# PHP TUTORIAL NOTES

# by: SIR MUHAMMAD FARHAN

Contact No: +92316-2859445

Email: mohammadfarhan44500@gmail.com

GitHub: https://github.com/muhammadfarhandevelper

# PHP TUTORIALS

## Include & Require:

**The include and require statements are identical, except upon failure:**

- require will produce a fatal error (E_COMPILE_ERROR) and stop the script
- include will only produce a warning (E_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of FrameWork, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution. This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

## Include (Example):

```
<?php include 'tutorial1.php'?>

<h1>Main Content</h1>
```

## Require (Example):

```
<?php require 'tutorial1.php'?>

<h1>Main Content</h1>
```

## Include_once & Require_once:

## Include_One:

The include_once keyword is used to embed PHP code from another file. If the file is not found, a warning is shown and the program continues to run. If the file was already included previously, this statement will not include it again.

## Require_Once:

The require_once keyword is used to embed PHP code from another file. If the file is not found, a fatal error is thrown and the program stops. If the file was already included previously, this statement will not include it again.

## Super Global Variables:

The superglobals are available throughout your script. These variables can be accessed from any function, class or any file without doing any special task such as declaring any global variable etc. They are mainly used to store and get information from one page to another etc in an application.

Below is the list of super global variables available in PHP:

1. $_GET
2. $_POST
3. $_REQUEST
4. $_SERVER
5. $_COOKIE
6. $_SESSION
7. $_FILES

### Example of $_SERVER:

```
<form action="<?php $_SERVER['PHP_SELF']?>" method="post">
 <div class="container mt-5">

     <div class="row">
         <h1 class="text-center">Insertion Data</h1>
         <div class="col-md-6 mt-5">
             <input type="text"  placeholder="Enter First Name" name="fname" required class="form-control"/>
         </div>
         <div class="col-md-6 mt-5">
             <input type="text" placeholder="Enter Last Name" name="lname" required class="form-control"/>
         </div>
         <div class="col-md-12 mt-5">
             <input type="email" placeholder="Enter Email Name"  name="email" required class="form-control"/>
         </div>
     </div>
             <div class="row mt-5">
                 <div class="offset-md-4 col-md-4">
                     <button type="submit" class="btn btn-primary w-100" name="send">Submit</button>
```

```
                    </div>
                </div>
        </div>

    </form>

<?php

    if(isset($_POST['send'])){
        echo $_POST['fname'] . "<br> ";
        echo $_POST['lname'];
    }

?>
```

## $_Cookies:

A **cookie** in PHP is a small file with a maximum size of 4KB that the web server stores on the **client computer**. They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time. A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

## Create_Cookies:

Setcookies(name,value,expire,path,domain,secure,httponly)

## View Cookie

$_COOKIE[name]

## Example:

```
    $cookie_name = "user";
    $cookie_value = "Sir Muhammad Farhan";

    setcookie($cookie_name,$cookie_value,time() + (20),"/");

    echo $_COOKIE[$cookie_name];
```

## DELETE COOKIE:

```
setcookie($cookie_name,$cookie_value,time() - (20),"/");
```

## $_SESSION:

In general, session refers to a frame of communication between two medium. A PHP session is used to store data on a server rather than the computer of the user. Session identifiers or SID is a unique number which is used to identify every user in a session based environment. The SID is used to link the user with his information on the server like posts, emails etc.

## $_SESSION Deletion:

Session_unset();                    //Remove all session variables or values.

Session_destroy();               // Destroy the session

## Example:

## Session Start:

```
<?php

   session_start();

   $_SESSION['name']  = "muhammadfarhan445@gmail.com";

   echo "session is set";


?>
```

## Show Session:

```
<?php
   session_start();

      print_r($_SESSION);
```

```
  ?>
```

## Deletion of Session:

```

  session_start();

  session_unset();
  session_destroy();

```

## $_FILE:

## Example:

## HTML:

```

<form action="" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="image" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>
```

## PHP CODE:

```
    if(isset($_FILES['image'])){
      print_r($_FILES);

      $file_name = $_FILES['image']['name'];
      $file_tmp = $_FILES['image']['tmp_name'];

      move_uploaded_file($file_tmp,"images/" . $file_name);
    }
```

## PHP DIE Function:

```
echo "this is line1";
echo "this is line2";
die();
echo "this is line3";
echo "this is line4";
```

### PHP EXIT Function:

```
echo "this is line1";
echo "this is line2";
exit();
echo "this is line3";
echo "this is line4";
```

# CRUD

## Create Read Update Delete

### PHP AND MYSQL:

There are 3 steps to perform PHP and MYSQL.

- Connection
- SQL Query
- Close connection

### Connection Method:

Mysqli_connect(Server Name,Username , password,Database Name);

### SQL Query Method:

Mysqli_query(Connection Name,SQL Query);

### Close Connection:

Mysqli_close(connection Name);

### First Example:

```

$connection_name =  mysqli_connect("localhost","root","","batch09") or die();
            echo "COnnection Successfully";
            $sql1 = "select * from student";
            $result_set  = mysqli_query($connection_name,$sql1);
          print_r($result_set);
```

## Read Data From Database:

```
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>ID</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
          <th>Edit</th>
          <th>Delete</th>
        </tr>
      </thead>
      <tbody>
        <?php

$connection_name =  mysqli_connect("localhost", "root", "", "batch09") or die();
        echo "COnnection Successfully";
        $sql1 = "select * from student";
        $result_set  = mysqli_query($connection_name, $sql1);
        print_r($result_set);
        if (mysqli_num_rows($result_set) > 0) {

          while ($row = mysqli_fetch_assoc($result_set)) { ?>
            <tr>
              <td><?php echo $row['id']; ?></td>
              <td><?php echo $row['st_fname'] ?></td>
              <td><?php echo $row['st_lname']; ?></td>
              <td><?php echo $row['email'] ?></td>
              <td>Edit</td>
              <td>Delete</td>
            </tr>
          <?php } ?>
      </tbody>
    </table>
  <?php } ?>
```

## Insert Data into Database:

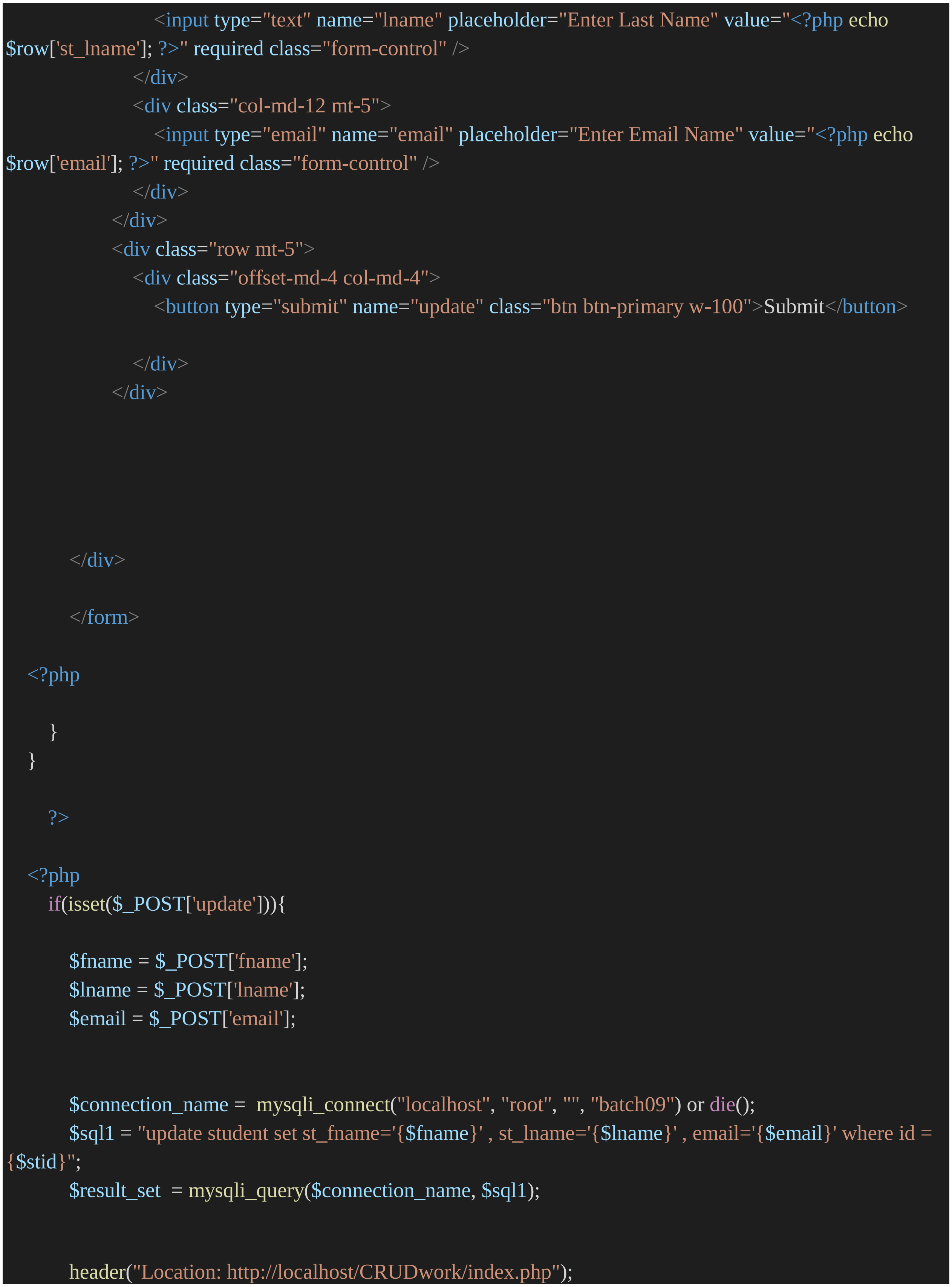
```
<?php
```

```
   if(isset($_POST['send'])){
       $firstname =  $_POST['fname'];
       $lastname = $_POST['lname'];
       $email = $_POST['email'];

       $connection = mysqli_connect("localhost","root","","batch09");
       $sql = "insert into student(st_fname,st_lname,email) values('{$firstname}','{$lastname}','{$email}')";
       $result = mysqli_query($connection,$sql);
       echo $result;
       mysqli_close($connection);
   }


?>
```

## Update Data into Database:

## Example:

```
 <?php
   $stid = $_GET['id'];

   $conn = mysqli_connect("localhost", "root", "", "batch09") or die();
   $sql = "select * from student where id={$stid}";
   $result = mysqli_query($conn, $sql);

   if (mysqli_num_rows($result) > 0) {

       while ($row = mysqli_fetch_assoc($result)) {



   ?>


           <div class="container mt-5">
               <form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
                   <div class="row">
                       <h1 class="text-center">Update Data</h1>
                       <div class="col-md-6 mt-5">
                           <input type="text" name="fname" value="<?php echo $row['st_fname']; ?>"
placeholder="Enter First Name" required class="form-control" />
                       </div>
                       <div class="col-md-6 mt-5">
```



```
              <input type="text" name="lname" placeholder="Enter Last Name" value="<?php echo
$row['st_lname']; ?>" required class="form-control" />
            </div>
            <div class="col-md-12 mt-5">
              <input type="email" name="email" placeholder="Enter Email Name" value="<?php echo
$row['email']; ?>" required class="form-control" />
            </div>
          </div>
          <div class="row mt-5">
            <div class="offset-md-4 col-md-4">
              <button type="submit" name="update" class="btn btn-primary w-100">Submit</button>

            </div>
          </div>






      </div>

      </form>

  <?php

    }
  }

    ?>

  <?php
    if(isset($_POST['update'])){

      $fname = $_POST['fname'];
      $lname = $_POST['lname'];
      $email = $_POST['email'];


      $connection_name =  mysqli_connect("localhost", "root", "", "batch09") or die();
      $sql1 = "update student set st_fname='{$fname}' , st_lname='{$lname}' , email='{$email}' where id =
{$stid}";
      $result_set  = mysqli_query($connection_name, $sql1);


      header("Location: http://localhost/CRUDwork/index.php");
    }
?>
```

## Delete:

```
  <?php
  $stid = $_GET['id'];

  ?>

      <div class="container mt-5">
        <form action="<?php $_SERVER['PHP_SELF'];?>" method="post">
          <div class="row">
            <h1 class="text-center">Delete Data</h1>
            <div class="offset-md-4 col-md-3 mt-5">
              <input type="text" name="fname" value="<?php echo $stid; ?>" placeholder="Enter First Name" required class="form-control" />
            </div>

          <div class="row mt-5">
            <div class="offset-md-5 col-md-2">
              <button type="submit" name="delete" class="btn btn-danger w-100">Delete</button>

            </div>
          </div>






      </div>

      </form>


  <?php

  if(isset($_POST['delete'])){


    $conn = mysqli_connect("localhost", "root", "", "batch09") or die();
    $sql = "delete from student where id={$stid}";
    $result = mysqli_query($conn, $sql);
```

```
        header("Location: http://localhost/CRUDwork/index.php");


    }
?>
```

## LOGIN / LOGOUT using Session

## Register and Login with security

Mysqli_real_escape_string($con,string)

And md5

## How to Get List from Database?

## On Change:

```
<form action="" method="post"  >


<select name="list" onchange="this.form.submit()">
<option value="">Select Course</option>

</select>

</form>
```

## How to upload Single Images:

## HTML:

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

<input type="file" name="myimg" multiple />
```

```
<button type="submit" name="upload">Upload</button>

</form>
```

## PHP:

```
<?php

$con = mysqli_connect("localhost","root","","batch09") or die("Not connect db");

if(isset($_POST['upload'])){

    $filename = $_FILES['myimg']['name'];
    $tmpname = $_FILES['myimg']['tmp_name'];
    $filesize = $_FILES['myimg']['size'];
    $filetype = $_FILES['myimg']['type'];

    if((strtolower($filetype) == "image/jpg" || strtolower($filetype) == "image/png" || strtolower($filetype)   ==
"image/jpeg")){

     if($filesize > 2097152){
       echo "file size is greater than 2 MB";
     }
     else{
       $sql = "insert into imagetbl(src) values('{$filename}')";
       $result  = mysqli_query($con,$sql) or die("Query Expired!");
       if($result){
         move_uploaded_file($tmpname,"images/".$filename);
         echo " <img src='images/$filename' width='200px'/>";
       }   }
  } else{
     echo "File type does not match!";
  }
}
?>
```

## How to Upload Multiple Images:

## HTML:

```

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

<input type="file" name="myimg[]" multiple />

```

```
<button type="submit" name="upload">Upload</button>

</form>
```

## PHP:

```
<?php

$con = mysqli_connect("localhost","root","","batch09") or die("Not connect db");

if(isset($_POST['upload'])){

  for($i =0;$i<count($_FILES['myimg']['name']);$i++){

    $filename = $_FILES['myimg']['name'][$i];
    $tmpname = $_FILES['myimg']['tmp_name'][$i];
    $filesize = $_FILES['myimg']['size'][$i];
    $filetype = $_FILES['myimg']['type'][$i];

    if((strtolower($filetype) == "image/jpg" || strtolower($filetype) == "image/png" || strtolower($filetype)   == "image/jpeg")){

      if($filesize > 2097152){
        echo "file size is greater than 2 MB";
      }
      else{

      $sql = "insert into imagetbl(src) values('{$filename}')";
      $result  = mysqli_query($con,$sql) or die("Query Expired!");
      if($result){

        move_uploaded_file($tmpname,"images/".$filename);
        echo " <img src='images/$filename' width='200px'/>";

      }
    }

  }
  else{

    echo "File type does not match!";

  }
}


}
?>
```

## PHP Filter Validation:

PHP filters are used to validate and sanitize external input.

The PHP filter extension has many of the functions needed for checking user input, and is designed to make data validation easier and quicker.

### Example:

filter_var(variable,filter_validtion)

filter_var(variable,filter_sanitize)

| Filter_Validation props | Definition |
|---|---|
| FILTER_VALIDATE_INT | check the int if it is int return true |
| FILTER_VALIDATE_EMAIL | check the email it is valid email return true |
| FILTER_VALIDATE_BOOL | check the Boolean it is valid return true |
| FILTER_VALIDATE_FLOAT | Check the validate it is float return true |
| FILTER_VALIDATE_IP | Check the validate it is IP Address return true |
| FILTER_VALIDATE_MAC | Check the validate it is MAC Address return true |
| FILTER_VALIDATE_URL | Check the validate it is URL Address return true |
| FILTER_VALIDATE_DOMAIN | Check the validate it is domain return true |

### Example:

```
    $e = 100;
    echo filter_var($e,FILTER_VALIDATE_INT,
    array(
    "options" =>
    array("min_range"=>1,"max_range"=>100)     )  );
```

## Sanitize Filters:

| Filter Validation props |
|---|
| FILTER_SANITIZE_NUMBER_INT |
| FILTER_ SANITIZE _EMAIL |
| FILTER_SANITIZE _NUMBER_FLOAT |
| FILTER_ SANITIZE _STRING |
| FILTER_ SANITIZE _URL |
| FILTER_SANITIZE_ADD_SLASHES |

## Filter_Var_array():

**Syntax:** filter_var_array(data array, filter array);

## Example:

```
    $a = "far//h an@gma il.com";
    $b = 2.2;
    $c = "ht tps://www. you tube.com/";
    $d = 23.56;

    echo filter_var($a,FILTER_SANITIZE_EMAIL) . "<br>";
    echo filter_var($b,FILTER_SANITIZE_NUMBER_INT) . "<br>";
    echo filter_var($c,FILTER_SANITIZE_URL) . "<br>";
    echo filter_var($d,FILTER_SANITIZE_NUMBER_FLOAT,FILTER_FLAG_ALLOW_FRACTION) .
"<br>";
```

## Filter_var_array():

## filter_var_array(values_array,validate_array);

## Example:

```

    $values = array(
        "stname" => "far//h an@gma il.com",
        "age"  => 22,
        "price" => 23.56
    );

    $filters = array(
      "stname" => FILTER_SANITIZE_EMAIL,
      "age" => array(
        "filter" => FILTER_VALIDATE_INT,"options" => array("min_range" => 1 , "max_range" => 100)
      ),
      "price" => FILTER_VALIDATE_FLOAT
    );

    echo "<br><br>";


    print_r(filter_var_array($values,$filters));
```

## How Email Send:

```
    $to = "mohammadfarhan44500@gmail.com";
    $name = "Ali Rehman";
    $from = "alirehman11@gmail.com";
    $subject  = "Check Email ";
    $message = "this is message";
    $header = "From : $from";

    $rs = mail($to,$subject,$message,$header);

    if($rs){
        echo "email sent";
    }
```

## What is an Exception?

An exception is an object that describes an error or unexpected behavior of a PHP script.

Exceptions are thrown by many PHP functions and classes.

User defined functions and classes can also throw exceptions.

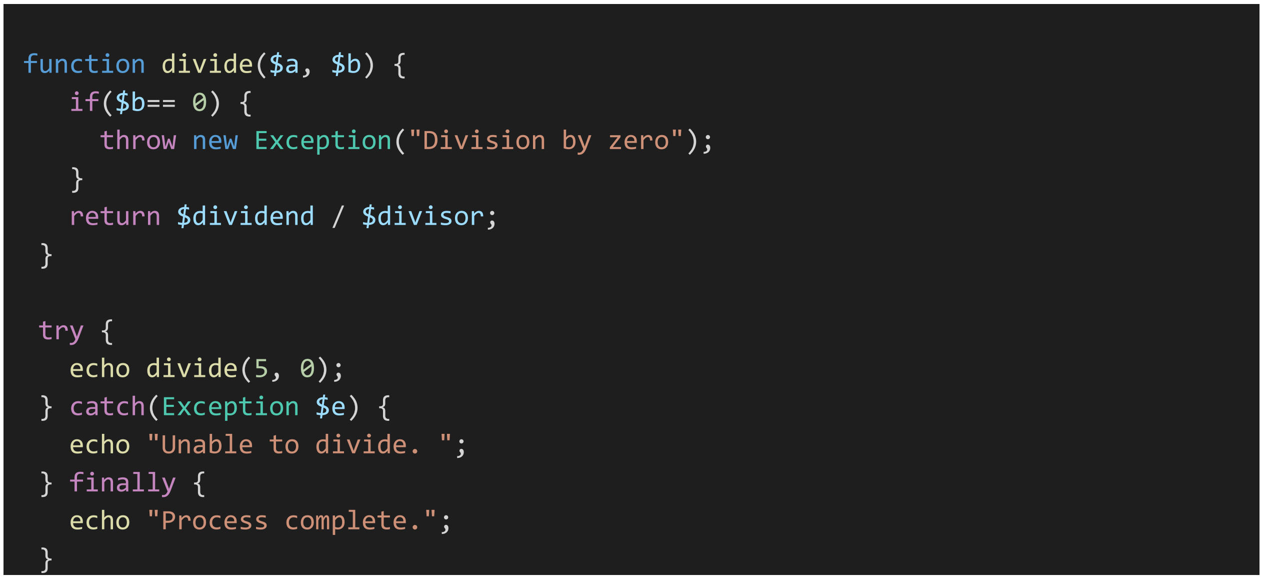
Exceptions are a good way to stop a function when it comes across data that it cannot use.

## Throwing an Exception:

The throw statement allows a user defined function or method to throw an exception. When an exception is thrown, the code following it will not be executed.

If an exception is not caught, a fatal error will occur with an "Uncaught Exception" message.

## Example:



```
function divide($a, $b) {
   if($b== 0) {
     throw new Exception("Division by zero");
   }
   return $dividend / $divisor;
 }

 try {
   echo divide(5, 0);
 } catch(Exception $e) {
   echo "Unable to divide. ";
 } finally {
   echo "Process complete.";
 }
```