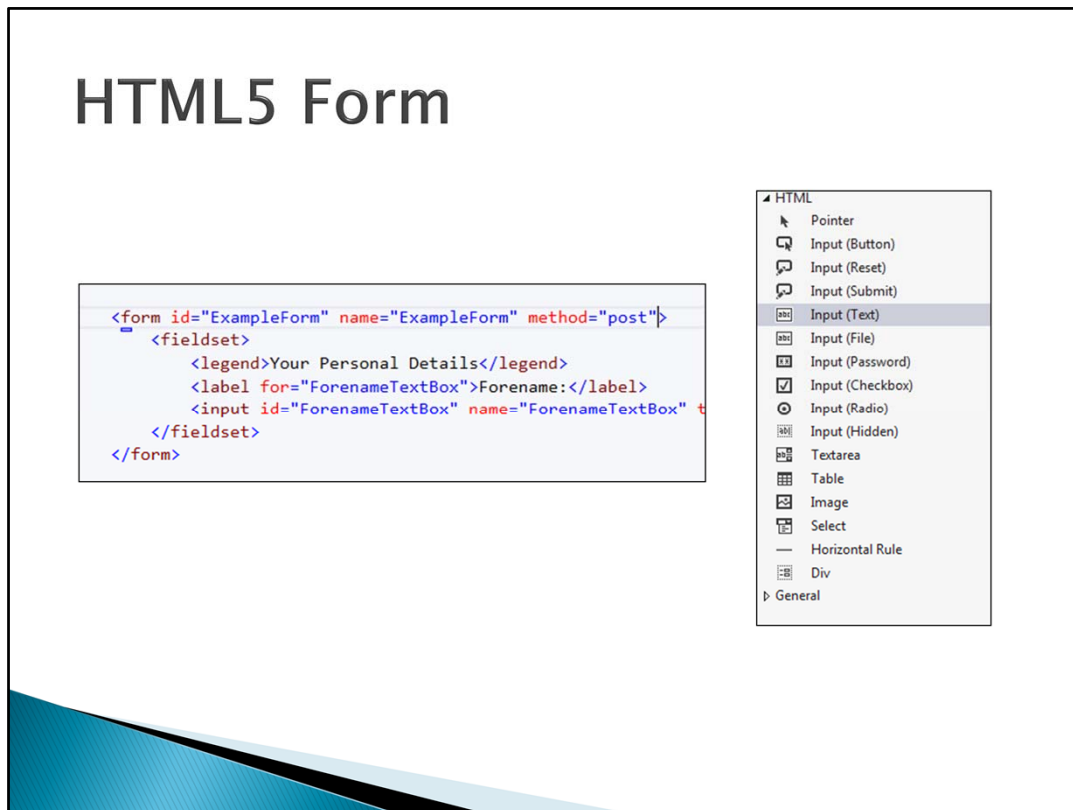




This is a short introduction into creating simple HTML forms. Most of the content is based on HTML, with a few HTML5 additions.

# HTML5 Form

```
<form id="ExampleForm" name="ExampleForm" method="post">
  <fieldset>
    <legend>Your Personal Details</legend>
    <label for="ForenameTextBox">Forename:</label>
    <input id="ForenameTextBox" name="ForenameTextBox" type="text">
  </fieldset>
</form>
```



The image shows a screenshot of an HTML5 form editor. On the left, there is a code editor displaying the following HTML code:

```
<form id="ExampleForm" name="ExampleForm" method="post">
  <fieldset>
    <legend>Your Personal Details</legend>
    <label for="ForenameTextBox">Forename:</label>
    <input id="ForenameTextBox" name="ForenameTextBox" type="text">
  </fieldset>
</form>
```

On the right, there is a list of HTML elements. The 'Input (Text)' element is selected. The list includes:

- HTML
- Pointer
- Input (Button)
- Input (Reset)
- Input (Submit)
- Input (Text)
- Input (File)
- Input (Password)
- Input (Checkbox)
- Input (Radio)
- Input (Hidden)
- Textarea
- Table
- Image
- Select
- Horizontal Rule
- Div
- General

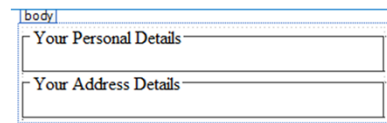
Forms should be given a **Name** and an **ID** to make it easier to code their actions, however such coding will not be covered at this point. The usual method is to **Post** the forms results so that the data it contains can be stored in a database or acted on in some other way. The alternative is a 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.

## Fieldsets

```
<form id="ExampleForm">
  <fieldset>
    <legend>Your Personal Details</legend>
    <!--Field List Goes Here-->
  </fieldset>
  <fieldset>
    <legend>Your Address Details</legend>
    <!--Field List Goes Here-->
  </fieldset>
</form>
```

### Design View

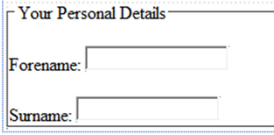


The design view shows a form with two fieldsets. The first fieldset is titled "Your Personal Details" and the second is titled "Your Address Details". Each fieldset contains a legend and a placeholder for field lists.

Forms contain **fields**, and fields can be sub-divided into **fieldsets**. 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.

## Inserting TextFields

```
<p>
  <label for="ForenameTextBox">Forename:</label>
  <input id="ForenameTextBox"
        name="ForenameTextBox"
        type="text" maxlength="30" />
</p>
<p>
  <label for="SurnameTextBox">Surname:</label>
  <input id="SurnameTextBox"
        name="SurnameTextBox"
        type="text" maxlength="30" />
</p>
```



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 drag **Input (Text)** into the paragraph. The label will have to be entered in code view. Inside the Label tag enter what the user will see (e.g. Surname).

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. Text boxes can also be separated by `</br>` or `<div>` tags.

Each TextField should be given a different name, and the size of the Textfield can be set by CSS. There is also a MaxLength attribute and if the Max Length is greater than the size (width) then the text will scroll horizontally to allow the editing of text that is bigger than the text box displayed.

## Input type=checkbox

```
<p>  
  <input id="NewsCheckbox" type="checkbox" checked="true" />  
  <label>Recieve News?</label>  
</p>  
<p>  
  <input id="UpdatesCheckbox" type="checkbox" checked="true" />  
  <label>Recieve Updates?</label>  
</p>
```

Your Personal Details

Forename:

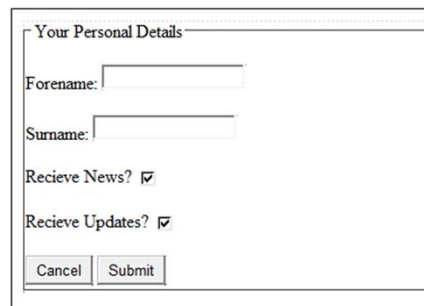
Surname:

Recieve News?

Recieve Updates?

## Form Buttons

```
<p>  
  <input id="CancelButton" type="button" value="Cancel" />  
  <input id="SubmitButton" type="submit" value="Submit" />  
</p>
```



Your Personal Details

Forename:

Surname:

Recieve News?


Recieve Updates?

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.

## Styling Forms

```
form
{
  background-color: #4cff00;
  margin: 10px;
  padding: 5px;
  width: 400px;
  border: thin solid #000000;
}

fieldset
{
  padding: 10px;
  margin: 5px;
  border: thin solid #6666;
  background-color: yellow;
}
```



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.

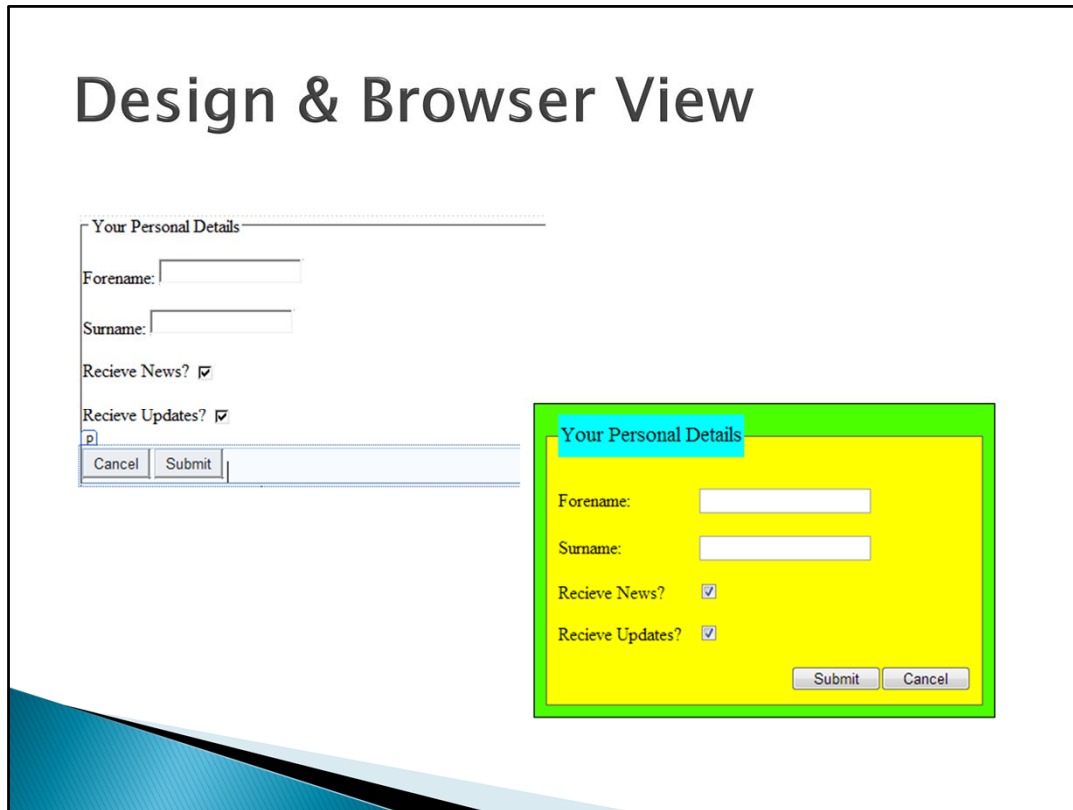
## Styling Forms (2)

```
legend
{
  line-height:2.0em;
  background-color:aqua;
  font-size:1.2em;
  font-weight:200;
}
```

```
label
{
  width:120px;
  display:inline-block;
}
```

```
input[type=button],
input[type=submit]
{
  width:80px;
  line-height:1.6em;
  float:right;
}
```





Design view does not always work for some reason. However the Page Inspector view is ok.

## HTML5 Form Validation

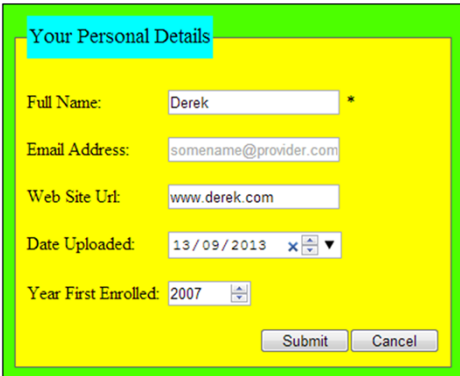
```
<p>
  <label for="FullNameTextBox">Full Name:</label>
  <input id="FullNameTextBox" type="text"
    required maxlength="10" />
  <span title="Required Field">*</span>
</p>
<p>
  <label for="EmailTextBox">Email Address:</label>
  <input id="EmailTextBox" type="email"
    placeholder="somename@provider.com" />
</p>
<p>
  <label>Web Site Url:</label>
  <input id="WebSiteTextBox" type="url" />
</p>
```

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.

## Dates & Numbers

```
<p>
  <label>Date Uploaded:</label>
  <input id="DateTextBox" type="date" />
</p>
<p>
  <label>Year First Enrolled:</label>
  <input id="YearTextBox" type="number"
    min="2000" max="2013" value="2013" />
</p>
```



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.

## Other Validation

- ▶ Date, DateTime and Time
- ▶ Month and Week
- ▶ Tel
- ▶ Search, password, hidden
- ▶ DataList
- ▶ color, image
- ▶ Autofocus, Autocomplete
- ▶ File and multiple files
- ▶ Range, Patterns, Steps