



PHP Includes:

```
<?php include("target.php"); ?>
```

PHP Tag:

```
<?php  
?>
```

Output Text:

```
echo "Yum, Muffins!";
```

Comment:

```
Some PHP Code//My comment
```

or

```
/*My  
*/
```

Variables:

All variables start with \$

```
$var_name = value;
```

Concatenation Operator:

```
$txt1="I like";  
$txt2="Muffins!";  
echo $txt1 . " " . $txt2;
```

Operators:

Code	Purpose
+	Add Values
-	Subtract Values
*	Multiply Values
/	Divide Values
%	Division Remainder
++	Increment By 1
--	Decrement By 1

Comparison Operators:

Code	Purpose
==	Equal To
!= or <>	Not Equal
>	Greater Than
<	Less Than
>=	Greater Than Or Equal To



<=	Less Than Equal To
----	--------------------

Logical Operators:

Code	Purpose
&&	And
	Or
!	Not

If Statement:

```
if (hasMuffins=="Yes") {
    echo "Yes! Muffins!";
}
```

If Else Statement:

```
if (condition){
    echo "Yes! Muffins!";
}else{
    echo "What? No Muffins?";
}
```

Elseif Statement:

```
if (hasMuffins=="Yes"){
    echo "Yes! Muffins!";
}elseif (hasMuffins=="No"){
    echo "What? No Muffins?";
}else{
    echo "You don't know if you have Muffins?";
}
```

Switch Statement:

```
switch ($numberOfMuffins){
    case 1:
        document.write("You have 1 Muffins!");
        break;
    case 2:
        document.write("You have 2 Muffins!");
        break;
    case 3:
        document.write("You have 3 Muffins!");
        break;
    default:
        document.write("Yum, Muffins!");
}
```

Set Arrays:



```
$muffin[0]="Banana";  
$muffin[1]="All Bran";  
$muffin[2]="Plain";
```

or

```
$age = array("All Bran"=>1, "Banana"=>3, "Plain"=>2);
```

Multidimensional Arrays:

```
$pastry = array (  
    "Muffins"=>array (  
        "Plain",  
        "Banana",  
        "Blueberry"  
    ),  
    "Cupcakes"=>array (  
        "Chocolate"  
    ),  
    "Cake"=>array (  
        "Butterscotch",  
        "Strawberry",  
        "Vanilla"  
    )  
);
```

While Loop:

```
$muffins = 3;  
while ($muffins >= 1){  
    echo("Yay, Muffins!");  
    echo("On, nom!");  
    $muffins = $muffins -1;  
}  
echo("Awe, No Muffins!");
```

Do While Loop:

```
do{  
    PHP Content Here;  
}while (variable<=endvalue);
```

For Loop:

```
for ($i=1; $i<=5; $i++){  
    echo "There is " . $i . " Muffins<br />";  
}
```

For Each Loop:

```
$muffins=array("Banana","Chocolate Chip","Plain");  
foreach ($muffins as $value){
```



```
echo $value . "<br />";
}
```

Create a PHP Function:

```
function checkMuffins (){
    Some PHP Code;
}
```

Pass Function Variables:

```
function writeTypes($name){
    echo $name . " Muffins.<br />";
}

echo "We Have: ";
writeName("Banana");
writeName("Chocolate Chip");
writeName("Plain");
```

Return Values:

```
function muffinStock($baked, $eaten){
    $total=$baked-$eaten;
    return $total;
}

echo "We have this many muffins: = " . muffinStock (1,16);
```

Basic Form:

```
<form action="target.php" method="post">
    Muffin Type: <input type="text" name=" muffinType " />
    <input type="submit" />
</form>
```

Post Values:

Information invisible to others and has no limits on the amount of information

```
<?php echo $_POST["muffinType"]; ?>
```

Get Values:

Information visible to everyone and has limits on the amount of information (2000 Characters)

* Can be bookmarked*

```
<?php echo $_GET["muffinType "]; ?>
```

Request Values:

Collect form data sent with both the GET and POST methods.

```
<?php echo $_REQUEST["muffinType "]; ?>
```

Open Files:



```
$file=fopen("welcome.txt","r");
```

File Open Modes:

Code	Purpose	Note
r	Read only	Starts at the beginning of the file
r+	Read / Write	Starts at the beginning of the file
w	Write only	Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	Read / Write	Opens and clears the contents of file; or creates a new file if it doesn't exist
a	Append	Opens and writes to the end of the file or creates a new file if it doesn't exist
a+	Read/Append	Preserves file content by writing to the end of the file
x	Write only	Creates a new file. Returns FALSE and an error if file already exists
x+	Read/Write	Creates a new file. Returns FALSE and an error if file already exists

Close Files:

```
fclose($file);
```

Check End Of File:

```
if (feof($file)) echo "End of file";
```

Read Single Line:

```
echo fgets($file);
```

Upload File Script:

```
if ($_FILES["file"]["error"] > 0){  
    echo "Error: " . $_FILES["file"]["error"] . "<br />";  
}else{  
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";  
    echo "Type: " . $_FILES["file"]["type"] . "<br />";  
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";  
    echo "Stored in: " . $_FILES["file"]["tmp_name"];  
}
```

Restricted Upload File Script:

```
if (((($_FILES["file"]["type"] == "image/gif")  
|| ($_FILES["file"]["type"] == "image/jpeg")  
|| ($_FILES["file"]["type"] == "image/pjpeg"))  
&& ($_FILES["file"]["size"] < 20000)){  
    Place Upload File Script Here;  
}else{
```



```
    echo "Invalid file";  
}
```

Save the Uploaded File Script:

```
if ((($_FILES["file"]["type"] == "image/gif")  
|| ($_FILES["file"]["type"] == "image/jpeg")  
|| ($_FILES["file"]["type"] == "image/pjpeg"))  
&& ($_FILES["file"]["size"] < 20000)){  
    if ($_FILES["file"]["error"] > 0){  
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";  
    }else{  
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";  
        echo "Type: " . $_FILES["file"]["type"] . "<br />";  
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";  
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";  
        if (file_exists("upload/" . $_FILES["file"]["name"])){  
            echo $_FILES["file"]["name"] . " already exists. ";  
        }else{  
            move_uploaded_file($_FILES["file"]["tmp_name"],  
"upload/" . $_FILES["file"]["name"]);  
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];  
        }  
    }  
}  
else{  
    echo "Invalid file";  
}
```

Create a Cookie:

Must be before the <HTML> tag

```
setcookie(name, value, expire, path, domain);
```

Cookie named "user" with value "Mr.Muffin" that expires after one day:

```
setcookie("user", "Mr.Muffin", time()+86400);
```

or

```
$expire=time()+60*60*24*1;  
setcookie("user", "Mr.Muffin", $expire;
```

Retrieve a Cookie:

```
if (isset($_COOKIE["user"])){  
    echo "Welcome " . $_COOKIE["user"] . "!<br />";  
}else{  
    echo "Welcome... You.<br />";  
}
```

Delete a Cookie:

Set expiration date to the past

```
setcookie("user", "", time()-3600);
```

Starting a PHP Session:



Must be before the <HTML> tag

```
<?php session_start(); ?>
```

Storing a Session Variable:

```
session_start();  
$_SESSION['muffinType']="BlueBerry";
```

Retrieve a Session Variable:

```
echo "Our Muffin Type=". $_SESSION['muffinType'];
```

Destroying a Session:

```
unset($_SESSION['views']);
```

Or destroy the whole session

```
session_destroy();
```

mail() Function:

Note FILTER_SANITIZE_EMAIL filter removes all illegal e-mail characters from a string and FILTER_VALIDATE_EMAIL filter validates value as an e-mail address.

```
$to = "MrCupCake@example.com";  
$subject = "My inners";  
$message = "Hello! Why am I more healthy than you?.";  
$from = "MrMuffin@example.com";  
$headers = "From:" . $from;  
mail($to,$subject,$message,$headers);
```

mail() Function:

```
$to = "MrCupCake@example.com";  
$subject = "My inners";  
$message = "Hello! Why am I more healthy than you?.";  
$from = "MrMuffin@example.com";  
$headers = "From:" . $from;  
mail($to,$subject,$message,$headers);
```

Connect to a MySQL Database:

```
$host="localhost";  
$username="db100139309";  
$password="82507";  
$database="db100139309";  
  
$dbc = mysqli_connect($host, $username, $password, $database)  
or die('Error connecting to MySQL server.');
```

Interact with a MySQL Database:

```
$query = "SELECT * FROM muffins";  
$result = mysqli_query($dbc,$query)  
or die('Error querying database.');
```



```
while($row = mysqli_fetch_array($result)){  
    $types = $row[types];  
    Echo ("We have ". $types);  
}
```

Close a MySQL Database:

```
mysql_close($dbc);
```