




This is a short introduction into creating simple HTML forms. Most of the content is based on HTML, with a few HTML5 additions.

## Insert Form

```
<body>  
  <form action="" method="post"  
    name="NewsletterForm">  
  </form>  
</body>
```




The image shows a screenshot of a web editor interface. On the left, there is a 'Insert' menu with various options like Form, Text Field, Hidden Field, Textarea, Checkbox, etc. A blue arrow points from the 'Form' option to a 'Tag Editor - form' dialog box. The dialog box has a 'form - General' tab with fields for Action, Method (set to 'post'), Encoding type, Name (set to 'NewsletterForm'), and Target. There are 'OK' and 'Cancel' buttons at the bottom right.

Forms can be inserted into an HTML page by using the **Insert** menu, or the Insert panel. Forms should be given a **Name** to make it easier to code their actions, however such coding will not be covered at this point. The usual method is to **Post** the forms results so that the data it contains can be stored in a database or acted on in some other way. The alternative is a Get method which is typically used for search forms where the page is updated when the search is submitted.

The **action** specifies which web page will be called after the **Submit** button is pressed.

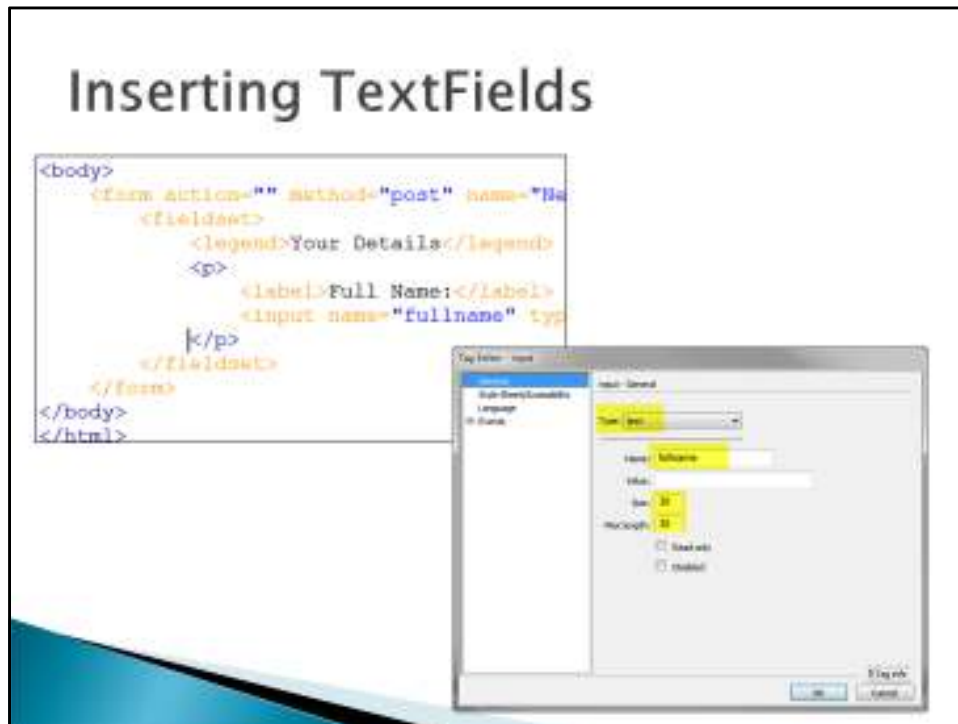
## Insert Fieldset

```
<form action="" method="post"...>
  <fieldset>
    <legend>Your
Details</legend>
  </fieldset>
</form>
```



The diagram illustrates the visual rendering of the HTML code. On the left, a 'Fieldset' dialog box is shown with a legend 'Your Details' and an input field. On the right, a 'Design View' shows the rendered form with the legend and input field.

Forms contain **fields**, and fields can be sub-divided into **fieldset**s. Each fieldset is part of the form. So for example a simple form might contain two fieldsets, the first containing the person's personal details, and the next the person's address. In this form there will only be one fieldset. Each fieldset can have a **legend** which is shown to the user to help identify what the fields are about. In this form there will only be one fieldset.



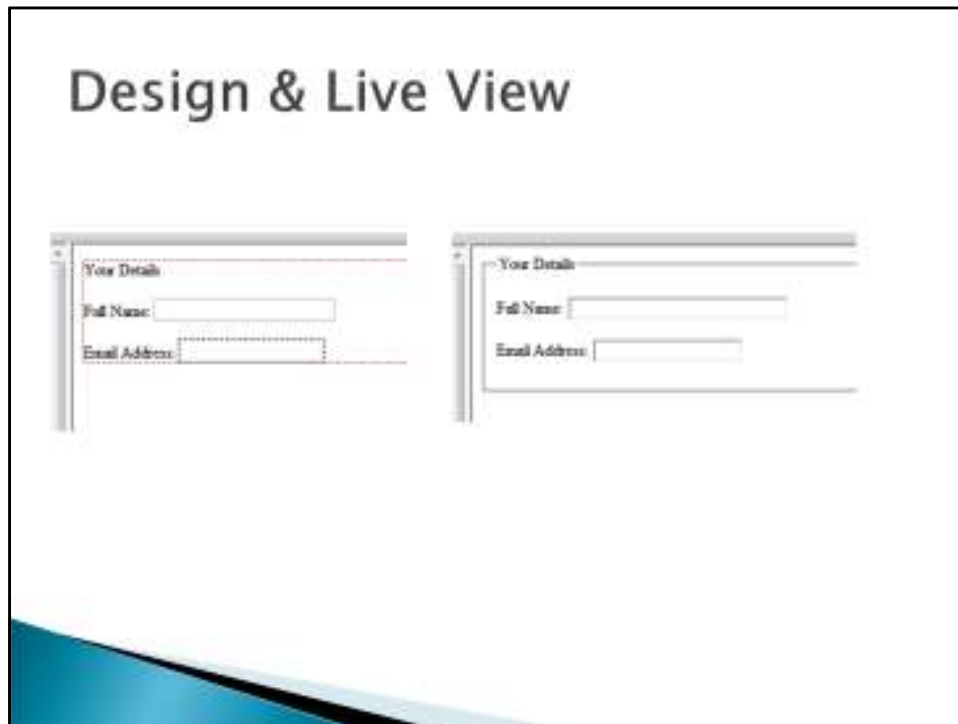
The most common field in a form is the **TextField** which can be used to take in single line text. Each TextField should be preceded by a **Label**, so the simplest way is to use design view. Press return or enter at the end of the legend to get an empty paragraph and then **Insert -> Form -> Label**. Inside the Label tag enter what the user will see (e.g. Full Name).

The TextField can either be inserted immediately after the Label, or press return or enter to get another empty paragraph so that the TextField is on the next line. Having the label on one line and the TextField on the next line is much easier to style.

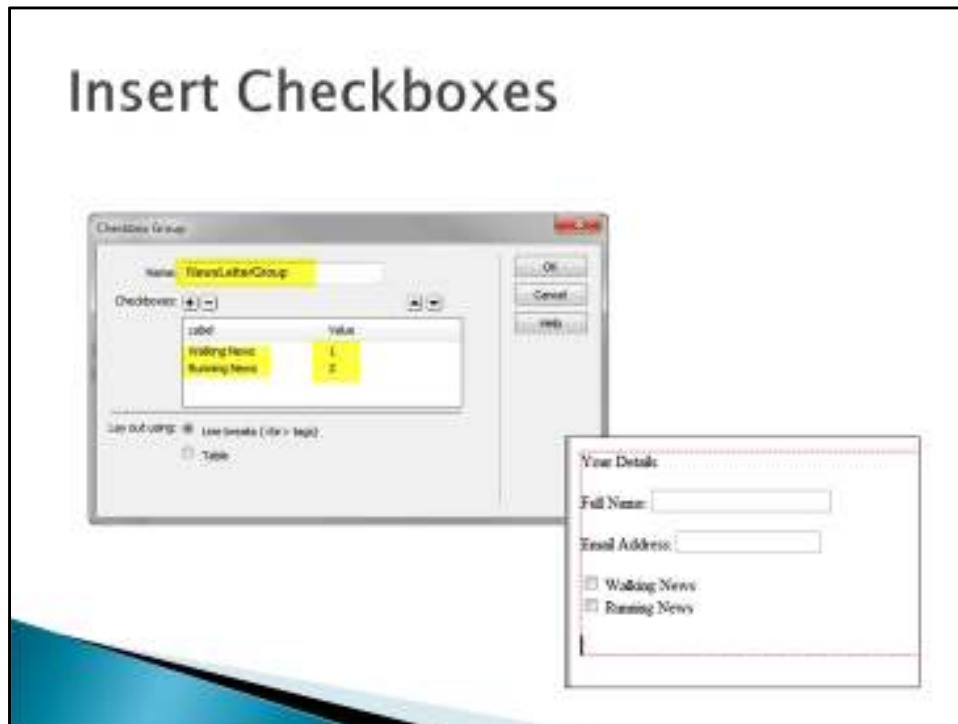
Each TextField should be given a different name, and the size of the Textfield can be set. If the Max Length is greater than the size then the text will scroll horizontally to allow the editing of text that is bigger than the text box displayed.



Dreamweaver has two dialogs for inserting TextFields, and I cannot quite work out the logic of which one will appear when you try to insert a TextField. The one shown above inserts a TextField and a label in one go. Create an empty paragraph first. Unfortunately this dialog does not allow you to change the size of the textfield. That can however be changed in code view, or in design view by selecting the TextField and changing it's properties.




In design view, or live view the disadvantage of putting Labels and TextFields on the same line can be seen. The TextFields do not line up easily, and require quite a bit of styling to get them looking good.



Checkboxes and Radio buttons are best inserted as a group. Select Insert -> Form -> Checkbox Group. Give each check box a label, and a value. The values can be names or just number them. Unfortunately the labels are put after the checkboxes, and if you want them the other way around, just move them in code view. There does not seem to be a design view way of doing that.

## Code View

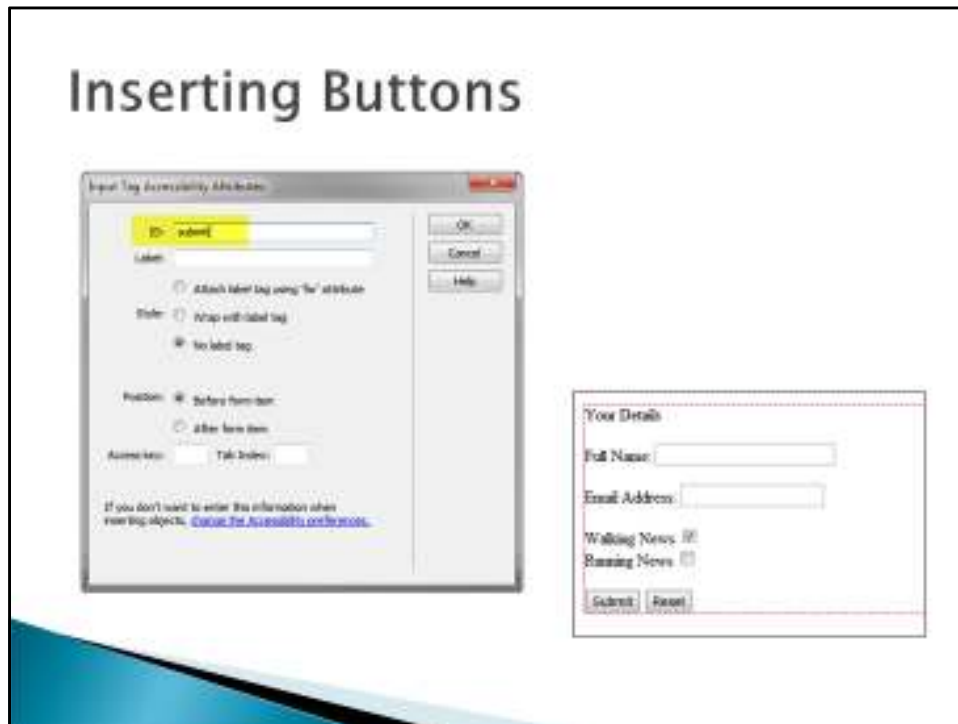
```
<p>  
  <label>  
    Walking News  
    <input name="NewsletterGroup" type="checkbox"  
      id="NewsletterGroup_0" value="1" checked/>  
  </label>  
<br>  
<label>  
  Running News  
  <input type="checkbox" name="NewsletterGroup"  
    value="2" id="NewsletterGroup_1">  
</label>  
<br>  
</p>
```



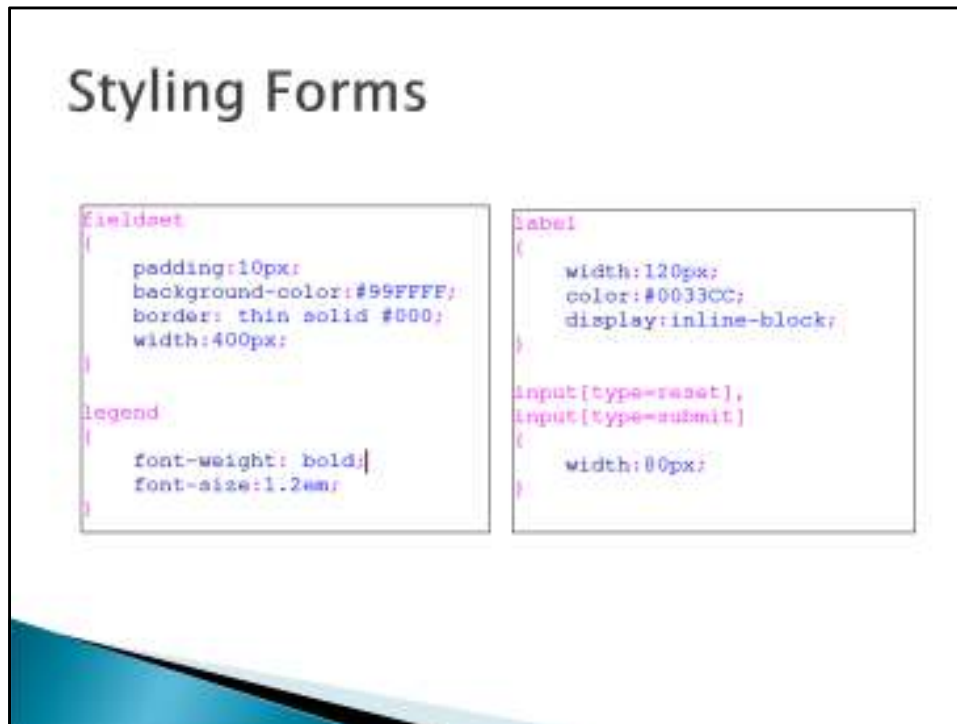
The image shows a code editor window displaying HTML code for a form. The code defines two checkboxes: 'Walking News' (checked) and 'Running News'. To the right, a preview of the rendered form is shown, titled 'Your Details'. It contains a 'Full Name' text input, an 'Email Address' text input, and two checkboxes: 'Walking News' (checked) and 'Running News' (unchecked).

Code view with the labels re-arranged so that they come before the checkbox. The values, names and IDs of the checkboxes are only important for coding purposes. If you want the checkboxes ticked by default add a checked property, this can be done in code or design view.



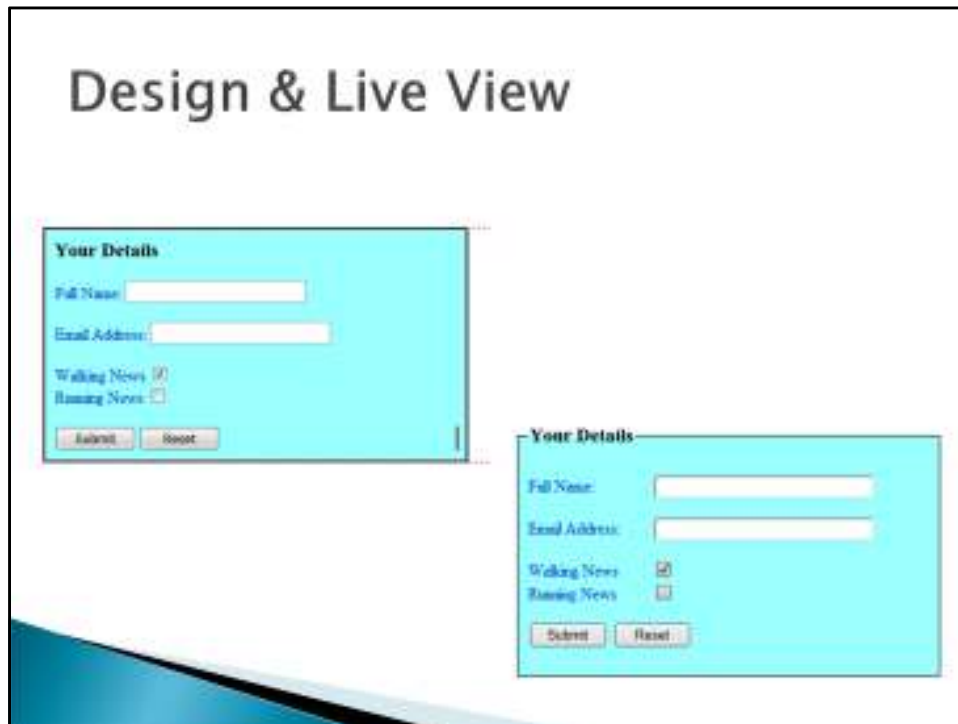


Buttons are usually added to forms without labels as they display the value attribute anyway. The value can be altered in design view (or code view), as can the action. There are only two actions, and the reset action is not that useful to users as they are unlikely to want to fill the form in twice with completely different entries. The reset action can be selected in design view.



Styling a form is fairly straight forward with two exceptions. If you want labels to be the same length, and in the same line as the TextFields then they need to be made **inline-block** and given a width. Width on its own does not work as the “extra spaces” are stripped off if it is not made block-inline.

As both TextFields and buttons use the same <input> tag, in order to style just the buttons the type attribute has to be added to the input tag in square brackets.



Design view does not show the correct spacing on the labels.

## HTML5 Additions

```
<p>  
<label for="Email">Email Address:</label>  
<input name="Email" type="email"  
  placeholder="myname@yahoo.com"  
  id="Email" size="30" maxlength="120">  
</p>  
</code>
```




HTML5 has many additions that are very useful form forms. In particular there are more types, including email. Changing **type="text"** to **type="email"** will (in modern browsers) add client side (i.e. immediate feedback) that you have entered an invalid email address. This kind of data validation greatly simplifies the need for added data validation code.

You can also add **Placeholders** which appear in grey, and disappear as soon as the user enters the TextField. Placeholders can give the user a better idea of what entry is required for the field. These additions need to be done on code view at the moment. Dreamweaver has not yet included them in design view.

## Required Fields

```
<p>  
  <label for="fullname">  
    <span>*</span>Your Name:  
  </label>  
  <input type="text" name="fullname" id="fullname"  
        placeholder="Your Name" type="text" required>  
</p>
```



```
@charset "utf-8";  
/* CSS Document */  
  
label span  
{  
  color:red;  
}
```

Required fields can simply add the required attribute to the **<input>** tag and the form will check that the user has entered an entry in the field before it is submitted for processing. However to provide the user with some indication that the field is required needs an \* symbol added to the label inside a **<span>** tag so that it can be formatted in red whilst the label itself is in a more normal colour.