**The INPUT Tag: TEXT and PASSWORD**

**Text INPUT Field**

Creating a single line text input field is simplicity itself. You'll need the INPUT tag with its TYPE attribute set equal to **"text"**, the NAME attribute, and the SIZE attribute; the VALUE attribute is not necessary with a text INPUT field.

Tag: **INPUT**  
Attribute: **SIZE**  
Value: an integer representing the number of characters wide the TEXT INPUT field will be (using the browser's default monospace font, usually Courier or Courier New 10 point).  
Description: The SIZE attribute of the INPUT tag determines the width of a text input field.  
Example (abbreviated):

<input type="text" size="30" name="fred" />

Here is an example using the text INPUT in context:

<html>

<head>

<title>Example Form</title>

</head>

<body>

<form>

<p>First Name:<br>

<input type="text" size="30" name="firstName" /></p>

<input type="submit" value="Submit Me, Please" />

</form>

</body>

</html>

Here is the above example displayed:  
  
First Name:  
  
  


If you submit this form, you will notice that the name-value pair in the query string is**"firstName=somename"** (where **"somename"** is whatever you have typed). The VALUE of a text INPUT field is whatever the user types into the text field; this is why you don't need to set a VALUE attribute for a text INPUT field.

In the following example, I have "mistakenly" set a VALUE attribute for the text INPUT field:

<p>First Name:<br>

<input type="text" size="20" name="firstName" value="Steve" /></p>

Let us look at this example:  
  
First Name:  
  
  


The first thing you will notice is that the text INPUT field says **"Steve"** in it, which is the value of the VALUE attribute; this default text is not considered to be desirable. Users expect text INPUT fields to be blank when they encounter one; you must therefore NOT set the VALUE attribute of the INPUT tag when creating text INPUT fields.

You may also notice that you can type more content into one of these text INPUT fields. In some cases, however, you may wish to restrict the number of characters typed into the text INPUT field; to do this, you must set the MAXLENGTH attribute.

Tag: **INPUT**  
Attribute: **MAXLENGTH**  
Value: an integer representing the maximum number of characters the user may type into this particular text INPUT field.  
Description: the MAXLENGTH attribute restricts the number of characters a user may type into a text INPUT field.

Example:

<html>

<head>

<title>Example Form</title>

</head>

<body>

<form>

<p>First Name:<br>

<input type="text" size="20" maxlength="30" name="firstName" /></p>

<input type="submit" value="Submit Me, Please" />

</form>

</body>

</html>

And here is the above example displayed.  
  
First Name:  
  
  


By setting the MAXLENGTH attribute equal to **"30"**, we have restricted the number of characters the user may type into the field to 30 characters (that includes spaces, punctuation, etc). The MAXLENGTH attribute may be set to any number, regardless of the SIZE of the field; for instance, MAXLENGTH might be less than or greater than the SIZE attribute value.

You would probably only need to restrict a text INPUT field using MAXLENGTH if you know you are going to be using the information gained for a particular purpose, such as printing names and addresses onto physical mailing labels (which have a maximum number of characters they can accept before running out of space). Generally, however, you don't need to set MAXLENGTH.

**Password INPUT Field**

Password INPUT fields are identical to text input fields (including the way they use SIZE and MAXLENGTH attributes) except for their use of the **"password"** value for the TYPE attribute.

Example:

<html>

<head>

<title>Example Form</title>

</head>

<body>

<form>

<p>Login Name:<br>

<input type="password" size="30" maxlength="30" name="thePassword" /></p>

<input type="submit" value="Submit Me, Please" />

</form>

</body>

</html>

Here is the above example [displayed.](http://fog.ccsf.edu/~srubin/formPas.html)

There is only one problem with the above example: if you submit the form, you can see the supposedly "secret" password displayed in the location bar of the browser; since you're not EVER supposed to be able to see a password (that's why the letters are blocked out in the field itself), users think that there's been some sort of security breach when the password appears as part of a query string.

Whenever you use a password INPUT field in a form, you MUST set the METHOD attribute of the FORM tag equal to **"post"**; this will hide the password from the user at all times.

Example:

<html>

<head>

<title>Example Form</title>

</head>

<body>

<form method="post">

<p>Login Name:<br>

<input type="password" size="30" maxlength="30" name="thePassword" /></p>

<input type="submit" value="Submit Me, Please" />

</form>

</body>

</html>

Remember, the password INPUT field doesn't actually have any real security attached to it beyond obscuring the letters with bullets. Real security requires a secure server connection using the "https" scheme and an appropriate server. HTML forms themselves have NO security built into them.