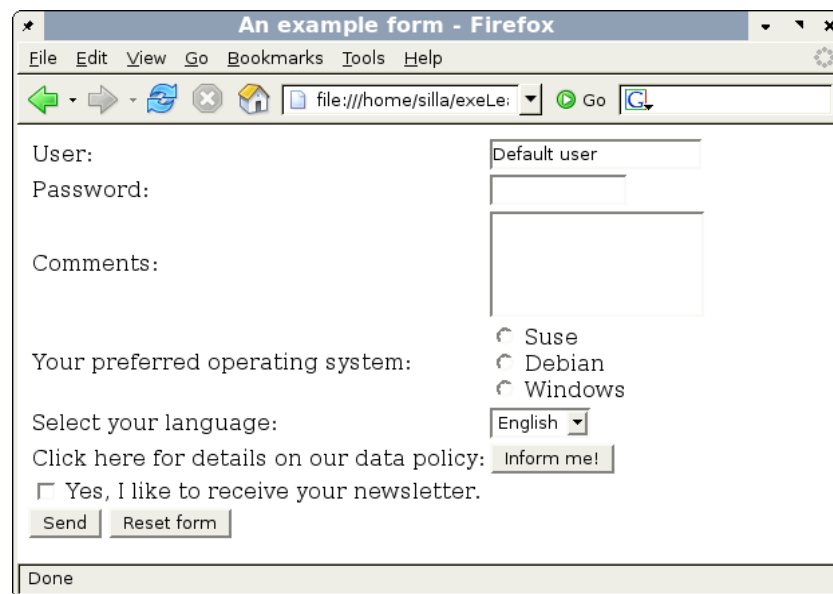


Embedding Forms

An HTML form is a section of a document containing text, special elements called controls (checkboxes, radio buttons, menus, etc.), and labels on those controls. Users enter information into a form (thereby "completing" a form) by modifying its controls (entering text, selecting menu items, etc.). Then they press a button in the form to submit the information to an agent for processing (e.g., to a web server, to a mail server, etc.). Here is an example for a form, containing all the controls that we will be discussing in this tutorial:



The screenshot shows a Firefox browser window with the title "An example form - Firefox". The address bar shows a local file path: "file:///home/silla/exeLe:". The form contains the following elements:

- User: A text input field with "Default user" entered.
- Password: An empty text input field.
- Comments: A large text area.
- Your preferred operating system: Three radio buttons labeled "Suse", "Debian", and "Windows".
- Select your language: A dropdown menu showing "English".
- Click here for details on our data policy: A button labeled "Inform me!".
- Yes, I like to receive your newsletter.
- Send and Reset form buttons.

To create a form, use the "form" tag:

```
<form action="process_form.php" method="post"
  enctype="text/plain">

</form>
```

action

This attribute specifies a form-processing agent. This is usually a script (e.g. a PHP script)

method = get|post

This attribute specifies which HTTP method will be used to submit the form data set. Possible (case-insensitive) values are "get" (the default) and "post".

post

the user agent conducts an HTTP "post" transaction using the value of the action attribute and a message created according to the content type specified by the enctype attribute.

get

the user agent takes the value of action, appends a '?' to it, then appends

the form data, encoded using the "application/x-www-form-urlencoded" content type. The user agent then traverses the link to this URI. In this scenario, form data is restricted to ASCII codes.

enctype

The enctype attribute of the "form" element is needed when using the "post" method. It specifies the content type used to encode the form data for submission to the server.

Textfields

If you would like the user to enter text (e.g. his or her name, a password, or longer text), use either the element "input" (with two different possible types) or the element "textarea".

An "input" element of type "text" is a single line of specific length, where the length is specified with the attribute "size". Type "password" is similar, but the text entered by the user does not show – instead, an asterisk (*) is displayed for each character. This is how you use them (compare the example HTML page to see what the result looks like):

```
<input type="text" size="20" value="Default user" name="user" >  
<input type="password" size="12" name="user_password">
```

The attribute "value" above can be used to specify the initial content of the text input line. Every control element has the attribute "name" that identifies the element in the form and can be used by the processing script to access its content.

Lastly, "textarea" is a textbox spanning several lines, where the user can enter as much text as he or she likes. The number of lines displayed in the textbox is specified by the "rows" attribute. The number of characters per line (the width of the box) is specified with the "cols" attribute. If the user enters more text than can be displayed in this area, scrollbars automatically appear on the side and/or bottom of the box. This is how you use it:

```
<textarea rows="4" cols="20" name="comments"></textarea>
```

Radio-buttons

Radio buttons are used to choose between possible answers, where only one choice is possible. By giving several radio buttons the same name, you specify that they belong to one choice, and that the user may only select one of them:

```
<input type="radio" name="os" value="Suse">  
<input type="radio" name="os" value="Debian">  
<input type="radio" name="os" value="Windows">
```

Drop-down menus

```
<select name="...">
  <option value="E">English</option>
  <option value="F">French</option>
</select>
```

The attribute “value” in “select” elements is optional – if it is given, this value is transmitted to the processing agent when the user chooses that option. Otherwise, the text between the element tags (here “English” or “French”) is transmitted.

Buttons

A button needs to be assigned an action that is carried out when the button is clicked, otherwise clicking the button will not do anything. The action needs to be specified in something other than pure HTML (e.g. JavaScript), so we will not go into detail about that here. Here is an example of a button with associated JavaScript code: when clicked, a pop-up window will come up and display the sentence “We will not pass on your data to anyone”.

```
<input type="button" name="info_button" value="Inform me!"
onclick="alert('We will not pass on your data to anyone.');">
```

Apart from normal buttons, there are two types of buttons that have special pre-defined actions associated with them: the “submit” button and the “reset” button. The “submit” button sends the form off to the processing agent, while the “reset” button clears all the data entered into a form. These buttons are usually located at the bottom of the form.

```
<input type="submit" name="send_button" value="Send">
<input type="reset" name="reset_button" value="Reset form">
```

Checkboxen

```
<input type="checkbox" name="newsletter" value="yes">
Yes, I'd like to receive your newsletter.
```