary) from user);

Group By and Having Clause

select dept , count(name) as total_employee from emp group by dept having count(name)>1;

Most Important Function How to connected database in Php ⇒

Database Connectivity
1 MYSQL
* mysqli procedural connectivity
* mysqli object based connectivity
2 PDO (PHP DATA OBJECT)
It can connect with more than 10 databases

----

hostname = "localhost"
username = "root"
password = ""

mysqli procdural connectivity

1 Database Connect
syntax

connection name = mysqli_connect("hostname","username","password","dbname") or die(mysqli_connect_error());

e.g.

$link = mysqli_connect("localhost","root","","2pm") or die(mysqli_connect_error());

2. Execute Query

syntax

```
mysqli_query(connection name, query);

mysqli_query(connection name, query) or die(mysqli_error(connection name));
```

e.g.

```
$result = mysqli_query($link,"select * from employee");
```

e.g.

```
$result = mysqli_query($link,"select * from employee") or die(mysqli_error($link));
```

3 Count Number of Rows

```
mysqli_num_rows();
```

e.g

```
$result = mysqli_query($link,"select * from employee");
$record = mysqli_num_rows($result);

if($record > 0)
{
 echo "Record Found";
}
else
{
 echo "No Record Found";
}
```

4 Fetch Records

```
$arr = mysqli_fetch_assoc($result)   $arr['index']
```

```
$arr = mysqli_fetch_array($result)   $arr['index'] or $arr[0]
$arr = mysqli_fetch_row($result)   $arr[0]
$arr = mysqli_fetch_object($result)   $arr->index
```

5 Close Connection

syntax

```
mysqli_close(connection name)
```

e.g.

```
mysqli_close($link);
```
