

This page explains how you can embed input forms into wiki pages.

Input forms don't actually handle processing of the form data -- the feature simply allows creation of forms inside wiki pages. Forms processing can be found in the Cookbook (see below).

Markup

Two directives are used to begin and end forms:

```
(:input form "url" method:) ... (:input end:)
```

The (:input form:) directive starts a form that will post to *url* (optional *action=url*) using the supplied *method* (optional *method=method*). The *url* must be in quotes if not specified via *action=*. If the url is omitted, then the current page is assumed. If *method* is omitted then "POST" is assumed. An optional name="FormName" argument can be used to name the form. You can explicitly state *action=url* or *method=get* or you can simply use them as positional parameters.

If your site uses ?n=Group.Page to specify the pagename then having a field (:input hidden name=*n* value={\$FullName}:) will allow your form to post to the current page as an alternative to fully specifying the *action=url*.

The (:input end:) directive ends the current form.

Note that this feature doesn't ensure that the form output is correct HTML -- it assumes the author knows a little bit of what he or she is doing. Notably, (:input form:) and (:input end:) shouldn't appear inside tables, and all form fields and controls should be inside an (:input form:)...(:input end:) block.

Standard input controls

The standard input controls are:

```
(:input text name value size=n:) (:input hidden name value:) (:input password name value:) (:input radio name value:) (:input checkbox name value:) (:input select name value label:) (:input default default-name default-value:) (:input submit name value:) (:input textarea name [=value=] rows=n cols=n:) (:input reset name label:) (:input file name label:) (:input image name "src" alt:)
```

Where *name* and *value* are in the HTML syntax: name="addr" value="808 W Franklin".

For most controls the markup has the form:

```
(:input type name value parameter=value:)
```

where *type* is the type of input element (described below), *name* is the name of the control, *value* is its initial value, and *parameters* are used to specify additional attributes to the control. If *value* contains spaces, enclose it in quotes; if it contains newlines (for textarea and hidden elements), enclose it in [=...=].

For example, the following creates a text input control with a size of 30 characters:

```
(:input text authorid "Jane Doe" size=30:)  
Jane Doe
```

For convenience, an author can also specify *name* and *value* arguments directly using *name=* and *value=* attributes (same as HTML):

```
(:input text name=authorid value="Jane Doe" size=30:)  
Jane Doe
```

For the `textarea` control a value can be set from [Pm Wiki](#) 2.2.0beta45 onwards. Enclose the value in [=...=] if it contains spaces or new lines.

The `submit` control will more often be written as:

```
(:input submit value=label:)
```

Here's a more complete example, e.g., for a login prompt:

```
(:input form "http://www.example.com":) (:input  
hidden action login:) || Name:||(:input text  
username:) || || Password:||(:input password  
password:)|| || ||(:input submit value="Log In":)  
|| (:input end:)
```

```
login
```

Name:

General form field attributes

- (:input ... focus=1:) Setting `focus=1` causes that field to receive the initial focus when the form is first opened.
- The following advanced HTML attributes are supported: `name`, `value`, `id`, `class`, `rows`, `cols`, `size`, `maxlength`, `action`, `method`, `accesskey`, `tabindex`, `multiple`, `checked`, `disabled`, `readonly`, `enctype`, `src`, `alt`. For a more detailed description, see their counterparts in the [w3c reference: HTML forms](#) (not all of them can be used for all types of form fields).

```
(:input select ... :)
```

The basic form of a select box is a sequence of options:

```
(:input form:) (:input select name=abc value=1 label=alpha :) (:input select  
name=abc value=2 label=beta :) (:input select name=abc value=3 label=gamma :)  
(:input submit:) (:input end:)
```

alpha	▼	
-------	---	--

The values can be specified positionally:

```
(:input select abc 1 alpha :)
```

We can specify the size of the selection box:

```
(:input select abc 1 alpha size=3 :)
```

You can specify a multiple select box:

```
(:input select abc 1 alpha size=3 multiple:)
```

To have an element selected, use `selected=selected`:

```
(:input select abc 2 beta selected=selected:)
```

Note that to have two select boxes inline, not only should you give them different `name=` parameters, but also place a separator, like a character, ` ` or even the null sequence `[==]` between them:

```
(:input form:) (:input select name=FIRST value=1:)(:input select name=FIRST  
value=2:)[==] (:input select name=SECOND value=3:)(:input select name=SECOND  
value=4:)(:input end:)
```

1	▼
3	▼

See Also

- [Cookbook:Input Default](#)
- [Cookbook:Form Validation](#)
- [Cookbook:Form Extensions](#)
- [Cookbook:Input Forms and JavaScript](#)

Compatible recipes:

- [Cookbook:PmForm](#)
- [Cookbook:Fox](#)
- [Cookbook:Wiki Forms](#)
- [Cookbook:ProcessForm](#)