Nature of form element:

1. Name: is used to specify a name for the value which is given in input field by user.
2. Radio: radio buttons treated as a group (the user should select either 0, or 1, or 2, etc.). The value attribute differs for each button to provide the different selections that can be made.
3. Select: name attribute of the select element is given as an array (with square brackets) because multiple choices are allowed (because of the presence of the multiple attribute). Without multiple, only one option would be selectable. Each option specified within the select has a distinct value.
4. Checkbox: Its value will be set to “Yes” if the box has been checked.
5. Submit button: Its value is presented as its label, and clicking on it will trigger the form submission.
6. Action: defines where the form contents are sent when the form is submitted. $\_SERVER["PHP\_SELF"] means form is submitted in the same page.
7. Method: determines how the form’s contents is submitted like whether method is post or get.
8. Form Element:

There are two specific attributes that need to be set in the form tag, action and method:

htmlspecialchars() is used here to convert specific HTML characters to their HTML entity names, e.g. > will be converted to &gt;. This prevents the form from breaking if the user submits HTML markup and also is a means of protecting against XSS (Cross Site Scripting) attacks, which attackers will use to try to exploit vulnerabilities in web applications.

1. Remember that when working with HTML forms:

All of the form controls must be enclosed within the form tags.

The alignment of text and form controls can be achieved in many ways. CSS is the preferred option for many, but be prepared to see tables used for alignment in older HTML.

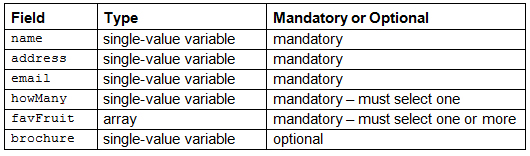
Not discussed here, but important for accessibility is the label element.

1. Form Processing

Now that you’ve defined the form in HTML, let’s identify the two stages the form goes through – when the blank form is loaded, and when the user has completed the form and clicks the submit button. The script needs to be able to differentiate between the two stages so the form behaves properly.

1. Validating the Form Contents

When the form is submitted, the following entries will be stored in the $\_POST array (or $\_GET array depending on the form’s method attribute). The values in the left-hand column are taken from the control’s name attribute, and I’ve also marked whether or not the field is a required field for validation purposes.



1. To print the value submitted in the text fields so the user does not have to retype it. The value will of course be blank if the form has just been loaded and not submitted.

To print any error messages. Again these will be blank if the form has just been loaded or if there is no error for the field.

1. We will do two more things when the user submits the form:

* Strip unnecessary characters (extra space, tab, newline) from the user input data (with the PHP trim() function)
* Remove backslashes (\) from the user input data (with the PHP stripslashes() function)