CHAPTER 5 HTML Forms

- HTML forms are used to pass data to a server.
- The most important form element is the input element.
- The ***input element*** is used to select user information.
- An input element can vary in many ways, depending on the type attribute.
- A form can contain input elements like text fields, checkboxes, radio buttons, submit buttons and more. A form can also contain select lists, text area, field set, legend, and label elements.
- The <form> tag is used to create an HTML form:

```
<form>


input elements
    .
 </form>
```

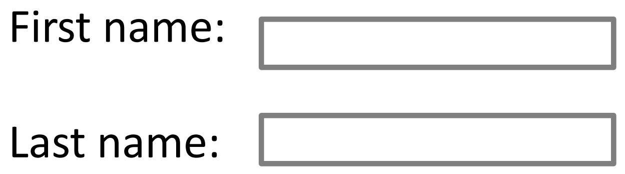
Input elements

Text Fields

```
<input type="text" />
```

- defines a one-line input field that a user can enter text into:

```
<form>
First name:
 <input type="text" name="firstname" /><br />
 Last name:
<input type="text" name="lastname" />
</form>
```



First name:

Last name:

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Input elements

<input> auto complete Attribute

- Auto complete on (and off for one input field):
- The auto complete attribute specifies whether or not an input field should have auto complete enabled.
- Auto complete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.
- **Note:** The auto complete attribute works with the following <input> types: text, search, url, tel, email, password, date pickers, range, and color.

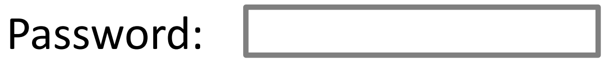
Input elements

Password Field

```
<input type="password" />
```

- defines a password field. This feature allows the user to enter a password that does not appear on screen but will be sent to you. The command for the password is similar to a text line and can be edited the same way.

```
<form>
 Password:
 <input type="password" name="pwd" />
</form>
```


Password:

Note: The characters in a password field are masked (shown as asterisks or circles).

Input elements

- ***Textarea***
- This command allows you to generate a text box on your form for user input, not just a line of wrapping text. The basic command for this is:

```
<TEXTAREA name="anyname"> </TEXTAREA>
```

- We can edit this by adding more attributes within the tag. Columns and rows can be described. Also text added between the starting and ending Textarea tags appear within the text box. This text is formatted exactly as typed including tabs, spacing and returns.

Example:

```
<p> Please add any comments you may have about this form class here</p>
<TEXTAREA name="Comments" rows="9" cols="44">
	Constructive criticism carries more clout than negative does.
		Tabs and  returns work within TEXTAREAS.
 </TEXTAREA>
```

Constructive criticism carries more clout than negative does. Tabs and returns work within TEXTAREAS.

Input elements

<u>Push buttons</u>

Reset button, Submit button and Javascript button

- The ***Reset*** button allows the user to clear the data they have entered in the form and start fresh.
- A ***submit*** button is used to send form data to a server.
- These buttons are created with the INPUT command and the TYPE and VALUE features. The INPUT starts the tag. The TYPE is either Reset or Submit. The VALUE is the words that you want to appear in the box.

Input elements

Submit Button

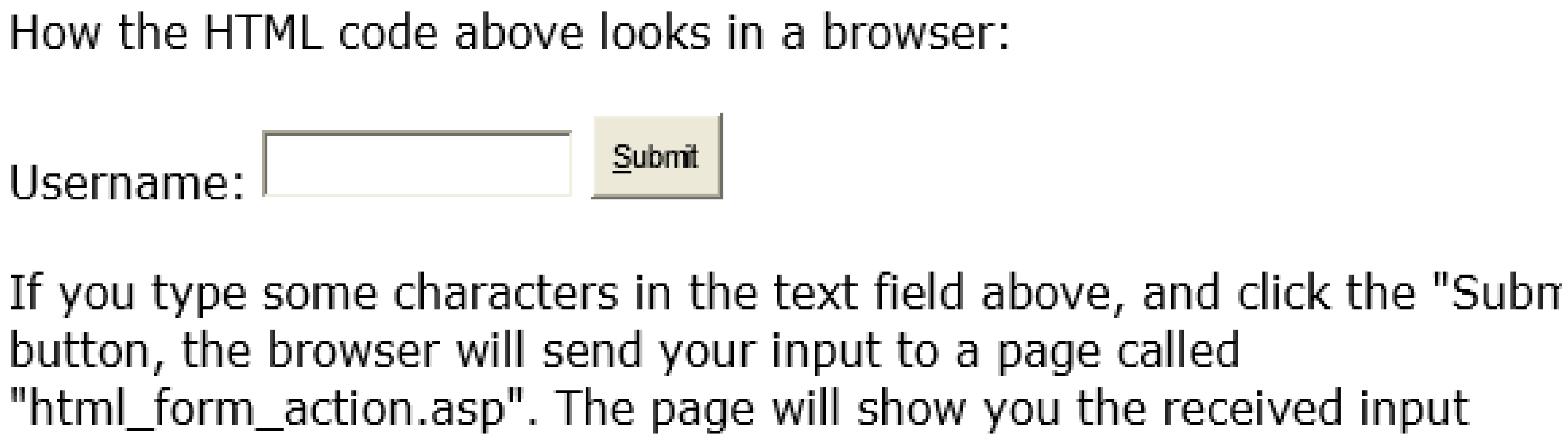
```
<input type="submit" />
```

- defines a submit button.
- A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp"
   method="get">
 Username:
 <input type="text" name="user" />
 <input type="submit" value="Submit" />
</form>
```

Input elements

Submit Button



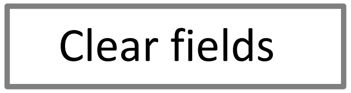
How the HTML code above looks in a browser:

Username:

If you type some characters in the text field above, and click the "Subm
button, the browser will send your input to a page called
"html_form_action.asp". The page will show you the received input

Reset button

```
<INPUT type="reset" value="clear fields">
```

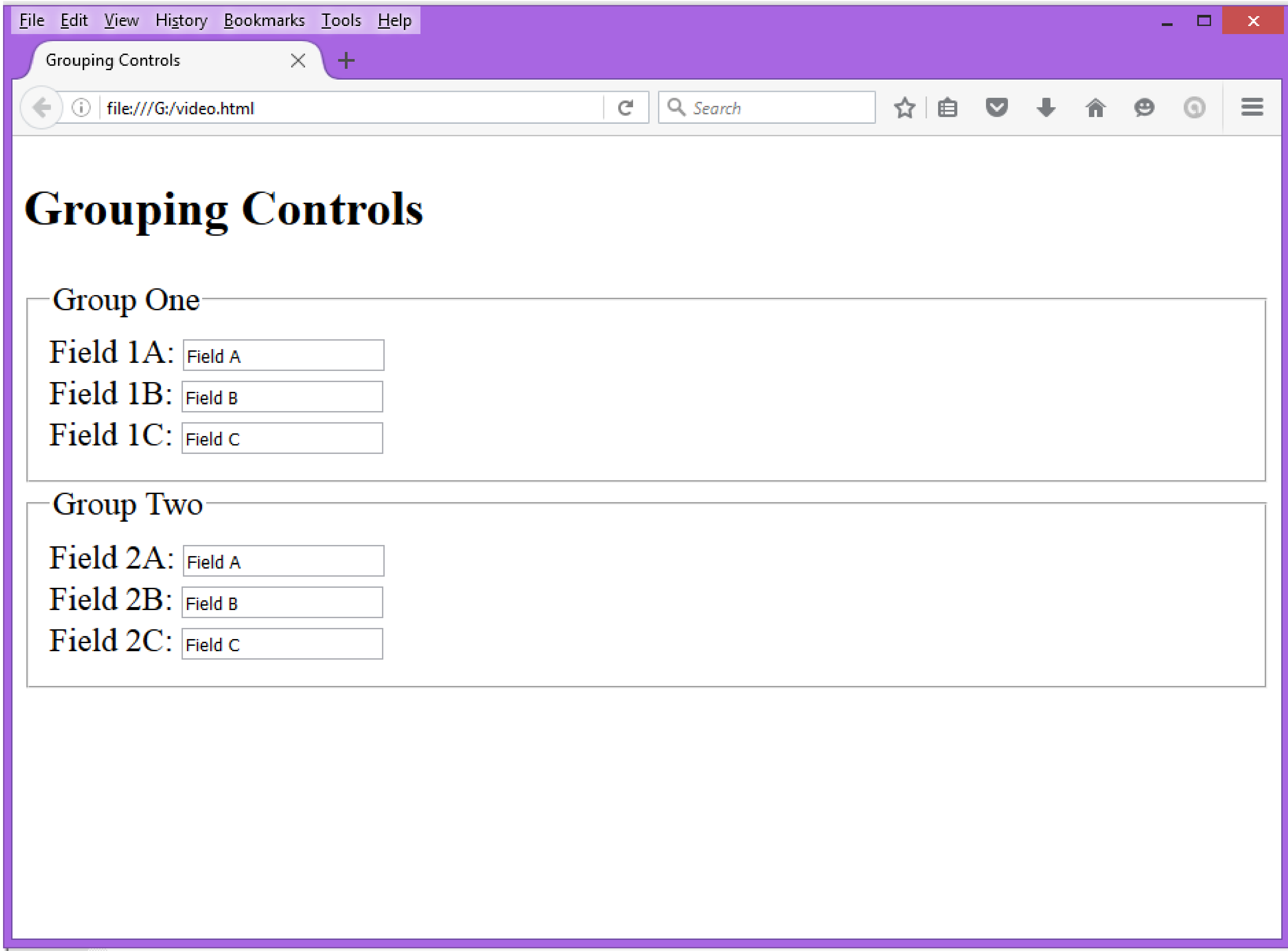


Groups of Controls

HTML Element: <FIELDSET> ... </FIELDSET>
Attributes: None.

- This element is used as a container to enclose controls and, optionally, a LEGEND element. It has no attributes beyond the universal ones for style sheets, language, and so forth.

```
<!DOCTYPE html>
<HTML>
<HEAD>  <TITLE>Grouping Controls</TITLE> </HEAD>
<BODY>
 <h2 >Grouping Controls</h2>
 <form  action="http://localhost:8088/SomeProgram">
 <fieldset> <legend>Group One</legend>
    Field 1A: <input type="text" name="field1A" value="Field A"><br>
     Field 1B:   <input type="text" name="field1B" value="Field B"><br>
     Field 1C: <input type="text" name="field1C" value="Field C"><br>
  </fieldset>
  <fieldset>
 <legend align="right">Group Two</legend>
   Field 2A: <input type="text" name="field2A" value="Field A"><br>
   Field 2B: <input type="text" name="field2B" value="Field B"><br>
   Field 2C: <input type="text" name="field2C" value="Field C"><br>
</fieldset> </form>
 </BODY>
</HTML>
```



Grouping Controls

Group One

Field 1A: Field A
Field 1B: Field B
Field 1C: Field C

Group Two

Field 2A: Field A
Field 2B: Field B
Field 2C: Field C

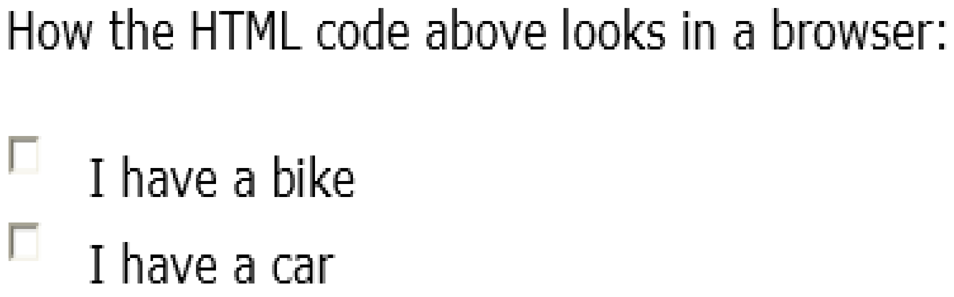
Input elements

Checkboxes

```
<input type="checkbox" />
```

- defines a checkbox. Checkboxes let a user select ONE or MORE options of a limited number of choices. Add ***checked*** to indicate default checked box.

```
<form>
 <input type="checkbox" name="vehicle" value="Bike" /> I have
    a bike<br />
 <input type="checkbox" name="vehicle" value="Car" /> I have a
    car
</form>
```


How the HTML code above looks in a browser:

☐ I have a bike

☐ I have a car

Input elements

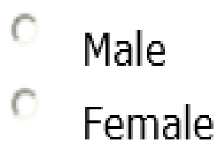
Radio Buttons

```
<input type="radio" />
```

- defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices. All radio buttons in a group should have same name:

```
<form>
 <input type="radio" name="“gender" value="male" />
   Male<br />
<input type="radio" name="“gender" value="female" />
   Female
</form>
```

How the HTML code above looks in a browser:



Combo Boxes

- These lists are drop-down windows in which a user selects a choice from a list of options selected by the programmer. The code for an option select list with three choices follows.
- Select gives a name.
- Option gives a value.

```
<SELECT  type="text" name=“language" >
       <OPTION value=“c">C
        <OPTION value=“c++">C++
       <OPTION value=“html">HTML
</SELECT>
```

List Boxes

- Identical to combo boxes but specify multiple.

```
<SELECT  name=“language" MULTIPLE>
    <OPTION value=“c">C
     <OPTION value=“c++">C++
    <OPTION value=“html">HTML
</SELECT>
```

File upload control

- Lets a user select a file and send it to the server.
- HTML Element:

  <INPUT TYPE="FILE“ name=“” ...>

(No End Tag) Attributes: NAME (required), VALUE (ignored), SIZE, MAXLENGTH, ACCEPT, ONCHANGE, ONSELECT, ONFOCUS, ONBLUR (nonstandard)

- This element results in a filename textfield next to a Browse button. Users can enter a path directly in the textfield or click on the button to bring up a file selection dialog that lets them interactively choose the path to a file.
- When the form is submitted, the contents of the file are transmitted as long as an ENCTYPE of multipart/formdata was specified in the initial FORM declaration. For multipart data, you also need to specify POST as the method type. This element provides a convenient way to make user-support pages, with which the user sends a description of a problem along with any associated data or configuration files.