

## php - post & get

Recall from the [PHP Forms Lesson](#) where we used an HTML form and sent it to a PHP web page for processing. In that lesson we opted to use the *post* method for submitting, but we could have also chosen the *get* method. This lesson will review both transferring methods.

tired of reading on your computer screen?  
[download our PHP Tutorial!](#)



Advertise on Tizag.com

## post - review

In our [PHP Forms Lesson](#) we used the *post* method. This is what the pertinent line of HTML code looked like:

### HTML Code Excerpt:

```
<form action="process.php" method="post">
<select name="item">
...
<input name="quantity" type="text" />
```

This HTML code specifies that the form data will be submitted to the "process.php" web page using the POST method. The way that PHP does this is to store all the "posted" values into an *associative array* called "\$\_POST". Be sure to take notice the names of the form data names, as they represent the *keys* in the "\$\_POST" associative array.

Now that you know about associative arrays, the PHP code from "process.php" should make a little more sense.

### PHP Code Excerpt:

```
$quantity = $_POST['quantity'];
$item = $_POST['item'];
```

The form names are used as the *keys* in the associative array, so be sure that you never have two input items in your HTML form that have the same name. If you do, then you might see some problems arise.

## php - get

As we mentioned before, the alternative to the *post* method is *get*. If we were to change our HTML form to the *get* method, it would look like this:

### HTML Code Excerpt:

```
<form action="process.php" method="get">
<select name="item">
...
<input name="quantity" type="text" />
```

The *get* method is different in that it passes the variables along to the "process.php" web page by appending them onto the end of the URL. The URL, after clicking submit, would have this added on to the end of it:

```
"?item=##&quantity=##"
```

The question mark "?" tells the browser that the following items are variables. Now that we changed the method of sending information on "order.html", we must change

that we changed the method of sending information on `order.html`, we must change the "process.php" code to use the "\$\_GET" associative array.

### PHP Code Excerpt:

```
$quantity = $_GET['quantity'];  
$item = $_GET['item'];
```

After changing the array name the script will function properly. Using the *get* method displays the variable information to your visitor, so be sure you are not sending password information or other sensitive items with the *get* method. You would not want your visitors seeing something they are not supposed to!

## security precautions

Whenever you are taking user input and using you need to be sure that the input is safe. If you are going to insert the data into a MySQL database, then you should be sure you have thought about preventing [MySQL Injection](#). If you are going to make a user's input available to the public, then you should think about [PHP htmlentities](#).

[Ads by Google](#) [What Is PHP](#) [PHP](#) [PHP and MySQL](#) [Tutorial](#)