**Php form validation:**

<?php

$action = $\_REQUEST['action'];

if(isset($\_POST['doReg'])) {

if(empty($\_POST['yourname'])) { $erryourname = "please enter your name"; }

if(empty($\_POST['emailaddress'])) { $erremailaddress = "please enter your email address"; }

if(empty($\_POST['mobilenumber'])) { $errmobilenumber = "please enter your mobile number"; }

if(empty($\_POST['pwd'])) { $errpwd = "please enter password"; }

if(empty($\_POST['gender'])) { $errgender = "please choose gender"; }

if(empty($\_POST['language'])) { $errlanguage = "please choose language"; }

if(empty($\_POST['country'])) { $errcountry = "please select country"; }

$yourname = $\_POST['yourname'];

$emailaddress = $\_POST['emailaddress'];

$mobilenumber = $\_POST['mobilenumber'];

$pwd = $\_POST['pwd'];

$gender = $\_POST['gender'];

$language = $\_POST['language'];

$country = $\_POST['country'];

$message = $\_POST['message'];

foreach($language as $key => $value) { $lang = $lang.$language[$key].", "; }

if(empty($erryourname) && empty($errpwd) && empty($errgender) && empty($errlanguage) && empty($errcountry) && empty($erremailaddress) && empty($errmobilenumber)) {

$sqlinsert = mysql\_query("insert into `usersdetails` (`fullname`, `emailaddress`, `mobilenumber`, `password`, `gender`, `language`, `country`, `message`) values ('$yourname', '$emailaddress', '$mobilenumber', '$pwd', '$gender', '$lang', '$country', '$message') ") or die("could not insert data : ". mysql\_error());

$sendmail = mail($emailaddress, 'Subject - Registered', 'Message - Registed Successfully');

if($sqlinsert) { echo '<META HTTP-EQUIV="Refresh" Content="0; URL=thankyou.php">'; } else { echo "Your Data not Inserted"; }

}

}

?>

<html>

<head>

<title>Untitled Document</title>

</head>

<body>

<?php

if($action=='newform') { ?>

<form name="reg" method="post" action="phpformvalidation.php">

<table border="1px">

<?php if(empty($erryourname)) { ?>

<tr><td>Your Name</td><td><input type="text" name="yourname" value="<?=$yourname?>" /></td></tr>

<?php } elseif(!empty($erryourname)) { ?>

<tr><td>Your Name</td><td><input type="text" name="yourname" value="" /><br /><span style="color:#FF0000"><?=$erryourname?></span></td></tr>

<?php } ?>

<?php if(empty($erremailaddress)) { ?>

<tr><td>Email Address</td><td><input type="text" name="emailaddress" value="<?=$emailaddress?>" /></td></tr>

<?php } elseif(!empty($erremailaddress)) { ?>

<tr><td>Email Address</td><td><input type="text" name="emailaddress" value="" /><br /><span style="color:#FF0000"><?=$erremailaddress?></span></td></tr>

<?php } ?>

<?php if(empty($errmobilenumber)) { ?>

<tr><td>Mobile Number</td><td><input type="text" name="mobilenumber" value="<?=$mobilenumber?>" /></td></tr>

<?php } elseif(!empty($errmobilenumber)) { ?>

<tr><td>Mobile Number</td><td><input type="text" name="mobilenumber" value="" /><br /><span style="color:#FF0000"><?=$errmobilenumber?></span></td></tr>

<?php } ?>

<?php if(empty($errpwd)) { ?>

<tr><td>Password</td><td><input type="password" name="pwd" value="<?=$pwd?>" /></td></tr>

<?php } elseif(!empty($errpwd)) { ?>

<tr><td>Password</td><td><input type="password" name="pwd" value="" /><br /><span style="color:#FF0000"><?=$errpwd?></span></td></tr>

<?php } ?>

<?php if(empty($errgender)) { ?>

<?php if($gender=='Male') { ?>

<tr><td>Gender</td><td><input type="radio" name="gender" value="Male" checked="checked" /> Male <input type="radio" name="gender" value="Female" /> Female </td></tr>

<?php } elseif($gender=='Female') { ?>

<tr><td>Gender</td><td><input type="radio" name="gender" value="Male" /> Male <input type="radio" name="gender" value="Female" checked="checked" /> Female </td></tr>

<?php } else { ?>

<tr><td>Gender</td><td><input type="radio" name="gender" value="Male" /> Male <input type="radio" name="gender" value="Female" /> Female </td></tr>

<?php } ?>

<?php } elseif(!empty($errgender)) { ?>

<tr><td>Gender</td><td><input type="radio" name="gender" value="Male" /> Male <input type="radio" name="gender" value="Female" /> Female <br /><span style="color:#FF0000"><?=$errgender?></span></td></tr>

<?php } ?>

<?php if(empty($errlanguage)) { ?>

<?php if($language=='') { ?>

<tr><td>Language</td><td><input type="checkbox" name="language[]" value="English" /> English <input type="checkbox" name="language[]" value="French" /> French <input type="checkbox" name="language[]" value="Italian" /> Italian <input type="checkbox" name="language[]" value="Others" /> Others</td></tr>

<?php } elseif($language!='') { ?>

<tr><td>Language</td><td><input type="checkbox" name="language[]" value="English" /> English <input type="checkbox" name="language[]" value="French" /> French <input type="checkbox" name="language[]" value="Italian" /> Italian <input type="checkbox" name="language[]" value="Others" /> Others <br />You have Choose : <?php foreach($language as $key => $value) { echo $language[$key]."<br/>";} ?></td></tr>

<?php } ?>

<?php } elseif(!empty($errlanguage)) { ?>

<tr><td>Language</td><td><input type="checkbox" name="language[]" value="English" /> English <input type="checkbox" name="language[]" value="French" /> French <input type="checkbox" name="language[]" value="Italian" /> Italian <input type="checkbox" name="language[]" value="Others" /> Others <br /><span style="color:#FF0000"><?=$errlanguage?></span></td></tr>

<?php } ?>

<?php if(empty($errcountry)) { ?>

<?php if($country=='') { ?>

<tr><td>Country</td>

<td>

<select name="country">

<option value="">Choose Country</option>

<option value="India">India</option>

<option value="USA" selected>USA</option>

<option value="UK">UK</option>

<option value="Canada">Canada</option>

<option value="Other">Other</option>

</select>

</td>

</tr>

<?php } elseif($country!='') { ?>

<tr><td>Country</td>

<td>

<select name="country">

<option value="">Choose Country</option>

<option value="India">India</option>

<option value="USA" selected>USA</option>

<option value="UK">UK</option>

<option value="Canada">Canada</option>

<option value="Other">Other</option>

</select>

<br />You Choose : <?php echo $country; ?>

</td>

</tr>

<?php } ?>

<?php } elseif(!empty($errcountry)) { ?>

<tr><td>Country</td>

<td>

<select name="country">

<option value="">Choose Country</option>

<option value="India">India</option>

<option value="USA" selected>USA</option>

<option value="UK">UK</option>

<option value="Canada">Canada</option>

<option value="Other">Other</option>

</select>

<br /><span style="color:#FF0000"><?=$errcountry?></span>

</td>

</tr>

<?php } ?>

<tr><td>Message</td><td><textarea name="message"></textarea></td></tr>

<tr><td>&nbsp;</td><td><input type="submit" name="doReg" value="Registration" /></td></tr>

</table>

</form>

<?php }

if($action=='viewresults') { include 'thankyou.php'; }

?>

</body>

</html>

**Functions:**

1. **passing aurgument by reference**

**<html>**

**<head>**

**<title>Passing Argument by Reference</title>**

**</head>**

**<body>**

**<?php**

**function addFive($num) {**

**$num += 5;//$num = 10 + 5;**

**}**

**function addSix(&$num) {**

**$num += 6;**

**}**

**$orignum = 10;**

**addFive( $orignum );**

**echo "Original Value is $orignum<br />";**

**addSix( $orignum );**

**echo "Original Value is $orignum<br />";**

**?>**

**</body>**

**</html>**

**2) <html>**

**<head>**

**<title>Writing PHP Function with Parameters</title>**

**</head>**

**<body>**

**<?php**

**function addFunction($num1, $num2) {**

**$sum = $num1 + $num2;**

**echo "Sum of the two numbers is : $sum";**

**}**

**addFunction(10, 20);**

**?>**

**</body>**

**</html>**

**3) <html>**

**<head>**

**<title>Writing PHP Function which returns value</title>**

**</head>**

**<body>**

**<?php**

**function addFunction($num1, $num2) {**

**$sum = $num1 + $num2;**

**return $sum;**

**}**

**$return\_value = addFunction(10, 20);**

**echo "Returned value from the function : $return\_value";**

**?>**

**</body>**

**</html>**

**4)** **<html>**

**<head>**

**<title>Dynamic Function Calls</title>**

**</head>**

**<body>**

**<?php**

**function sayHello() {**

**echo "Hello<br />";**

**}**

**$function\_holder = "sayHello";**

**$function\_holder();**

**?>**

**</body>**

**</html>**

**OOP’S:**

1. **constructor:**

**<?php**

**function \_\_construct( $par1, $par2 ) {**

**$this->title = $par1;**

**$this->price = $par2;**

**}**

**$physics = new Books( "Physics for High School", 10 );**

**$maths = new Books ( "Advanced Chemistry", 15 );**

**$chemistry = new Books ("Algebra", 7 );**

**/\* Get those set values \*/**

**$physics->getTitle();**

**$chemistry->getTitle();**

**$maths->getTitle();**

**$physics->getPrice();**

**$chemistry->getPrice();**

**$maths->getPrice();**

**?>**

**2) Abstract classes**

**<?php**

**class Foo {**

**public static $my\_static = 'foo';**

**public function staticValue() {**

**return self::$my\_static;**

**}**

**}**

**print Foo::$my\_static . "\n";**

**$foo = new Foo();**

**print $foo->staticValue() . "\n";**

**?>**

**3)creating objects in php**

**<?php**

**$physics = new Books;**

**$maths = new Books2;**

**$chemistry = new Books3;**

**$physics->setTitle( "Physics for High School" );**

**$chemistry->setTitle( "Advanced Chemistry" );**

**$maths->setTitle( "Algebra" );**

**$physics->setPrice( 10 );**

**$chemistry->setPrice( 15 );**

**$maths->setPrice( 7 );**

**$physics->getTitle();**

**$chemistry->getTitle();**

**$maths->getTitle();**

**$physics->getPrice();**

**$chemistry->getPrice();**

**$maths->getPrice();**

**?>**

**4) Inheritence**

**<?php**

**class Novel extends Books {**

**var $publisher;**

**function setPublisher($par){**

**$this->publisher = $par;**

**}**

**function getPublisher(){**

**echo $this->publisher. "<br />";**

**}**

**}**

**?>**

**5) Function overriding:**

**<?php**

**function getPrice() {**

**echo $this->price . "<br/>";**

**return $this->price;**

**}**

**function getTitle(){**

**echo $this->title . "<br/>";**

**return $this->title;**

**}**

**?>**

**6) private function**

**<?php**

**class MyClass {**

**private $car = "skoda";**

**$driver = "SRK";**

**function \_\_construct($par) {**

**}**

**function myPublicFunction() {**

**return("I'm visible!");**

**}**

**private function myPrivateFunction() {**

**return("I'm not visible outside!");**

**}**

**}**

**?>**

**7) Protected function**

**<?php**

**class MyClass {**

**protected $car = "skoda";**

**$driver = "SRK";**

**function \_\_construct($par) {**

**}**

**function myPublicFunction() {**

**return("I'm visible!");**

**}**

**protected function myPrivateFunction() {**

**return("I'm visible in child class!");**

**}**

**}**

**?>**

**8) constant function:**

**<?php**

**class MyClass {**

**const requiredMargin = 1.7;**

**function \_\_construct($incomingValue) {**

**is created.**

**}**

**}**

**?>**

**9) Static keyword:**

**<?php**

**class Foo {**

**public static $my\_static = 'foo';**

**public function staticValue() {**

**return self::$my\_static;**

**}**

**}**

**print Foo::$my\_static . "\n";**

**$foo = new Foo();**

**print $foo->staticValue() . "\n";\*/**

**?>**

**10) Final keyword:**

**class BaseClass {**

**public function test() {**

**echo "BaseClass::test() called<br>";**

**}**

**final public function moreTesting() {**

**echo "BaseClass::moreTesting() called<br>";**

**}**

**}**

**class ChildClass extends BaseClass {**

**public function moreTesting() {**

**echo "ChildClass::moreTesting() called<br>";**

**}**

**}**

**?>**

**11) Calling parent constructor:**

**<?php**

**class Name {**

**var $\_firstName;**

**var $\_lastName;**

**function Name($first\_name, $last\_name) {**

**$this->\_firstName = $first\_name;**

**$this->\_lastName = $last\_name;**

**}**

**function toString() {**

**return($this->\_lastName .", " .$this->\_firstName);**

**}**

**}**

**class NameSub1 extends Name {**

**var $\_middleInitial;**

**function NameSub1($first\_name, $middle\_initial, $last\_name) {**

**Name::Name($first\_name, $last\_name);**

**$this->\_middleInitial = $middle\_initial;**

**}**

**function toString() {**

**return(Name::toString() . " " . $this->\_middleInitial);**

**}**

**}**

**?>**

**EXCEPTION HANDLING FUNCTIONS:**

1. **die() function:**

**<?php**

**if(!file\_exists("test.txt")) {**

**die("File not found");**

**}else {**

**$file = fopen("test.txt","r");**

**print "Opend file sucessfully";**

**}**

**exit(0);**

**?>**

1. **TRY, CATCH,& THROW:**

**<?php**

**try {**

**$error = 'Always throw this error';**

**throw new Exception($error);**

**// Code following an exception is not executed.**

**echo 'Never executed';**

**} catch (Exception $e) {**

**echo 'Caught exception: ', $e->getMessage(), "\n";**

**}**

**echo 'Hello World';**

**?>**

**Jquery:**

1. **hover:**

|  |
| --- |
|  |
| <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <script src="[https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js](https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js" \t "_blank)"></script> |
|  | <script> |
|  | $(document).ready(function(){ |
|  | $("#p1").hover(function(){ |
|  | alert("You entered p1!"); |
|  | }, |
|  | function(){ |
|  | alert("Bye! You now leave p1!"); |
|  | }); |
|  | }); |
|  | </script> |
|  | </head> |
|  | <body> |
|  |  |
|  | <p id="p1">This is a paragraph.</p> |
|  |  |
|  | </body> |
|  | </html>  **2) ID selector**   |  |  | | --- | --- | |  | <!DOCTYPE html> | |  | <html> | |  | <head> | |  | <script src="[https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js](https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js" \t "_blank)"></script> | |  | <script> | |  | $(document).ready(function(){ | |  | $("button").click(function(){ | |  | $("#test").hide(); | |  | }); | |  | }); | |  | </script> | |  | </head> | |  | <body> | |  |  | |  | <h2>This is a heading</h2> | |  |  | |  | <p>This is a paragraph.</p> | |  | <p id="test">This is another paragraph.</p> | |  |  | |  | <button>Click me</button> | |  |  | |  | </body> | |  | </html> |   **3)Htoggle:**   |  |  | | --- | --- | |  | <!DOCTYPE html> | |  | <html> | |  | <head> | |  | <script src="[https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js](https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js" \t "_blank)"></script> | |  | <script> | |  | $(document).ready(function(){ | |  | $("button").click(function(){ | |  | $("p").toggle(); | |  | }); | |  | }); | |  | </script> | |  | </head> | |  | <body> | |  |  | |  | <button>Toggle between hiding and showing the paragraphs</button> | |  |  | |  | <p>This is a paragraph with little content.</p> | |  | <p>This is another small paragraph.</p> | |  |  | |  | </body> | |  | </html> |   **4)mcrosoft cdn**   |  |  | | --- | --- | |  | <!DOCTYPE html> | |  | <html> | |  | <head> | |  | <script src="[https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.min.js](https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.min.js" \t "_blank)"> | |  | </script> | |  | <script> | |  | $(document).ready(function(){ | |  | $("button").click(function(){ | |  | $("p").hide(); | |  | }); | |  | }); | |  | </script> | |  | </head> | |  | <body> | |  |  | |  | <h2>This is a heading</h2> | |  |  | |  | <p>This is a paragraph.</p> | |  | <p>This is another paragraph.</p> | |  |  | |  | <button>Click me</button> | |  |  | |  | </body> | |  | </html> | |  |  | |
|  |  |

**ZENFRAME WORK:**

$foo = new Zend\_CodeGenerator\_Php\_Class();

$docblock = new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Sample generated class',

'longDescription' => 'This is a class generated with Zend\_CodeGenerator.',

'tags' => array(

array(

'name' => 'version',

'description' => '$Rev:$',

),

array(

'name' => 'license',

'description' => 'New BSD',

),

),

));

$foo->setName('Foo')

->setDocblock($docblock)

->setProperties(array(

array(

'name' => '\_bar',

'visibility' => 'protected',

'defaultValue' => 'baz',

),

array(

'name' => 'baz',

'visibility' => 'public',

'defaultValue' => 'bat',

),

array(

'name' => 'bat',

'const' => true,

'defaultValue' => 'foobarbazbat',

),

))

->setMethods(array(

// Method passed as array

array(

'name' => 'setBar',

'parameters' => array(

array('name' => 'bar'),

),

'body' => '$this->\_bar = $bar;' . "\n" . 'return $this;',

'docblock' => new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Set the bar property',

'tags' => array(

new Zend\_CodeGenerator\_Php\_Docblock\_Tag\_Param(array(

'paramName' => 'bar',

'datatype' => 'string'

)),

new Zend\_CodeGenerator\_Php\_Docblock\_Tag\_Return(array(

'datatype' => 'string',

)),

),

)),

),

// Method passed as concrete instance

new Zend\_CodeGenerator\_Php\_Method(array(

'name' => 'getBar',

'body' => 'return $this->\_bar;',

'docblock' => new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Retrieve the bar property',

'tags' => array(

new Zend\_CodeGenerator\_Php\_Docblock\_Tag\_Return(array(

'datatype' => 'string|null',

)),

),

)),

)),

));

echo $foo->generate();

$foo = new Zend\_CodeGenerator\_Php\_Class();

$docblock = new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Sample generated class',

'longDescription' => 'class generated with Zend\_CodeGenerator.',

'tags' => array(

array(

'name' => 'version',

'description' => '$Rev:$',

),

array(

'name' => 'license',

'description' => 'New BSD',

),

),

));

$foo->setName('Foo')

->setDocblock($docblock)

->setProperties(array(

array(

'name' => '\_bar',

'visibility' => 'protected',

'defaultValue' => 'baz',

),

array(

'name' => 'baz',

'visibility' => 'public',

'defaultValue' => 'bat',

),

array(

'name' => 'bat',

'const' => true,

'defaultValue' => 'foobarbazbat',

),

));

echo $foo->generate();

$file = new Zend\_CodeGenerator\_Php\_File(array(

'classes' => array($foo);

'docblock' => new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Foo class file',

'tags' => array(

array(

'name' => 'license',

'description' => 'New BSD',

),

),

)),

'body' => 'define(\'APPLICATION\_ENV\', \'testing\');',

));

$foo = new Zend\_CodeGenerator\_Php\_Class();

$docblock = new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Sample generated class',

'longDescription' => 'class generated with Zend\_CodeGenerator.',

'tags' => array(

array(

'name' => 'version',

'description' => '$Rev:$',

),

array(

'name' => 'license',

'description' => 'New BSD',

),

),

));

$foo->setName('Foo')

->setDocblock($docblock);

echo $foo->generate();

generator = Zend\_CodeGenerator\_Php\_Class::fromReflection(

new Zend\_Reflection\_Class($class)

);

$generator->setMethod(array(

'name' => 'setBaz',

'parameters' => array(

array('name' => 'baz'),

),

'body' => '$this->\_baz = $baz;' . "\n" . 'return $this;',

'docblock' => new Zend\_CodeGenerator\_Php\_Docblock(array(

'shortDescription' => 'Set the baz property',

'tags' => array(

new Zend\_CodeGenerator\_Php\_Docblock\_Tag\_Param(array(

'paramName' => 'baz',

'datatype' => 'string'

)),

new Zend\_CodeGenerator\_Php\_Docblock\_Tag\_Return(array(

'datatype' => 'string',

)),

),

)),

));

$code = $generator->generate();