



Experiment No. 2

**Title: Arrays, Image and File handling functions, User
Defined function in PHP**



Batch:A3**Roll No.:16010421073****Experiment No.:2****Aim:** Write PHP script for demonstrating use of arrays, functions, image and file handling functions.**Resources needed:** Windows OS, Web Browser, Editor, and XAMPP Server**Pre Lab/ Prior Concepts:**

Students should have prior knowledge of HTML/CSS/Basic Programming.

Theory:**What is PHP Arrays**

Arrays are complex variables that allow us to store more than one value or a group of values under a single variable name.

Types of Arrays in PHP

There are three types of arrays that you can create. These are:

1. Indexed array — An array with a numeric key.
2. Associative array — An array where each key has its own specific value.
3. Multidimensional array — An array containing one or more arrays within itself.

1. Indexed Arrays Example

```
<?php
    echo "<br><b> --CREATING & DISPLAYING NUMERIC ARRAY--
    </b><br><br>";
    $languages = array("PHP", "JAVA", "PYTHON", "C++");
    $languages[0]="C"; //will overwrite "PHP" at index 0
    $languages[4]="PHP";//will add PHP at index 4
    var_dump($languages);
?>
```

var_dump() function dumps information about one or more variables. The information holds type and value of the variable(s).

2. Associative Array Example

```
<?php
    echo "<br><br><b> --CREATING & DISPLAYING ASSOCIATIVE ARRAY-
    - </b><br>";
    $subjectcode=array(111=>"C",222=>"JAVA",333=>"PYTHON",444=>"C++",5
    55=>"PHP");
    echo "<br><br><b> ACCESSING ASSOCIATIVE ARRAY USING KEY
    </b><br>";
    echo "SUBJECT with code 333 is :". $subjectcode[333];
    //echo $subjectcode[777]; //throw php error
```

```

echo "<br><br><b> DISPLAYING ASSOCIATIVE ARRAY USING KEY
THROUGH LOOPS</b><br>";
foreach ($subjectcode as $code => $subvalue){
    echo "SubjectCode $code ::$subvalue <br>";}
echo "<br><b> Displaying raw value of array</b><br>";
print_r($subjectcode);

?>

```

print_r() function prints the information about a variable in a more human-readable way.

3. Multidimensional Array

```

<?php
echo "<br><b> --CREATING & DISPLAYING MULTIDIMENSIONAL ARRAY --
</b><br>";
$courses = array(
    array("code" => "001", "subject" => "PIC", "sem" => "First"),
    array( "code" => "003", "subject" => "PYTHON", "sem" => "Third"),
    array( "code" => "004", "subject" => "Advance JAVA/PYTHON", "sem" => "Fourth")
);
echo "<b> Display multidimensional array</b><br>";
$keys = array_keys($courses);
for($i = 0; $i < count($courses); $i++) {
    echo $keys[$i] . "<br>";
    foreach($courses[$keys[$i]] as $key => $value) {
        echo $key . " : " . $value . "<br>";
    }
}

?>

```

array_keys() function returns an array containing the keys.

Functions in PHP

1. PHP Built-in Functions

-A function is a self-contained block of code that performs a specific task.
 -PHP has a huge collection of internal or built-in functions that you can call directly within your PHP scripts to perform a specific task, like `gettype()`, `print_r()`, `var_dump()`, etc.

2. PHP User-Defined Functions

a. Syntax for creating Functions

```

function functionName(){
    // Code to be executed
}

```

b. function myFunc(\$oneParameter, \$anotherParameter){

```

    // Code to be executed
}

```

c. function myFunc(\$oneParameter, \$anotherParameter=value){

```

    // Code to be executed
}

```

```
//calling such functions
$myFunc("value1", "value2"); $myFunc("value1");// both will work
```

```
d. function myFunc($oneParameter, $anotherParameter){
    return $returnValue;
}
```

Built in Functions:

Built in functions are functions that comes along with PHP installation package. The built in functions are what make PHP a very efficient and productive scripting language. The built in functions can be classified into many categories.

- 1) String Functions
- 2) Date and Time Functions
- 3) Math Functions
- 4) Image Handling Functions
- 5) File Handling Functions

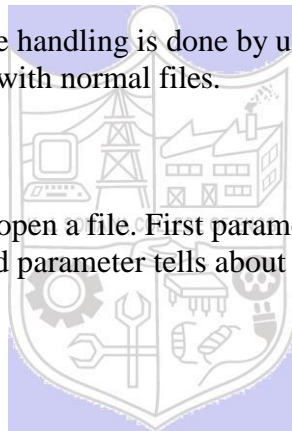
File Handling Functions:

File handling in PHP is similar as file handling is done by using any programming language like C. PHP has many functions to work with normal files.

Those functions are:

- 1) fopen() – This function is used to open a file. First parameter of fopen() contains name of the file which is to be opened and second parameter tells about mode in which file needs to be opened, e.g.,

```
<?php
$file = fopen("demo.txt", 'w');
?>
```



Files can be opened in any of the following modes :

- “w” – Opens a file for write only. If file not exist then new file is created and if file already exists then contents of file is erased.
- “r” – File is opened for read only.
- “a” – File is opened for write only. File pointer points to end of file. Existing data in file is preserved.
- “w+” – Opens file for read and write. If file not exist then new file is created and if file already exists then contents of file is erased.
- “r+” – File is opened for read/write.
- “a+” – File is opened for write/read. File pointer points to end of file. Existing data in file is preserved. If file is not there then new file is created.
- “x” – New file is created for write only.

- 2) fread() — After file is opened using fopen() the contents of data are read using fread(). It takes

two arguments. One is file pointer and another is file size in bytes, e.g.,

```
<?php
$filename = "demo.txt";
$file = fopen( $filename, 'r' );
$size = filesize( $filename );
$filedata = fread( $file, $size );
?>
```

3) fwrite() – New file can be created or text can be appended to an existing file using fwrite() function. Arguments for fwrite() function are file pointer and text that is to be written to file.

```
<?php
$file = fopen("demo.txt", 'w');
$text = "Hello world\n";
fwrite($file, $text);
?>
```

4) fclose() – file is closed using fclose() function. Its argument is file which needs to be closed, e.g.,

```
<?php
$file = fopen("demo.txt", 'r');
fclose($file);
?>
```

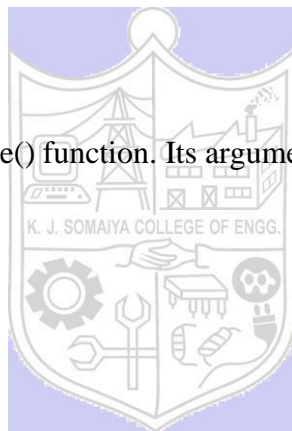


Image Handling Function:

imagecreate() returns an image identifier representing a blank image of specified size. In general, we recommend the use of imagecreatetruecolor() instead of imagecreate() so that image processing occurs on the highest quality image possible.

imagecreate (int \$width, int \$height)

```
<?php
header("Content-Type: image/png");
$im = @imagecreate(110, 20)
    or die("Cannot Initialize new GD image stream");
$background_color = imagecolorallocate($im, 0, 0, 0);
$text_color = imagecolorallocate($im, 233, 14, 91);
imagestring($im, 1, 5, 5, "A Simple Text String", $text_color);
imagepng($im);
imagedestroy($im);
```

?>

The above example will output something similar to:

A Simple Text String

Image Upload using File Upload

The process of uploading a file follows these steps –

The user opens the page containing a HTML form featuring a text files, a browse button and a submit button.

The user clicks the browse button and selects a file to upload from the local PC.

The full path to the selected file appears in the text filed then the user clicks the submit button.

The selected file is sent to the temporary directory on the server.

The PHP script that was specified as the form handler in the form's action attribute checks that the file has arrived and then copies the file into an intended directory.

The PHP script confirms the success to the user.

An uploaded file could be a text file or image file or any document.

```
<?php
if(isset($_FILES['image'])){
    $errors= array();
    $file_name = $_FILES['image']['name'];
    $file_size =$_FILES['image']['size'];
    $file_tmp =$_FILES['image']['tmp_name'];
    $file_type=$_FILES['image']['type'];
    $file_ext=strtolower(end(explode('.',$_FILES['image']['name'])));

    $extensions= array("jpeg","jpg","png");

    if(in_array($file_ext,$extensions)=== false){
        $errors[]="extension not allowed, please choose a JPEG or PNG file.";
    }

    if($file_size > 2097152){
        $errors[]='File size must be excately 2 MB';
    }

    if(empty($errors)==true){
        move_uploaded_file($file_tmp,"images/".$file_name);
        echo "Success";
    }else{
        print_r($errors);
    }
}
```

```

    }
?>
<html>
  <body>

    <form action="" method="POST" enctype="multipart/form-data">
      <input type="file" name="image" />
      <input type="submit"/>
    </form>

  </body>
</html>

```

PHP GD

GD is an open-source code library that is required to create and manipulate images in PHP. It is used for creating PNG, JPEG, and GIF images. It is commonly used to create charts, graphics, thumbnails, etc, and website development is the most common application of GD



Installing PHP GD in Windows

To install the PHP GD follow the following steps:

Step 1: Install [XAMPP](#) in your windows system.

Step 2: Verify if GD is already installed or not. So to verify GD we need to follow the following steps:

- Open XAMPP and click on the start button in front of Apache and MySQL to start php server, and go to the admin.
- A web page will open. Go to the 'PHPInfo' option on the top of the page.
- A PHPInfo dashboard will open up.
- Scroll down the page and search for 'gd'. If present, GD is already installed
 - If 'gd' is not present on the phpinfo page, you can follow the next to install GD.

Step 3: Locate and open php.ini in your editor.

Step 4: Find ;extension=gd.

Step 5: Remove semicolon from ;extension=gd and save the file.

Step 6: Go to php folder. It is usually present in C:\xampp.

Step 7: Look for php_gd.dll in the ext folder.

Step 8: Copy php_gd.dll and paste it into the following folder.

C:\Windows\System32

Step 9: Restart the XAMPP server. Now use the phpinfo() method as mentioned in step 2 to check whether the GD library is installed in the PHP server. It will show information about the PHP's configuration.

FTP support	enabled
FTPS support	enabled
gd	
GD Support	enabled
GD Version	bundled (2.1.0 compatible)
FreeType Support	enabled
FreeType Linkage	with freetype
FreeType Version	2.9.1
GIF Read Support	enabled
GIF Create Support	enabled
JPEG Support	enabled
libJPEG Version	8
PNG Support	enabled
libPNG Version	1.6.34
WMF Support	enabled
XPM Support	enabled
libXpm Version	30512
XBM Support	enabled
WebP Support	enabled
BMP Support	enabled
AVIF Support	enabled
TGA Read Support	enabled

Directive	Local Value	Master Value
gd.jpeg_ignore_warning	1	1

gettext

GetText Support	enabled
-----------------	---------

hash

hash support	enabled
Hashing Engines	md2 md4 md5 sha1 sha224 sha256 sha384 sha512/224 sha512/256 sha512 sha3-224 sha3-256 sha3-384 sha3-512 ripemd128 ripemd160 ripemd256 ripemd320 whirlpool tiger128,3 tiger160,3 tiger192,3 tiger128,4 tiger160,4 tiger192,4 snortu snortu256 gost gost-crypto adia32 crc12b crc128b hv113 hv113a hv113b hv113c hv113d hv113e hv113f hv113g hv113h hv113i hv113j hv113k hv113l hv113m hv113n hv113o hv113p hv113q hv113r hv113s hv113t hv113u hv113v hv113w hv113x hv113y hv113z hv113aa hv113ab hv113ac hv113ad hv113ae hv113af hv113ag hv113ah hv113ai hv113aj hv113ak hv113al hv113am hv113an hv113ao hv113ap hv113aq hv113ar hv113as hv113at hv113au hv113av hv113aw hv113ax hv113ay hv113az hv113ba hv113bb hv113bc hv113bd hv113be hv113bf hv113bg hv113bh hv113bi hv113bj hv113bk hv113bl hv113bm hv113bn hv113bo hv113bp hv113bq hv113br hv113bs hv113bt hv113bu hv113bv hv113bw hv113bx hv113by hv113bz hv113ca hv113cb hv113cc hv113cd hv113ce hv113cf hv113cg hv113ch hv113ci hv113cj hv113ck hv113cl hv113cm hv113cn hv113co hv113cp hv113cq hv113cr hv113cs hv113ct hv113cu hv113cv hv113cw hv113cx hv113cy hv113cz hv113da hv113db hv113dc hv113dd hv113de hv113df hv113dg hv113dh hv113di hv113dj hv113dk hv113dl hv113dm hv113dn hv113do hv113dp hv113dq hv113dr hv113ds hv113dt hv113du hv113dv hv113dw hv113dx hv113dy hv113dz hv113ea hv113eb hv113ec hv113ed hv113ee hv113ef hv113eg hv113eh hv113ei hv113ej hv113ek hv113el hv113em hv113en hv113eo hv113ep hv113eq hv113er hv113es hv113et hv113eu hv113ev hv113ew hv113ex hv113ey hv113ez hv113fa hv113fb hv113fc hv113fd hv113fe hv113ff hv113fg hv113fh hv113fi hv113fj hv113fk hv113fl hv113fm hv113fn hv113fo hv113fp hv113fq hv113fr hv113fs hv113ft hv113fu hv113fv hv113fw hv113fx hv113fy hv113fz hv113ga hv113gb hv113gc hv113gd hv113ge hv113gf hv113gg hv113gh hv113gi hv113gj hv113gk hv113gl hv113gm hv113gn hv113go hv113gp hv113gq hv113gr hv113gs hv113gt hv113gu hv113gv hv113gw hv113gx hv113gy hv113gz hv113ha hv113hb hv113hc hv113hd hv113he hv113hf hv113hg hv113hh hv113hi hv113hj hv113hk hv113hl hv113hm hv113hn hv113ho hv113hp hv113hq hv113hr hv113hs hv113ht hv113hu hv113hv hv113hw hv113hx hv113hy hv113hz hv113ia hv113ib hv113ic hv113id hv113ie hv113if hv113ig hv113ih hv113ii hv113ij hv113ik hv113il hv113im hv113in hv113io hv113ip hv113iq hv113ir hv113is hv113it hv113iu hv113iv hv113iw hv113ix hv113iy hv113iz hv113ja hv113jb hv113jc hv113jd hv113je hv113jf hv113jg hv113jh hv113ji hv113jj hv113jk hv113jl hv113jm hv113jn hv113jo hv113jp hv113jq hv113jr hv113js hv113jt hv113ju hv113jv hv113jw hv113jx hv113jy hv113jz hv113ka hv113kb hv113kc hv113kd hv113ke hv113kf hv113kg hv113kh hv113ki hv113kj hv113kk hv113kl hv113km hv113kn hv113ko hv113kp hv113kq hv113kr hv113ks hv113kt hv113ku hv113kv hv113kw hv113kx hv113ky hv113kz hv113la hv113lb hv113lc hv113ld hv113le hv113lf hv113lg hv113lh hv113li hv113lj hv113lk hv113ll hv113lm hv113ln hv113lo hv113lp hv113lq hv113lr hv113ls hv113lt hv113lu hv113lv hv113lw hv113lx hv113ly hv113lz hv113ma hv113mb hv113mc hv113md hv113me hv113mf hv113mg hv113mh hv113mi hv113mj hv113mk hv113ml hv113mn hv113mo hv113mp hv113mq hv113mr hv113ms hv113mt hv113mu hv113mv hv113mw hv113mx hv113my hv113mz hv113na hv113nb hv113nc hv113nd hv113ne hv113nf hv113ng hv113nh hv113ni hv113nj hv113nk hv113nl hv113nm hv113nn hv113no hv113np hv113nq hv113nr hv113ns hv113nt hv113nu hv113nv hv113nw hv113nx hv113ny hv113nz hv113oa hv113ob hv113oc hv113od hv113oe hv113of hv113og hv113oh hv113oi hv113oj hv113ok hv113ol hv113om hv113on hv113oo hv113op hv113oq hv113or hv113os hv113ot hv113ou hv113ov hv113ow hv113ox hv113oy hv113oz hv113pa hv113pb hv113pc hv113pd hv113pe hv113pf hv113pg hv113ph hv113pi hv113pj hv113pk hv113pl hv113pm hv113pn hv113po hv113pp hv113pq hv113pr hv113ps hv113pt hv113pu hv113pv hv113pw hv113px hv113py hv113pz hv113qa hv113qb hv113qc hv113qd hv113qe hv113qf hv113qg hv113qh hv113qi hv113qj hv113qk hv113ql hv113qm hv113qn hv113qo hv113qp hv113qq hv113qr hv113qs hv113qt hv113qu hv113qv hv113qw hv113qx hv113qy hv113qz hv113ra hv113rb hv113rc hv113rd hv113re hv113rf hv113rg hv113rh hv113ri hv113rj hv113rk hv113rl hv113rm hv113rn hv113ro hv113rp hv113rq hv113rr hv113rs hv113rt hv113ru hv113rv hv113rw hv113rx hv113ry hv113rz hv113sa hv113sb hv113sc hv113sd hv113se hv113sf hv113sg hv113sh hv113si hv113sj hv113sk hv113sl hv113sm hv113sn hv113so hv113sp hv113sq hv113sr hv113ss hv113st hv113su hv113sv hv113sw hv113sx hv113sy hv113sz hv113ta hv113tb hv113tc hv113td hv113te hv113tf hv113tg hv113th hv113ti hv113tj hv113tk hv113tl hv113tm hv113tn hv113to hv113tp hv113tq hv113tr hv113ts hv113tt hv113tu hv113tv hv113tw hv113tx hv113ty hv113tz hv113ua hv113ub hv113uc hv113ud hv113ue hv113uf hv113ug hv113uh hv113ui hv113uj hv113uk hv113ul hv113um hv113un hv113uo hv113up hv113uq hv113ur hv113us hv113ut hv113uu hv113uv hv113uw hv113ux hv113uy hv113uz hv113va hv113vb hv113vc hv113vd hv113ve hv113vf hv113vg hv113vh hv113vi hv113vj hv113vk hv113vl hv113vm hv113vn hv113vo hv113vp hv113vq hv113vr hv113vs hv113vt hv113vu hv113vv hv113vw hv113vx hv113vy hv113vz hv113wa hv113wb hv113wc hv113wd hv113we hv113wf hv113wg hv113wh hv113wi hv113wj hv113wk hv113wl hv113wm hv113wn hv113wo hv113wp hv113wq hv113wr hv113ws hv113wt hv113wu hv113wv hv113ww hv113wx hv113wy hv113wz hv113xa hv113xb hv113xc hv113xd hv113xe hv113xf hv113xg hv113xh hv113xi hv113xj hv113xk hv113xl hv113xm hv113xn hv113xo hv113xp hv113xq hv113xr hv113xs hv113xt hv113xu hv113xv hv113xw hv113xx hv113xy hv113xz hv113ya hv113yb hv113yc hv113yd hv113ye hv113yf hv113yg hv113yh hv113yi hv113yj hv113yk hv113yl hv113ym hv113yn hv113yo

Activity: Write PHP program for implementing different types of arrays and their associated functions. Also write program for implementing image and file handling functions.

Output(Code with result Snapshot)

Arrays and its associated functions

Code:

```
<!DOCTYPE html>
<html>

<head>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    .container {
      width: 70%;
      margin: 0 auto;
    }

    h1 {
      text-align: center;
    }

    table {
      width: 100%;
```



```

        border-collapse: collapse;
    }

    th, td {
        padding: 20px;
        text-align: left;
        border-bottom: 1px solid #ddd;
        border: 2px solid black;
        background-color: #bfe3f5 ;
    }

    th {
        background-color: #6860bd;
    }
</style>
</head>

<body>
    <div class="container">
        <h1>Array Examples</h1>

        <?php
            // Indexed Array
            $fruits = array("Apple", "Banana", "Cherry", "strawberry");
            $moreFruits = array("Grapes", "Kiwi", "Lemon");

            // Associative Array
            $person = array("first_name" => "Keyur", "last_name" => "Patel", "age" => 20);

            // Multidimensional Array
            $employees = array(
                array("name" => "Keyur", "position" => "Manager"),
                array("name" => "Mukesh Ambani", "position" => "Businessman"),
                array("name" => "Adani", "position" => "HR")
            );

            echo "<h2>Indexed Array</h2>";
            echo "<h3>Fruits 1</h3>";
            echo "<ul>";
            foreach ($fruits as $fruit) {
                echo "<li>$fruit</li>";
            }
            echo "</ul>";

            echo "<h3>Fruits 2</h3>";
            echo "<ul>";

```

```

    foreach ($moreFruits as $fruit) {
        echo "<li>$fruit</li>";
    }
    echo "</ul>";

    echo "<h2>Associative Array</h2>";
    echo "<p>Name: " . $person["first_name"] . " " . $person["last_name"] . "</p>";
    echo "<p>Age: " . $person["age"] . "</p>";

    echo "<h2>Multidimensional Array</h2>";
    echo "<table>";
    echo "<tr><th>Name</th><th>Position</th></tr>";
    foreach ($employees as $employee) {
        echo "<tr>";
        echo "<td>" . $employee["name"] . "</td>";
        echo "<td>" . $employee["position"] . "</td>";
        echo "</tr>";
    }
    echo "</table>";

    // Using Array Functions
    echo "<h2>Array Functions</h2>";

    // Count the number of elements in the indexed array
    echo "Number of fruits: " . count($fruits) . "<br>";

    // Check if an element exists in the indexed array
    $searchFruit = "Strawberry";
    echo "Is '$searchFruit' in the fruits array? " . (in_array($searchFruit, $fruits)
? "Yes" : "No") . "<br>";

    // Combine two arrays (indexed)
    $combinedFruits = array_merge($fruits, $moreFruits);
    echo "Combined Fruits: " . implode(", ", $combinedFruits) . "<br>";

    // Sorting Indexed Array
    sort($fruits);
    echo "<h2>Sorted Fruits</h2>";
    echo "<ul>";
    foreach ($fruits as $fruit) {
        echo "<li>$fruit</li>";
    }
    echo "</ul>";

    // Sorting Multidimensional Array
    function sortEmployeesByName($a, $b) {
        return $a["name"] < $b["name"] ? -1 : 1;
    }

```

```

    }

    usort($employees, "sortEmployeesByName");
    echo "<h2>Sorted Employees by Name</h2>";
    echo "<table>";
    echo "<tr><th>Name</th><th>Position</th></tr>";
    foreach ($employees as $employee) {
        echo "<tr>";
        echo "<td>" . $employee["name"] . "</td>";
        echo "<td>" . $employee["position"] . "</td>";
        echo "</tr>";
    }
    echo "</table>";

    $slicedFruits = array_slice($fruits, 1, 2); // Get elements from index 1 to 2
(exclusive)
    echo "<h2>Sliced Fruits</h2>";
    echo "<ul>";
    foreach ($slicedFruits as $fruit) {
        echo "<li>$fruit</li>";
    }
    echo "</ul>";

    // Array Reverse
    $reversedFruits = array_reverse($fruits);
    echo "<h2>Reversed Fruits 1</h2>";
    echo "<ul>";
    foreach ($reversedFruits as $fruit) {
        echo "<li>$fruit</li>";
    }
    echo "</ul>";

    $reversedmoreFruits = array_reverse($moreFruits);
    echo "<h2>Reversed Fruits 2</h2>";
    echo "<ul>";
    foreach ($reversedmoreFruits as $moreFruit) {
        echo "<li>$moreFruit</li>";
    }
    echo "</ul>";
    ?>
</div>
</body>

</html>

```

Output:

Array Examples

Indexed Array

Fruits 1

- Apple
- Banana
- Cherry
- strawberry

Fruits 2

- Grapes
- Kiwi
- Lemon

Associative Array

Name: Keyur Patel

Age: 20

Multidimensional Array

Name	Position
Keyur	Manager
Mukesh Ambani	Businessman
Adani	HR

Array Functions

Number of fruits: 4

Is 'Strawberry' in the fruits array? No

Combined Fruits: Apple, Banana, Cherry, strawberry, Grapes, Kiwi, Lemon

Sorted Fruits

Sorted Fruits

- Apple
- Banana
- Cherry
- strawberry

Sorted Employees by Name

Name	Position
Adani	HR
Keyur	Manager
Mukesh Ambani	Businessman

Sliced Fruits

- Banana
- Cherry
- Reversed Fruits 1
 - strawberry
 - Cherry
 - Banana
 - Apple
- Reversed Fruits 2
 - Lemon
 - Kiwi
 - Grapes

File Handling

file_choose.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>File Upload</title>
  <style>
    body {
      font-family: Arial, sans-serif;
```

```

        background-color: #f4f4f4;
        text-align: center;
    }

    .container {
        background-color: #fff;
        padding: 80px;
        border-radius: 5px;
        box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
        width: 300px;
        margin: 0 auto;
    }

    .upload-form {
        display: flex;
        flex-direction: column;
    }

    .upload-input {
        margin-bottom: 30px;
    }

    .upload-button {
        background-color: #4CAF50;
        color: white;
        border: none;
        padding: 10px;
        border-radius: 5px;
        cursor: pointer;
    }

    .upload-button:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>File Upload</h2>
        <form class="upload-form" method="POST" action="upload.php"
enctype="multipart/form-data">
            <input class="upload-input" type="file" name="fileToUpload"
id="fileToUpload">
            <input class="upload-button" type="submit" value="Upload File">
        </form>
    </div>
</body>

```

</html>

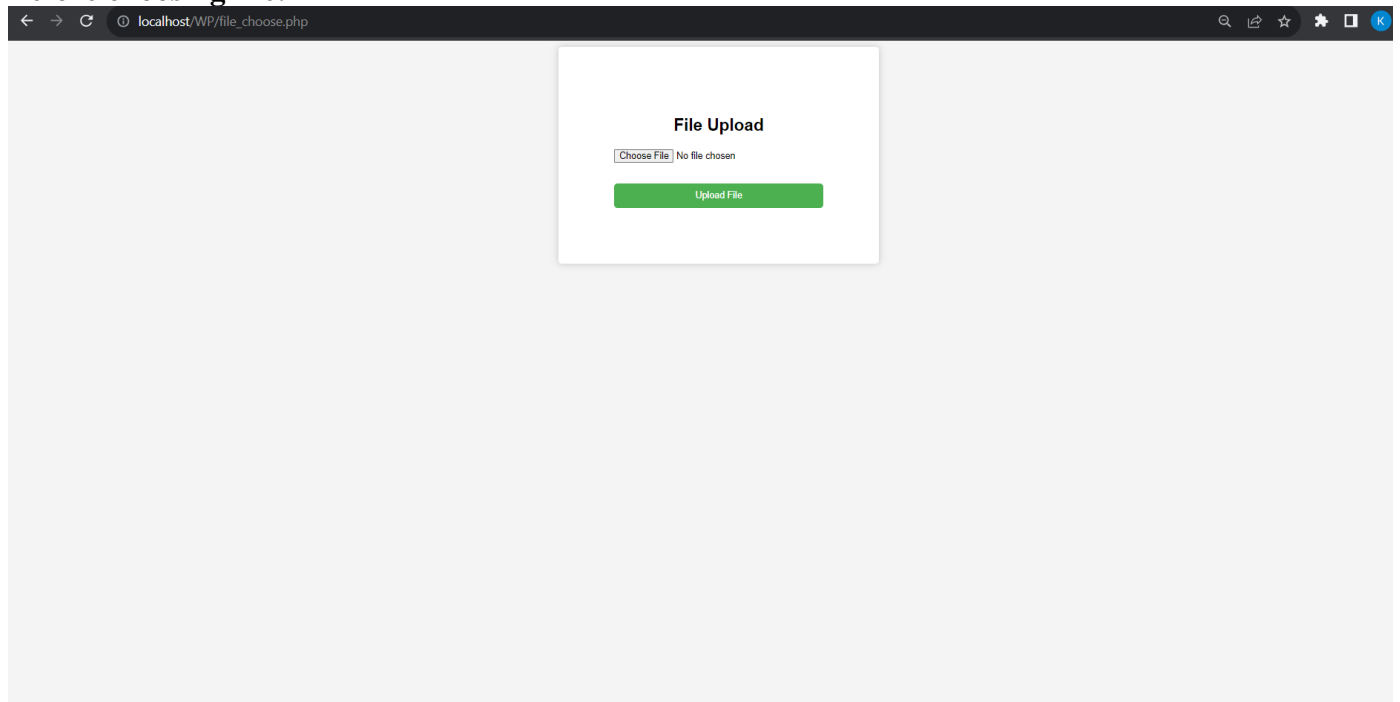
upload.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_FILES["fileToUpload"]))
{
    $targetDirectory = "uploads/";
    $targetFile = $targetDirectory . basename($_FILES["fileToUpload"]["name"]);

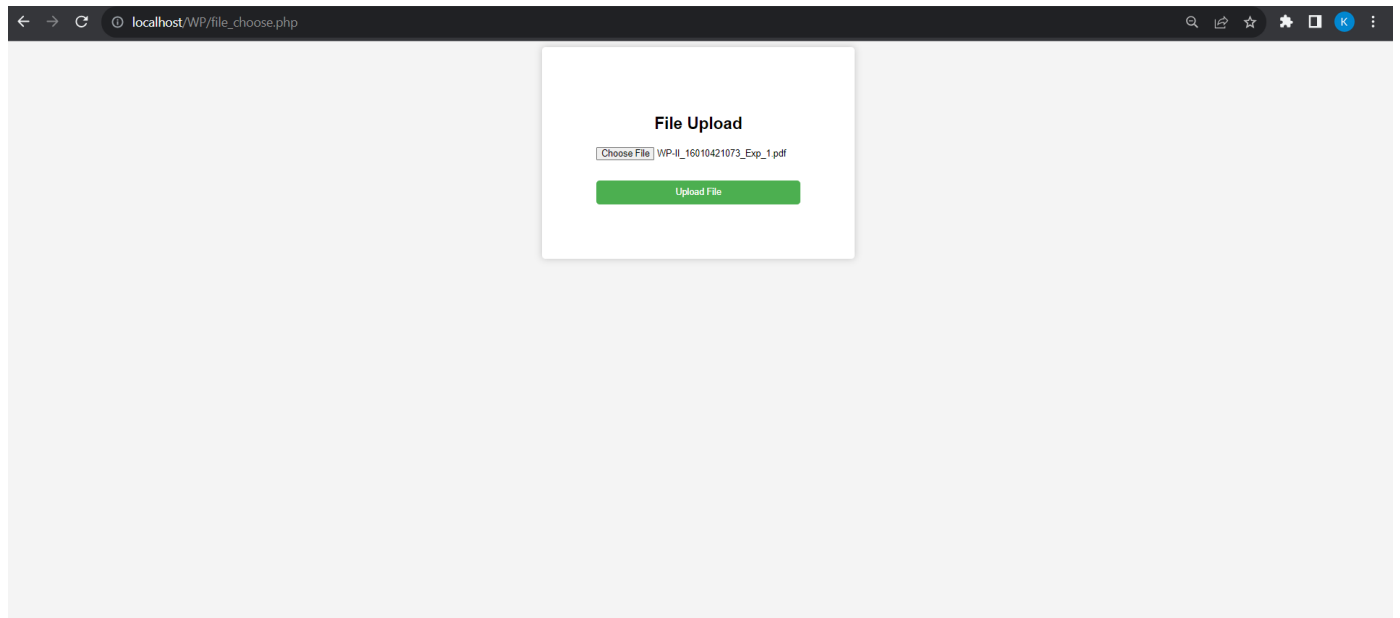
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile))
    {
        echo "The file " . basename($_FILES["fileToUpload"]["name"]) . " has been
uploaded.";
    }
    else
    {
        echo "Sorry, there was an error uploading your file.";
    }
}
?>
```

Output:

Before choosing file:



After choosing file:



File uploaded:



Image Handling:

image_choose.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Image Upload and Display</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f2f2f2;
      margin: 0;
      padding: 0;
      display: flex;
```

```

        justify-content: center;
        align-items: center;
        height: 100vh;
    }

    .container {
        background-color: #fff;
        border-radius: 5px;
        box-shadow: 0 2px 10px rgba(0, 0, 0, 0.2);
        padding: 20px;
        width: 400px;
        text-align: center;
    }

    h1 {
        margin: 0 0 20px;
    }

    form {
        margin: 0;
    }

    input[type="file"] {
        margin: 0 0 10px;
    }

    input[type="submit"] {
        display: block;
        margin: 10px auto 0;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Upload and Display Image</h1>
        <form action="fileupload.php" method="post" enctype="multipart/form-data">
            <input type="file" name="image" id="fileToUpload">
            <input type="submit" value="Upload Image" name="submit">
        </form>
    </div>
</body>
</html>

```

fileupload.php

```

<!DOCTYPE html>
<html>

```



```

<head>
    <title>Uploaded Image</title>
</head>
<body>
<h1>Uploaded Image</h1>

<?php
if (isset($_POST['submit'])) {
    $targetDir = "uploads/"; // Create an "uploads" directory to store uploaded images
    $targetFile = $targetDir . basename($_FILES["image"]["name"]);
    $uploadOk = 1;
    $imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));

    // Check if the file is an actual image
    $check = getimagesize($_FILES["image"]["tmp_name"]);
    if ($check !== false) {
        echo "File is an image - " . $check["mime"] . ".<br>";
        $uploadOk = 1;
    } else {
        echo "File is not an image.<br>";
        $uploadOk = 0;
    }

    // Check file size (you can change the size limit)
    if ($_FILES["image"]["size"] > 500000) {
        echo "Sorry, your file is too large.<br>";
        $uploadOk = 0;
    }

    // Allow certain image file formats (you can add more)
    if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" &&
$imageFileType != "gif") {
        echo "Sorry, only JPG, JPEG, PNG, and GIF files are allowed.<br>";
        $uploadOk = 0;
    }

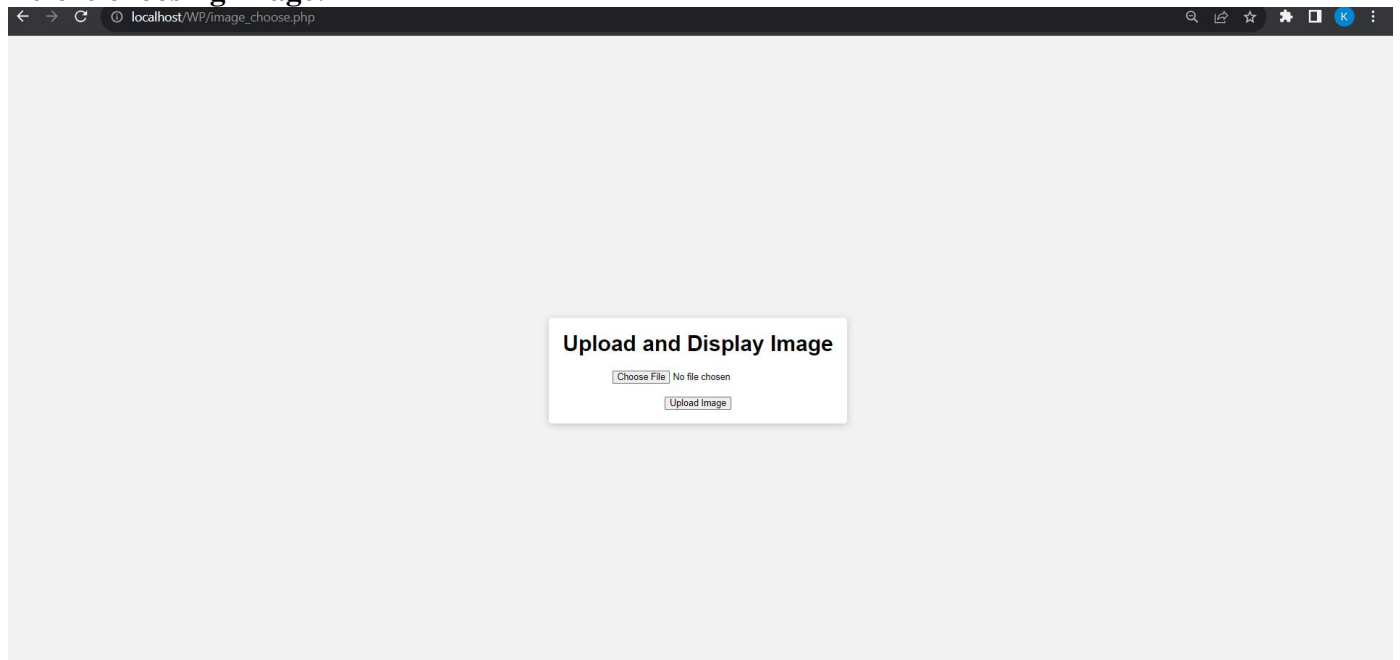
    if ($uploadOk == 0) {
        echo "Sorry, your file was not uploaded.<br>";
    } else {
        if (move_uploaded_file($_FILES["image"]["tmp_name"], $targetFile)) {
            echo "The image " . htmlspecialchars(basename($_FILES["image"]["name"])) . "
has been uploaded and displayed below:<br>";
            echo "<img src='" . $targetFile . "' width='300'>";
        } else {
            echo "Sorry, there was an error uploading your file.<br>";
        }
    }
}

```

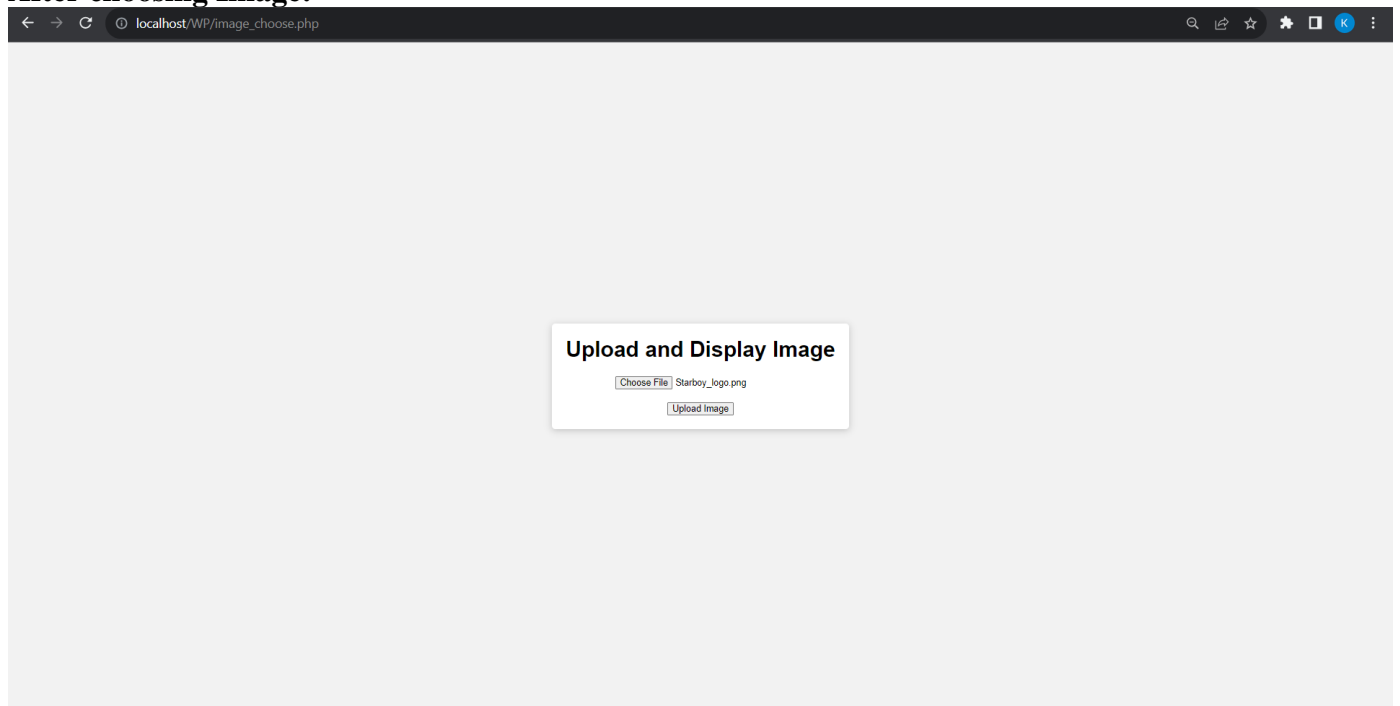
```
}  
?>  
  
</body>  
</html>
```

Output:

Before choosing image:



After choosing Image:



← → ↻ ⓘ localhost/WP/fileupload.php

Uploaded Image

File is an image - image/png.

The image Starboy_logo.png has been uploaded and displayed below:



Questions:-

1. What is difference between array_merge and array_combine? Answer:

Aspect	`array_merge`	`array_combine`
Purpose	Merges two or more arrays into one.	Creates an array by using one array for keys and another for values.
Parameters	Accepts multiple arrays as input.	Requires two arrays as input, one for keys and one for values.
Result	Combines all values into a single array.	Creates a new array with keys and values paired together.
Key Handling	If keys are not specified, they are reindexed numerically.	Uses one array for keys and another for values; keys must be unique.
Example	<code>`php \$merged = array_merge(\$array1, \$array2);`</code>	<code>`php \$combined = array_combine(\$keys, \$values);`</code>
Use Case	Combining multiple arrays into one.	Creating associative arrays from separate arrays.

2. Write a program based on functions passing reference as parameter.

Answer: In this program:

1. We define a function `incrementByReference` that takes two parameters: a reference to a number and a value by which the number should be incremented.
2. We declare the `$originalNumber` variable and set it to 10, and we specify the `$incrementValue` as 5.
3. We call the `incrementByReference` function, passing `$originalNumber` by reference. This means any changes made to `$number` inside the function will affect the original variable.
4. Inside the function, we increment the value of `$number` by the specified increment value.
5. After the function call, we print the original number to show that it has been modified by the function.

When we run this program, we'll see that the original number is incremented by the specified value, demonstrating the use of passing variables by reference to modify them inside a function.

```
<?php
function incrementByReference(&$number, $value) {
    $number += $value;
}

$originalNumber = 10;
$incrementValue = 5;

echo "Original Number: $originalNumber<br>";
```

```
incrementByReference($originalNumber, $incrementValue);

echo "Number after increment: $originalNumber";
?>
```

3. How can you display a file download dialog box using PHP?

Answer: To display a file download dialog box using PHP, we can use the header function to set the necessary HTTP headers. Here's a simple example of how to do this:

```
<?php
// Define the file path
$file = 'path_to_your_file.pdf'; // Replace with the path to your file

// Check if the file exists
if (file_exists($file)) {
    // Set the appropriate headers for file download
    header('Content-Description: File Transfer');
    header('Content-Type: application/octet-stream');
    header('Content-Disposition: attachment; filename=' . basename($file));
    header('Expires: 0');
    header('Cache-Control: must-revalidate');
    header('Pragma: public');
    header('Content-Length: ' . filesize($file));

    // Output the file
    readfile($file);
    exit;
} else {
    // File not found
    echo 'File not found.';
}
?>
```

In this PHP code:

1. Replace 'path_to_your_file.pdf' with the actual path to the file you want to make available for download.
2. The code first checks if the file exists using file_exists.
3. If the file exists, it sends the necessary HTTP headers to prompt the user's browser to display the file download dialog box.
4. Headers like Content-Disposition with the attachment value specify that the file should be treated as an attachment to download. The basename(\$file) provides the filename that will be displayed in the download dialog.
5. It sets other cache control headers to ensure the file is not cached.
6. The readfile function is used to output the file to the browser.

7. The exit function is called to prevent any further execution of the script. If the file does not exist, it will display "File not found." When a user accesses the PHP script, the browser will display a download dialog box for the specified file, allowing the user to download it to their local machine.

4. What is the purpose of php.ini file?

Answer: The php.ini file in PHP serves as the configuration file that controls various settings and options for the PHP interpreter. It plays a crucial role in defining how PHP behaves on your web server. Here are the primary purposes of the php.ini file:

- 1. Configuration Settings:** The php.ini file contains a wide range of configuration settings for PHP. These settings determine how PHP interacts with the server, handles error reporting, processes data, manages resources, and more. It allows you to customize PHP to meet the specific requirements of your applications.
- 2. Error Handling:** You can specify how PHP should handle errors and log error messages. This includes options for displaying errors on the screen, writing errors to log files, and configuring error reporting levels.
- 3. Security:** PHP configuration settings in php.ini can be used to enhance the security of your PHP applications. You can enable or disable features that may pose security risks, set permissions, and control access to certain functions.
- 4. Resource Management:** You can configure resource limits, such as memory limits, execution time limits, and file upload limits. This helps in preventing excessive resource consumption or long-running scripts that could impact server performance.
- 5. Extension Configuration:** PHP extensions (e.g., MySQL, GD, cURL) are often controlled through php.ini. You can enable or disable specific extensions and set their configuration options.
- 6. Output Buffering:** PHP offers output buffering, and you can configure its behavior in php.ini. This can be useful for improving performance and managing the content sent to the browser.

Outcomes: CO1: Illustrate use of basic PHP concepts to develop applications.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Understood how to create Indexed, Associative, Multi dimensional arrays in PHP and use the different array functions linked with them.

Learned how to use built-in PHP functions and how to define our own functions with or without parameters and returning value if needed.

Also learned to implement the file and image handling functions in php.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge

with dateReferences:

1. Instruction Link to install php GD –

<https://www.geeksforgeeks.org/how-to-install-php-gd-in-windows/>

Books:

1. Thomson PHP and MySQL Web Development Addison-Wesley Professional , 5th Edition 2016.
2. Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3. Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018

