

Forms

Forms are used to collect data inputted by a user. They can be used as an interface for a web application, for example, or to send data across the web.

On their own, forms aren't usually especially helpful. They tend to be used in conjunction with a programming language to process the information inputted by the user. These scripts take all manner of guises that are largely outside of the remit of this website because they require languages other than HTML and CSS.

The basic tags used in the actual HTML of forms are form, **input**, **textarea**, **select** and **option**.

form

form defines the form and within this tag, if you are using a form for a user to submit information (which we are assuming at this level), an **action** attribute is needed to tell the form where its contents will be sent to.

The **method** attribute tells the form how the data in it is going to be sent and it can have the value get, which is default, and latches the form information onto a web address, or post, which (essentially) invisibly sends the form's information.

***get** is used for shorter chunks of non-sensitive information - you might see the information you have submitted in a web site's search to appear in the web address of its search results page, for example.*

***post** is used for lengthier, more secure submissions, such as in contact forms.*

So a form element will look something like this:

```
<form action="processingscript.php" method="post">
</form>
```

input

The **input** tag is the daddy of the form world. It can take a multitude of guises, the most common of which are outlined below (see [the input reference page for](#) the whole crazy family):

- `<input type="text">` or simply `<input>` is a standard textbox. This can also have a `value` attribute, which sets the initial text in the textbox.
- `<input type="password">` is similar to the textbox, but the characters typed in by the user will be hidden.
- `<input type="checkbox">` is a checkbox, which can be toggled on and off by the user. This can also have a `checked` attribute (`<input type="checkbox" checked>` - the attribute doesn't require a value), and makes the initial state of the check box to be switched on, as it were.
- `<input type="radio">` is similar to a checkbox, but the user can only select one radio button in a group. This can also have a `checked` attribute.
- `<input type="submit">` is a button that when selected will submit the form. You can control the text that appears on the submit button with the `value` attribute, for example `<input type="submit" value="Ooo. Look. Text on a button. Wow">`.

Note that, like `img` and `br` tags, the `input` tag, which doesn't surround any content, doesn't require a closing tag.

textarea

textarea is, basically, a large, multi-line textbox. The anticipated number of rows and columns can be defined with **rows** and **cols** attributes, although you can manipulate the size to your heart's content using CSS.

```
<textarea rows="5" cols="20">A big load of  
text</textarea>
```

Any text you choose to place between the opening and closing tags (in this case “a big load of text”) will form the initial value of the text area.

select

The **select** tag works with the **option** tag to make drop-down select boxes.

```
<select>  
  <option>Option 1</option>  
  <option>Option 2</option>  
  <option value="third option">Option 3</option>  
</select>
```

When the form is submitted, the value of the selected option will be sent. This value will be the text between the selected opening and closing option tag unless an explicit value is specified with the **value** attribute, in which case this will be sent instead. So, in the above example, if the first item is selected, “Option 1” will be sent, if the third item is selected, “third option” will be sent.

Similar to the **checked** attribute of checkboxes and radio buttons, an **option** tag can also have a **selected** attribute, to start off with one of the items already being selected, eg. `<option selected>Rodent</option>` would pre-select “Rodent” from the items.

Names

All of the tags mentioned above will look very nice presented on the page but if you hook up your form to a form-handling script, they will all be ignored. This is because the form fields need **names**. So to all of the fields, the attribute **name** needs to be added, for example `<input type="text" name="talkingsponge">`.

A contact form for Noah's Ark, for example, might look something like the one below. (Note: this form will not work unless there is a "contactus.php" file, which is stated in the action attribute of the form tag, to handle the submitted data).

```
<form action="contactus.php" method="post">

    <p>Name:</p>
    <p><input type="text" name="name" value="Your name"></p>

    <p>Species:</p>
    <p><input name="species"></p>
    <!-- remember: 'type="text"' isn't actually necessary -->

    <p>Comments: </p>
    <p><textarea name="comments" rows="5" cols="20">Your
comments</textarea></p>

    <p>Are you:</p>
    <p><input type="radio" name="areyou" value="male">
Male</p>
    <p><input type="radio" name="areyou" value="female">
Female</p>
    <p><input type="radio" name="areyou"
value="hermaphrodite"> An hermaphrodite</p>
    <p><input type="radio" name="areyou" value="asexual">
Asexual</p>

    <p><input type="submit"></p>

</form>
```