

Forms

Taken from: <http://www.htmldog.com/guides/htmlbeginner/forms/>

The Links will connect to [htmldog.com](http://www.htmldog.com) website for more information

Forms can be used to send data across the web and are often used as **contact forms** to convert information inputted by a user into an email, such as [the one used on this website](#).

On their own, forms are useless. They need to be hooked up to a program that will process the data inputted by the user. These take all manner of guises and are outside of the remit of this website. If you use an internet service provider to host your HTML, they will be able to help you with this and will probably have clear and simple instructions on how, for example, to make a form-to-email form work.

The basic tags used in the actual HTML of forms are [form](#), [input](#), [textarea](#), [select](#) and [option](#).

[form](#) defines the form and within this tag, there is one required action attribute which tells the form where its contents will be sent to when it is submitted.

The optional `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) or `post`. This is commonly used, and often set to `post` which hides the information (`get` latches the information onto the URL).

So a form element will look something like this:

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

```
</form>
```

The [input](#) tag is the daddy of the form world. It can take ten forms, outlined below:

- `<input type="text" />` 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, which would be used in the format `<input type="checkbox" checked="checked" />`, 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, used in the same way as the checkbox.
- `<input type="file" />` is an area that shows the **files** on your computer, like you see when you open or save a document in most programs, and is used to enable users to upload files.
- `<input type="submit" />` is a button that when selected will **submit** the form. You can control the text that appears on the submit button (as you can with button and reset types - see below) with the `value` attribute, for example `<input type="submit" value="Ooo. Look. Text on a button. Wow" />`.
- `<input type="image" />` is an **image** that will submit the coordinates of where the user clicked on it. This also requires a `src` attribute, like the [img](#) tag.
- `<input type="button" />` is a **button** that will not do anything without extra code added.
- `<input type="reset" />` is a button that when selected will **reset** the form fields to their default values.
- `<input type="hidden" />` is a field that will not be displayed and is used to pass information such as the page name that the user is on or the email address that the form should be posted to.

Note that the [input](#) tag closes itself with a `</>` at the end.

A **textarea** is, basically, a large textbox. It requires a `rows` and `cols` attribute and is used like this:

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

The [select](#) tag works with the [option](#) tag to make **drop-down select boxes**.

They work like this:

```
<select>
  <option value="first option">Option 1</option>
  <option value="second 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.

Similar to the `checked` attribute of checkboxes and radio buttons, an [option](#) tag can also have a `selected` attribute, which would be used in the format `<option value="mouse" selected="selected">Rodent</option>`.

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 program, 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 form might look 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>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" checked="checked"  
/> Asexual</p>
```

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

```
  <p><input type="reset" /></p>
```

```
</form>
```