Corso di laurea in Informatica

Programmazione WEB

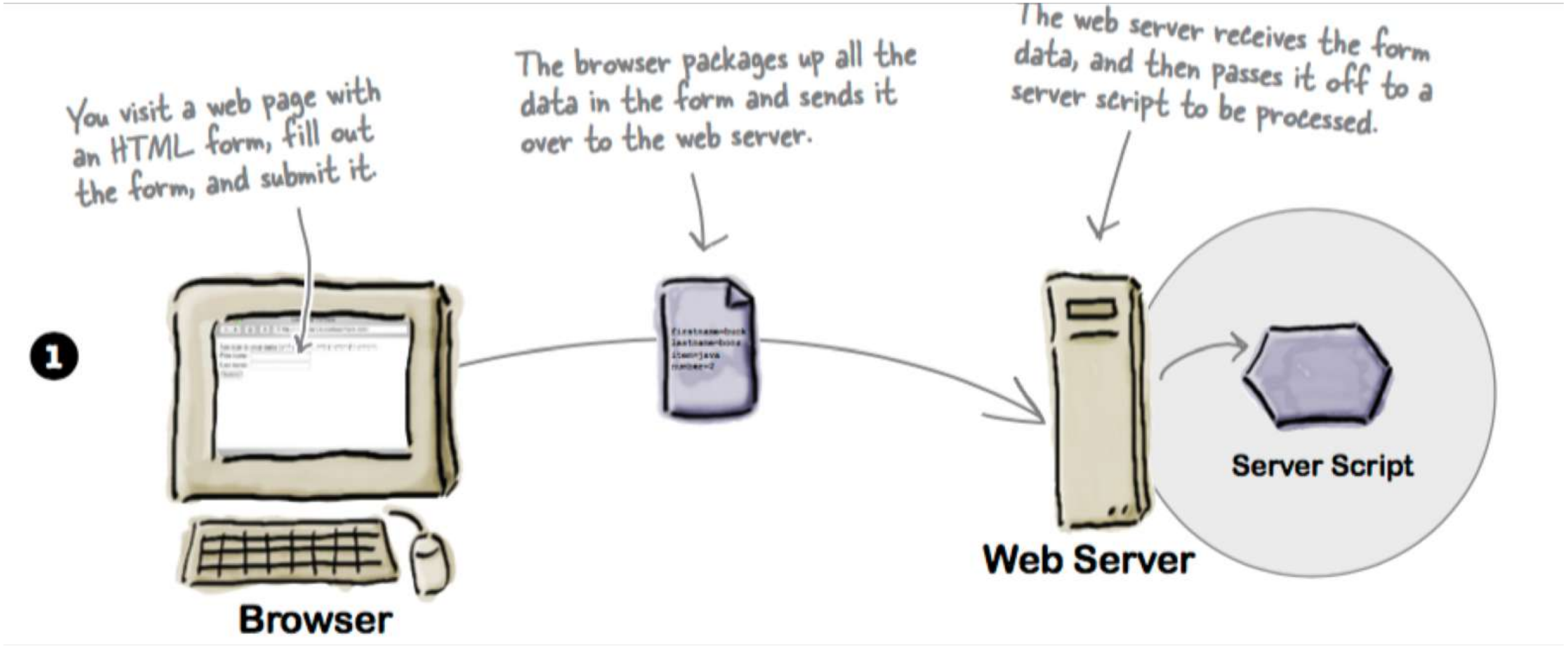
HTML - FORM

a.a 2018-2019

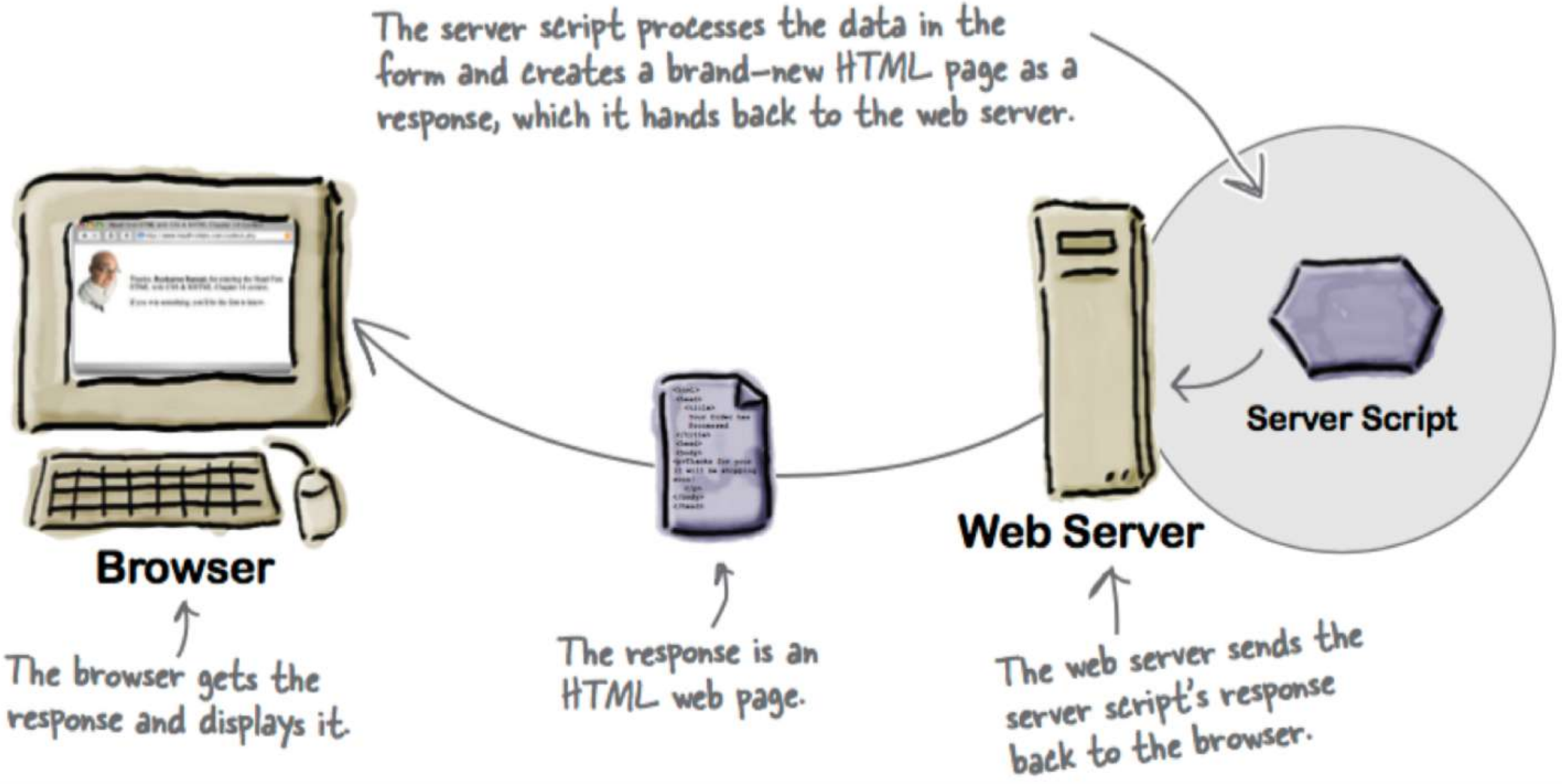
Form

- Un form (modulo) è una sezione di documento HTML che contiene elementi di controllo che l’utente può utilizzare per inserire dati o in generale per interagire
- I dati inseriti possono essere poi inoltrati al server dove un “agente” può processarli
- Gli elementi di controllo sono caratterizzati da un valore iniziale e da un valore corrente
- Gli elementi di controllo possono essere:
  - Bottoni di azione
  - Checkbox (caselle di spunta)
  - Radio Button (bottoni mutuamente esclusivi)
  - Liste di selezione (lista di opzioni)
  - Caselle di inserimento di testo
  - Oggetti nascosti (elementi valorizzati ma invisibili)
  - Selezione file

Forms


You visit a web page with an HTML form, fill out the form, and submit it.
The browser packages up all the data in the form and sends it over to the web server.
The web server receives the form data, and then passes it off to a server script to be processed.
1
firstname=buck
lastname=bone
item=java
number=2
Browser
Web Server
Server Script

2


The server script processes the data in the form and creates a brand-new HTML page as a response, which it hands back to the web server.
Server Script
Web Server
Browser
The browser gets the response and displays it.
The response is an HTML web page.
The web server sends the server script's response back to the browser.

<form> tag

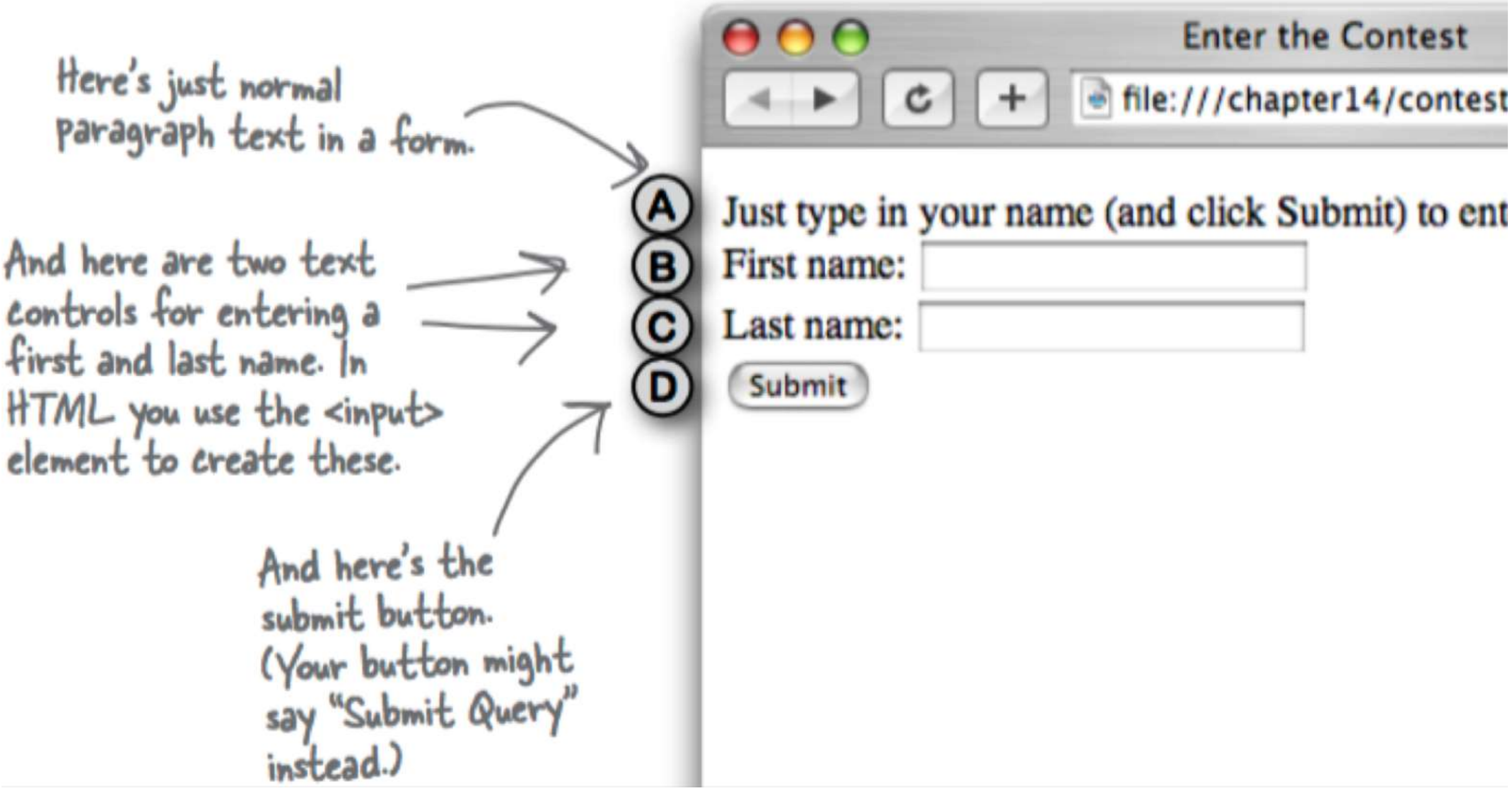
- Il tag **<form>** racchiude tutti gli elementi del modulo (è un elemento di tipo blocco)
- Attributi:
  - **action = uri** (URI dell'agente che riceverà i dati del form)
  - **name = text** (specifica il nome della form)
  - **method = {get|post}** (specifica il modo in cui i dati vengono inviati)
  - **enctype = content-type** se il metodo è post specifica il content type usato per la codifica (encoding) dei dati contenuti nel form
    - **application/x-www-form-urlencoded** (default)
    - **multipart/form-data**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Enter the Contest</title>
  </head>
  <body>
```

```
    <form action="http://wickedlysmart.com/hfhtmlcss/contest.php"
          method="POST">
(A)     <p>Just type in your name (and click Submit) to
          enter the contest: <br>

(B)      First name: <input type="text" name="firstname" value=""> <br>
(C)      Last name: <input type="text" name="lastname" value=""> <br>
(D)      <input type="submit">

        </p>
    </form>

  </body>
</html>
```

We've got the <form> element itself...

...and a bunch of elements nested inside it.

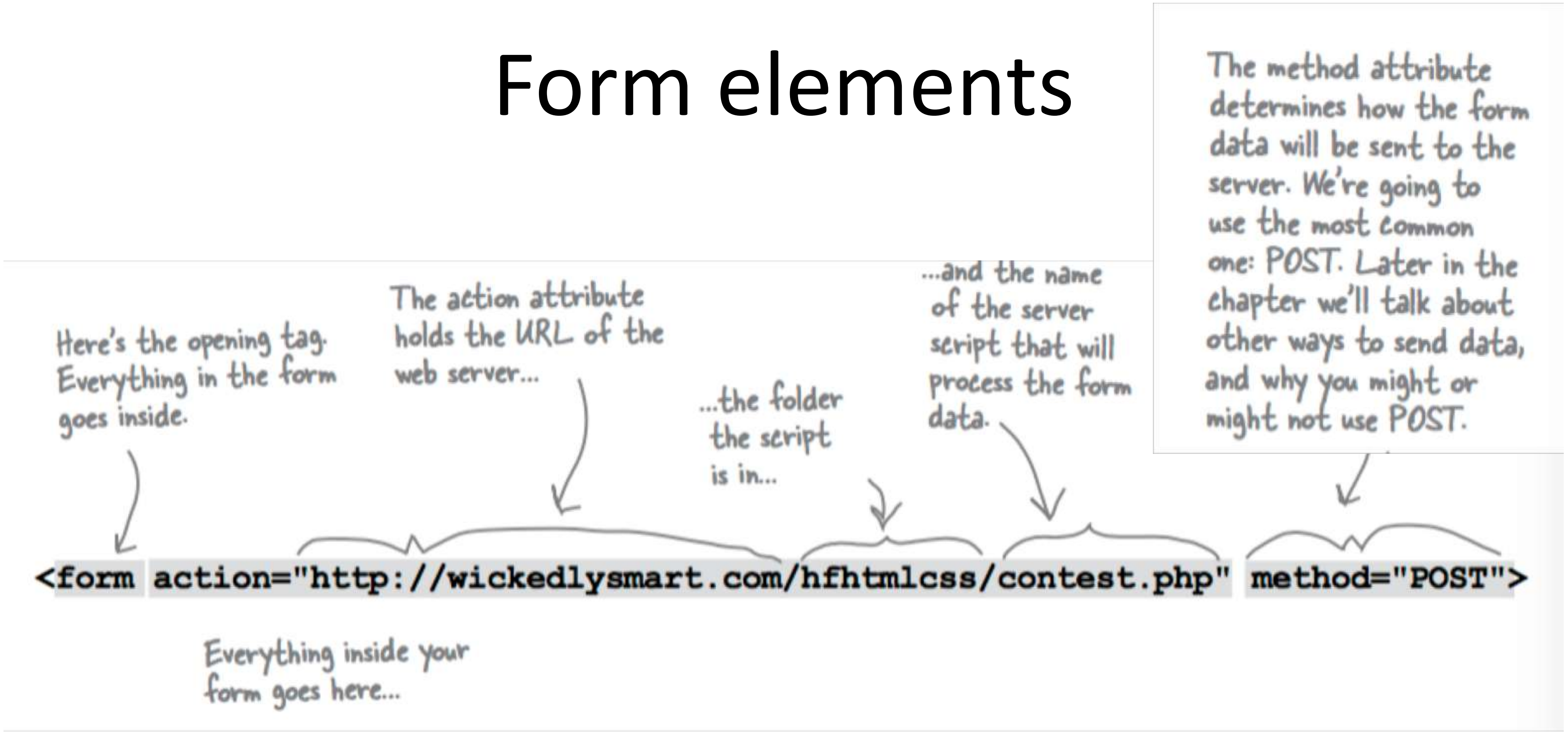

Here's just normal paragraph text in a form.
And here are two text controls for entering a first and last name. In HTML you use the <input> element to create these.
And here's the submit button. (Your button might say "Submit Query" instead.)
A
B
C
D
Enter the Contest
file:///chapter14/contest
Submit

Just type in your name (and click Submit) to ent

First name:

Last name:

Form elements



```
<form action="http://wickedlysmart.com/hfhtmlcss/contest.php" method="POST">
```

Here's the opening tag. Everything in the form goes inside.

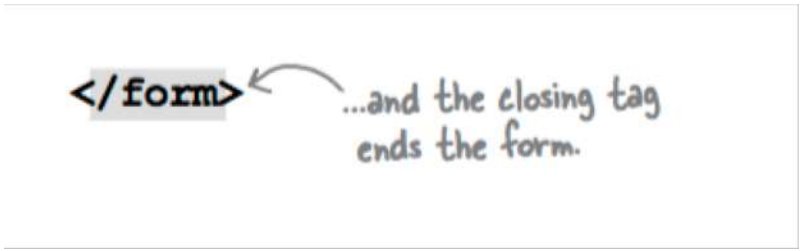
The action attribute holds the URL of the web server...

...the folder the script is in...

...and the name of the server script that will process the form data.

The method attribute determines how the form data will be sent to the server. We're going to use the most common one: POST. Later in the chapter we'll talk about other ways to send data, and why you might or might not use POST.

Everything inside your form goes here...

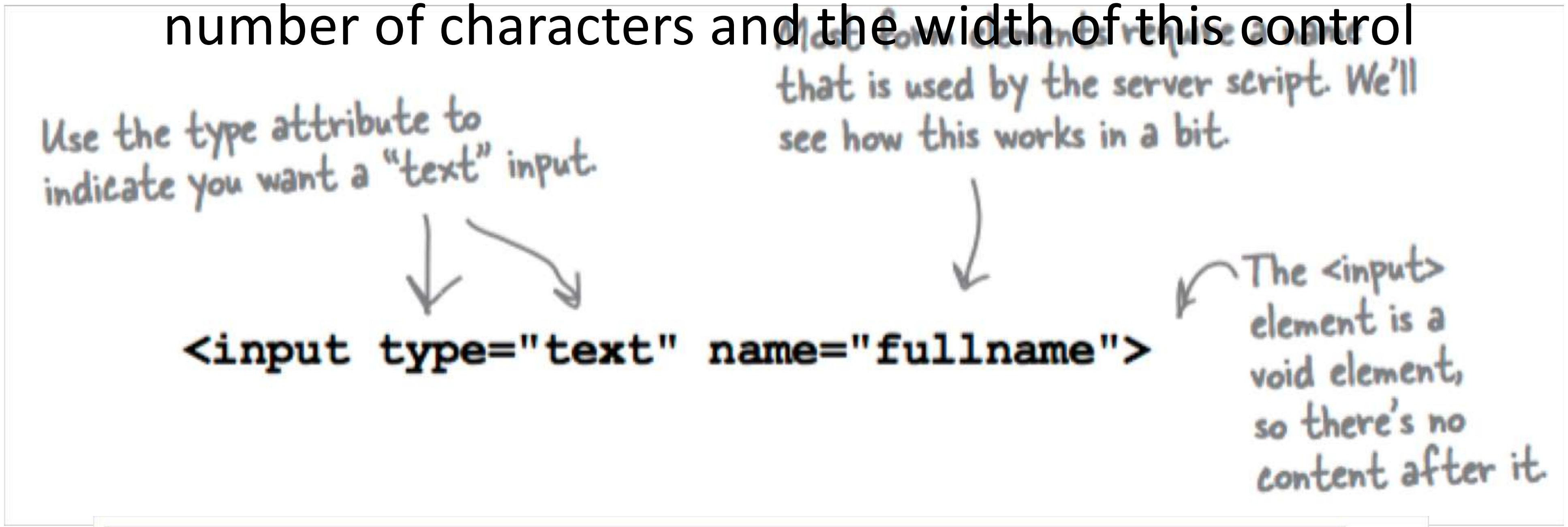


```
</form>
```

...and the closing tag ends the form.

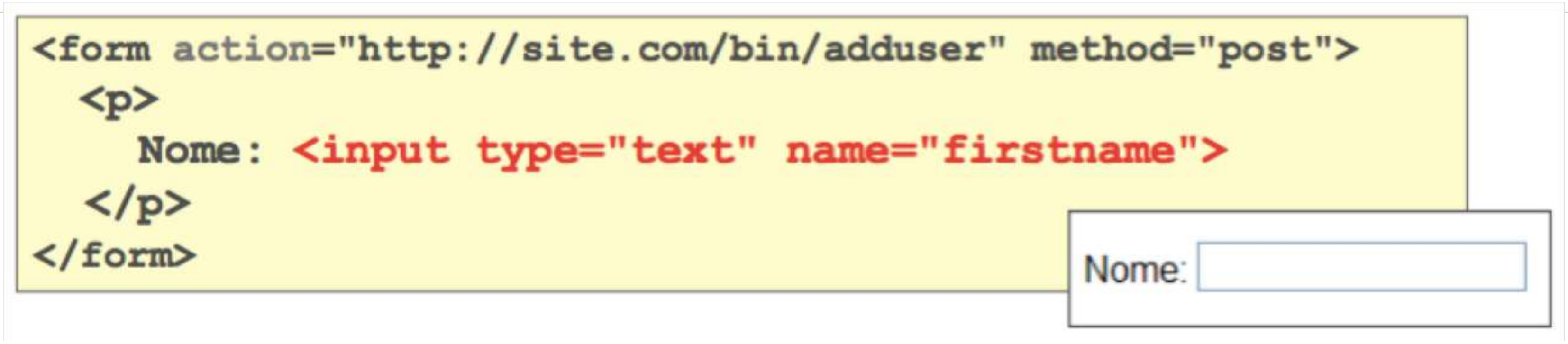
Text input

- The text **<input>** element is for entering one line of text. Optional attributes let you set a maximum number of characters and the width of this control



```
<input type="text" name="fullname">
```

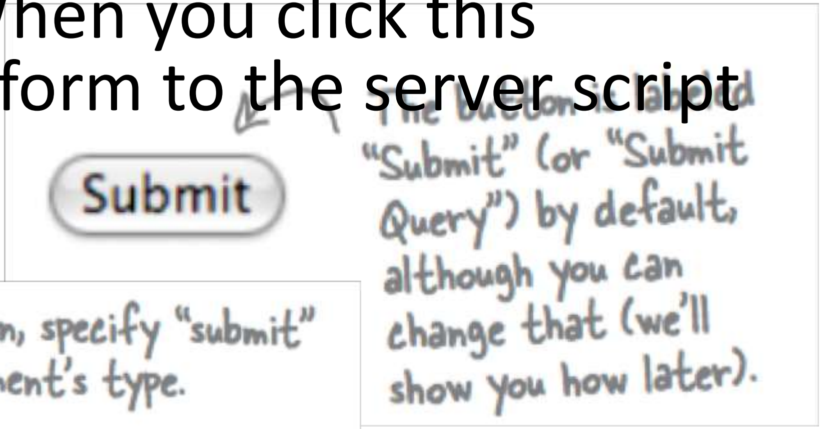
Use the type attribute to indicate you want a "text" input.
that is used by the server script. We'll see how this works in a bit.
The <input> element is a void element, so there's no content after it.

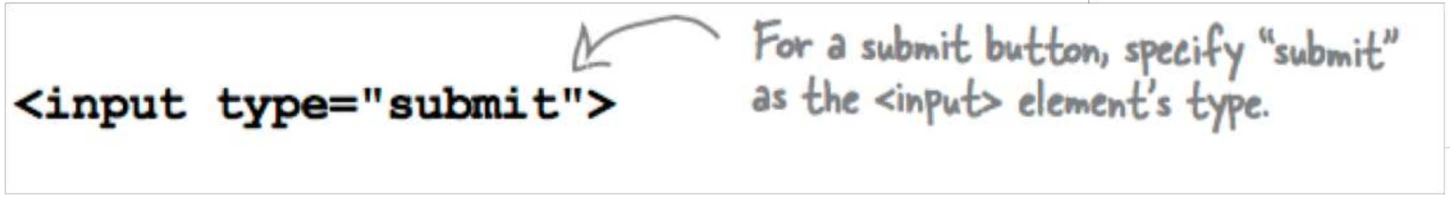


```
<form action="http://site.com/bin/adduser" method="post">
  <p>
    Nome: <input type="text" name="firstname">
  </p>
</form>
```

Submit input

- The submit **<input>** element creates a button that allows you to submit a form. When you click this button, the browser sends the form to the server script for processing

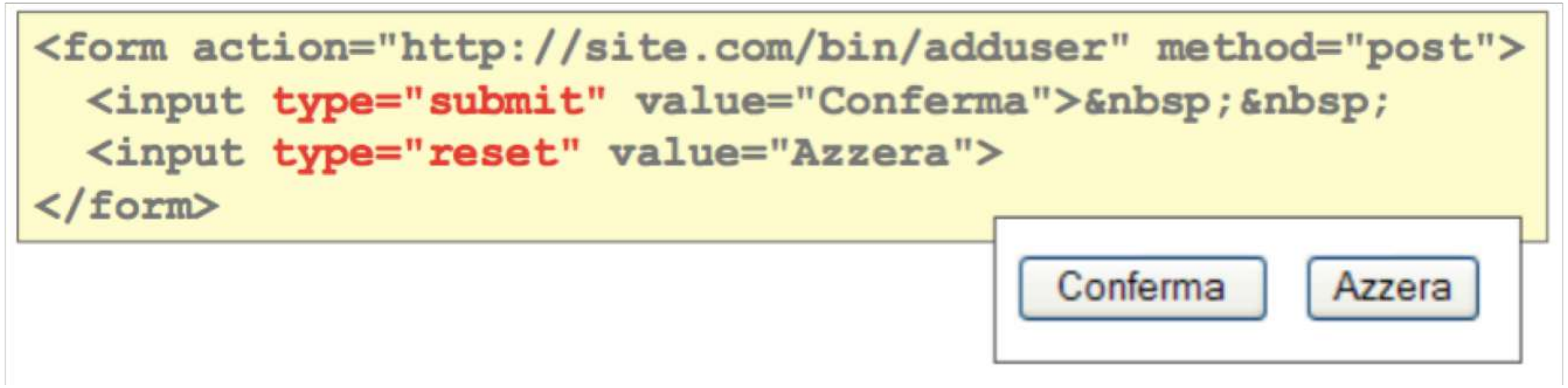

Submit
The button is labeled "Submit" (or "Submit Query") by default, although you can change that (we'll show you how later).



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

For a submit button, specify "submit" as the <input> element's type.

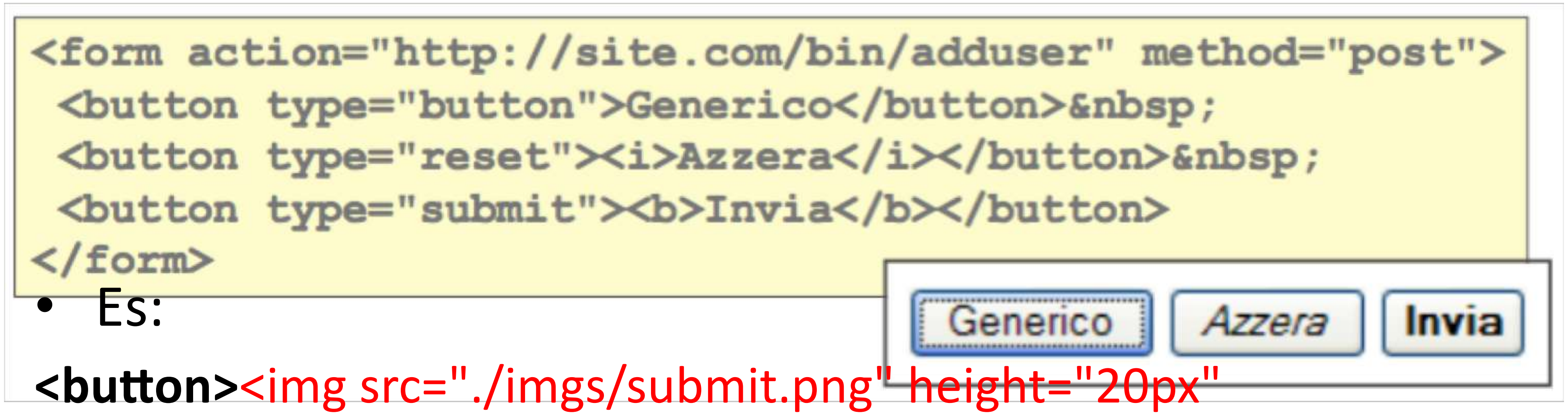
- **type=reset** resets all form values to default values
- **type=button** is a generic action button



```
<form action="http://site.com/bin/adduser" method="post">
  <input type="submit" value="Conferma">&nbsp;&nbsp;
  <input type="reset" value="Azzera">
</form>
```

Button…

- Il tag **<button>** consente di specificare anche codice HTML all'interno del tag: testo formattato ma anche immagini



```
<form action="http://site.com/bin/adduser" method="post">
 <button type="button">Generico</button>&nbsp;
 <button type="reset"><i>Azzera</i></button>&nbsp;
 <button type="submit"><b>Invia</b></button>
</form>
```

- Es:

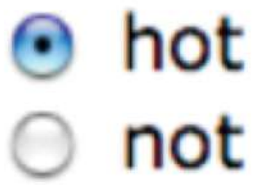
<button><img src="./imgs/submit.png" height="20px" alt="Submit"/>**</button>**

– in alternativa

<input type="image" style="height:20px;vertical-align: text-top;" src="./imgs/submit.png" alt="Submit">

radio input

The radio <input> element creates a single control with several buttons, only one of which can be selected at any time. These are like old-time car radio buttons; you "push" one in, and the rest "pop out."



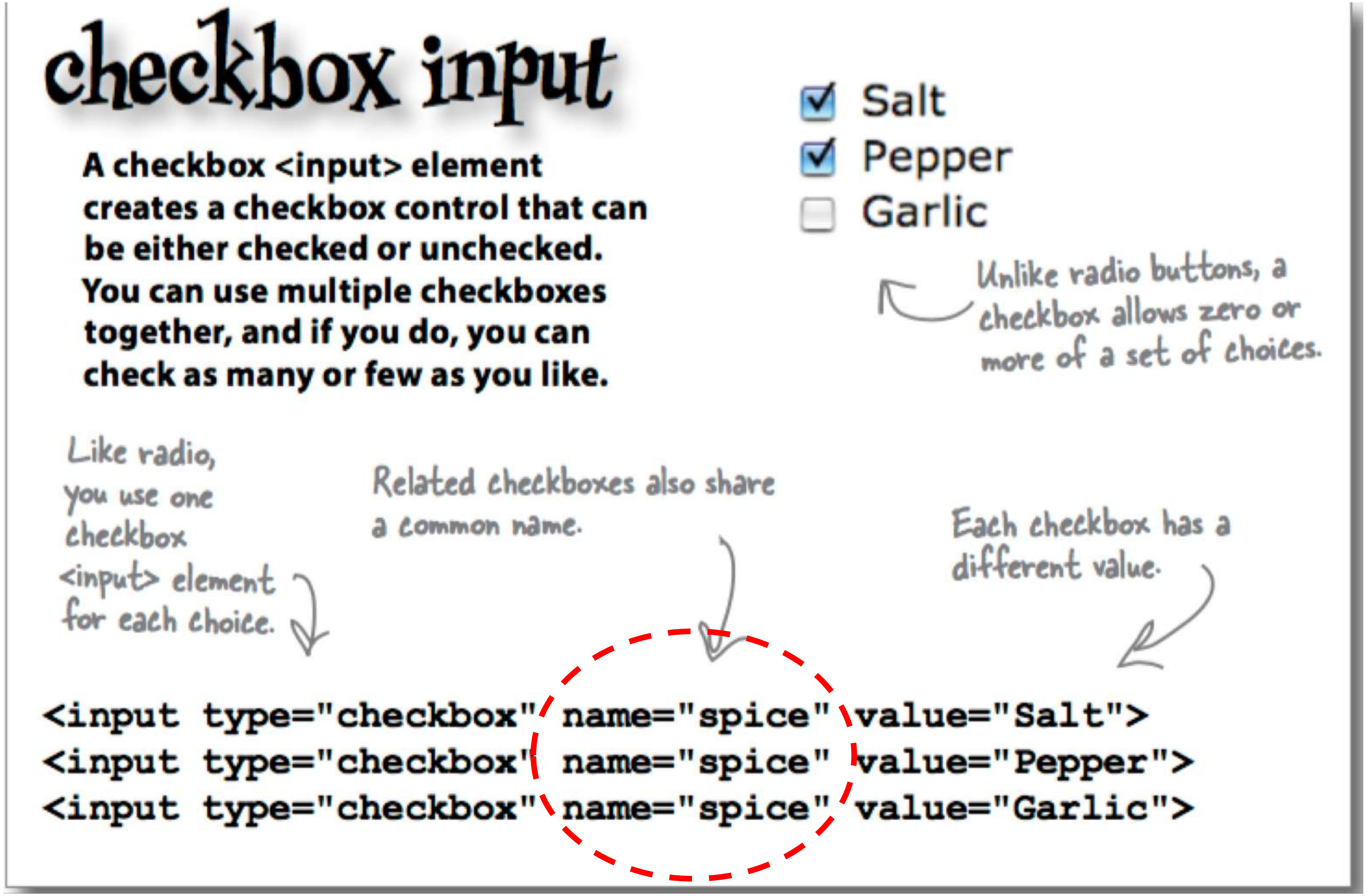
The radio control allows only one of a set of choices.

Use a radio <input> for each choice.

All the radio buttons associated with a given set of choices must have the same name...

...but each choice has a different value.

```
<input type="radio" name="hotornot" value="hot">
<input type="radio" name="hotornot" value="not">
```



checkbox input

A checkbox <input> element creates a checkbox control that can be either checked or unchecked. You can use multiple checkboxes together, and if you do, you can check as many or few as you like.

☑ Salt
☑ Pepper
☐ Garlic

Unlike radio buttons, a checkbox allows zero or more of a set of choices.

Like radio, you use one checkbox <input> element for each choice.

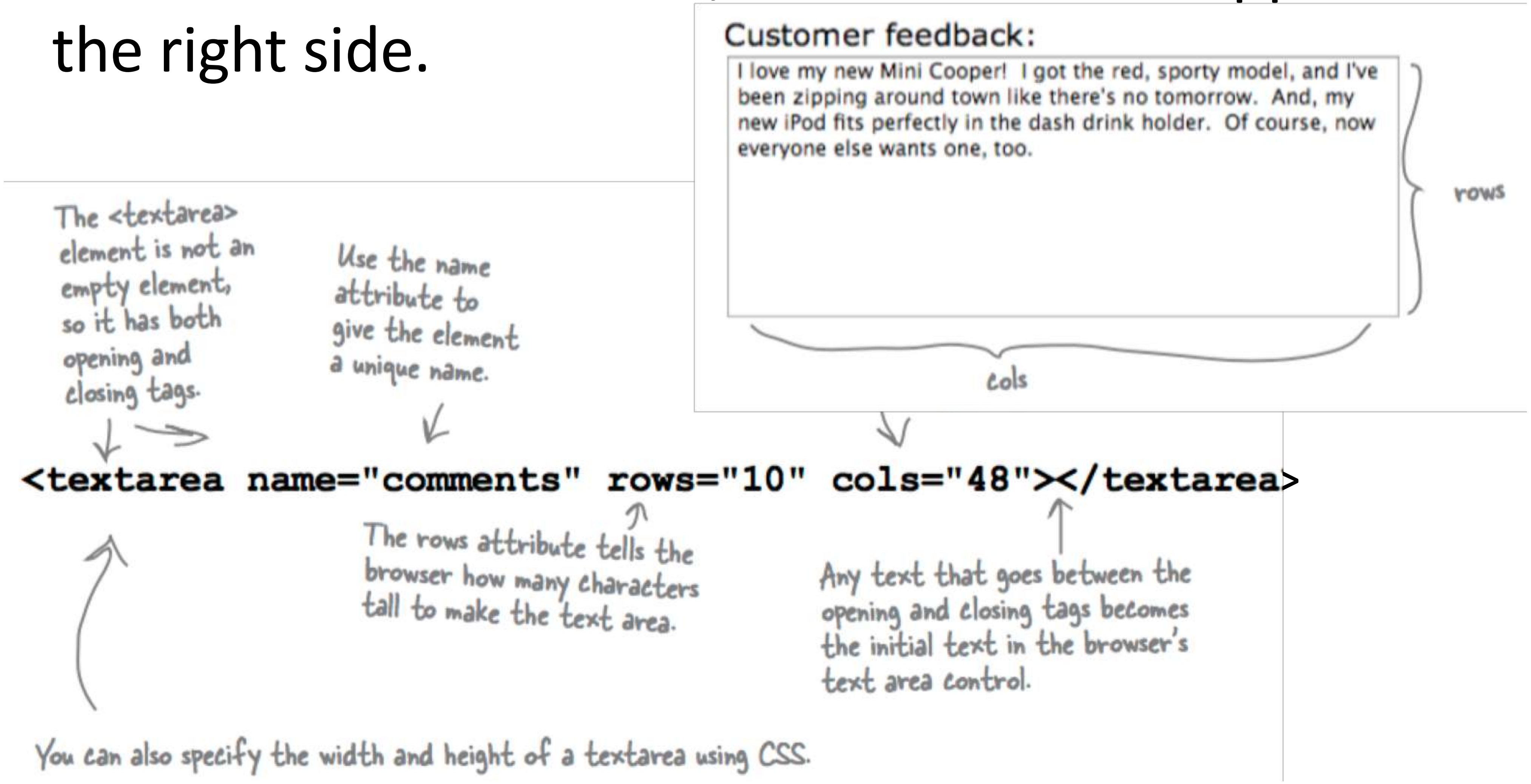
Related checkboxes also share a common name.

Each checkbox has a different value.

```
<input type="checkbox" name="spice" value="Salt">
<input type="checkbox" name="spice" value="Pepper">
<input type="checkbox" name="spice" value="Garlic">
```

textarea

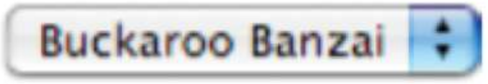
The **<textarea>** element creates a multiline text area that you can type into. If you type more text than will fit into the text area, then a scroll bar appears on the right side.


Customer feedback:
I love my new Mini Cooper! I got the red, sporty model, and I've been zipping around town like there's no tomorrow. And, my new iPod fits perfectly in the dash drink holder. Of course, now everyone else wants one, too.
rows
cols
The <textarea> element is not an empty element, so it has both opening and closing tags.
Use the name attribute to give the element a unique name.
The rows attribute tells the browser how many characters tall to make the text area.
Any text that goes between the opening and closing tags becomes the initial text in the browser's text area control.
You can also specify the width and height of a textarea using CSS.

```
<textarea name="comments" rows="10" cols="48"></textarea>
```

select

The <select> element creates a menu control in the web page. The menu provides a way to choose between a set of choices. The <select> element works in combination with the <option> element below to create a menu.



The select element creates a menu that looks like this (although the look will vary depending on the browser you're using).

The <select> element goes around all the menu options to group them into one menu.

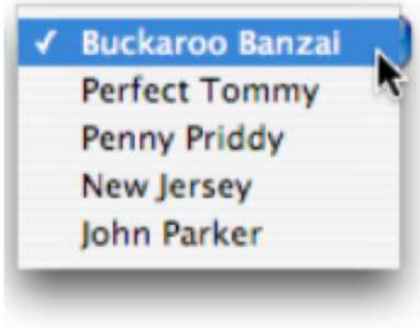
Just like the other form elements, give the select element a unique name using the name attribute.

```
<select name="characters">
    <option value="Buckaroo">Buckaroo Banzai</option>
    <option value="Tommy">Perfect Tommy</option>
    <option value="Penny">Penny Priddy</option>
    <option value="Jersey">New Jersey</option>
    <option value="John">John Parker</option>
</select>
```

option

The <option> element works with the <select> element to create a menu. Use an <option> element for each menu item.

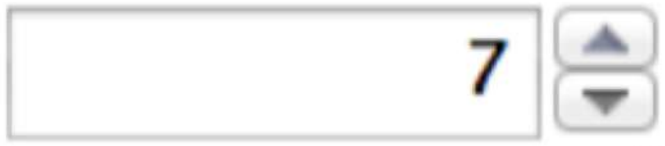
After clicking on the menu, the menu items drop down.



```
<select name="characters">
    <option value="Buckaroo">Buckaroo Banzai</option>
    <option value="Tommy">Perfect Tommy</option>
    <option value="Penny">Penny Priddy</option>
    <option value="Jersey">New Jersey</option>
    <option value="John">John Parker</option>
</select>
```

The content of the <option> element is used for the menu items' description. Each menu option also includes a value representing the menu item.

number input



The number <input> element restricts input to numbers. You can even specify a min and max number that is allowed with optional attributes.

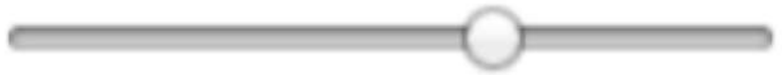
Some browsers show arrows next to the input area you can use to increase or decrease the number.

The "number" type means you're expecting a number only, not text.

Use the max and min attributes to restrict the numbers allowed.

```
<input type="number" min="0" max="20">
```

range input



The range <input> element is similar to number except that it displays a slider instead of an input box.

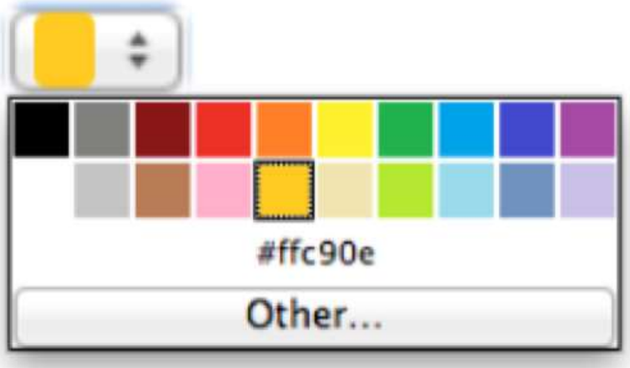
Both number and range have an optional step attribute you can use to specify the number of intervals for the values.

```
<input type="range" min="0" max="20" step="5">
```

color input

Use the color <input> to specify a color. When you click on the control, a color picker pops up that allows you to select a color rather than having to type in the color name or value.

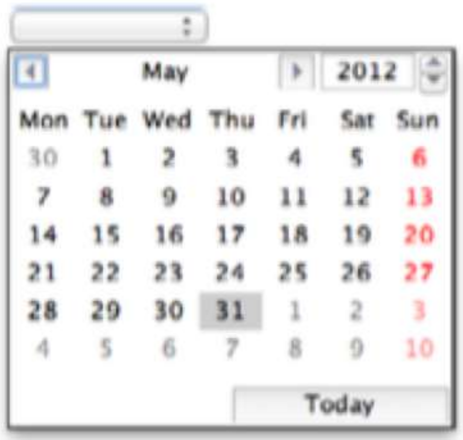
If the color input is not supported by the browser, you'll just get a regular text input instead.



```
<input type="color">
```

date input

Use the date <input> element to specify a date, with a date picker control. The control creates a valid date format string to send to the server script.



```
<input type="date">
```

Like with color, if the date input isn't supported by the browser yet, you'll get a regular text input instead.

email input

The email <input> element is just a text input, but on some mobile browsers, you'll get a custom keyboard for email when you start typing.

```
<input type="email">
```

Email: Buckaroo Banzai

url input

Like email and tel, the url <input> type is just a text input, but causes a custom keyboard to pop up on mobile devices.

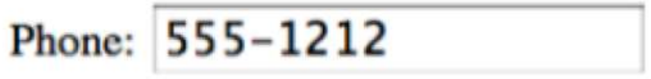
```
<input type="url">
```

URL: http://banzai.com

tel input

The tel <input> element is also just a text input, but like email, causes a custom keyboard to pop up on mobile devices.

```
<input type="tel">
```


Phone: 555-1212

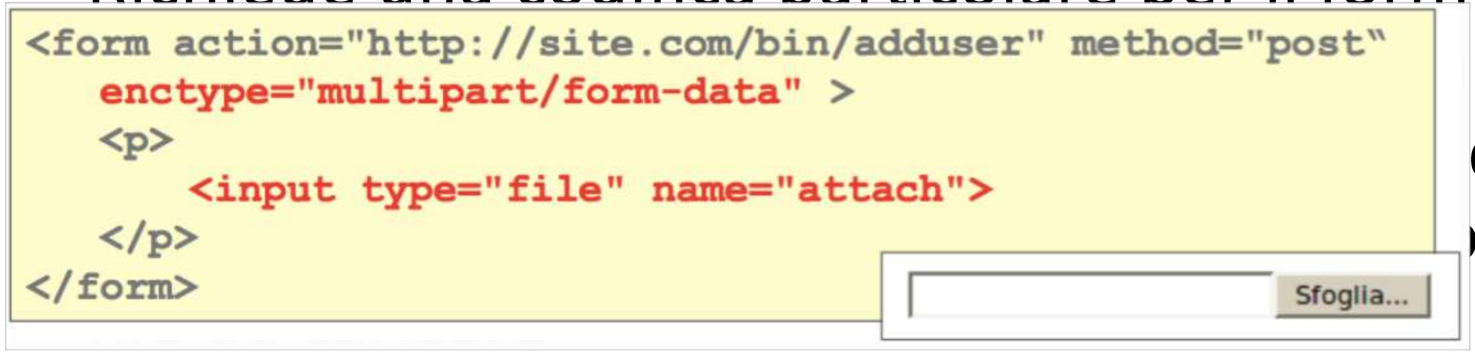
Input file

- Consente di fare l'upload di un file selezionandolo nel file system del client
- Attributi:
  - **name = text** (specifica il nome del controllo)
  - **value = content-type** (lista di MIME types per l'upload)
  - Richiede una codifica particolare per il form

ni

di dati

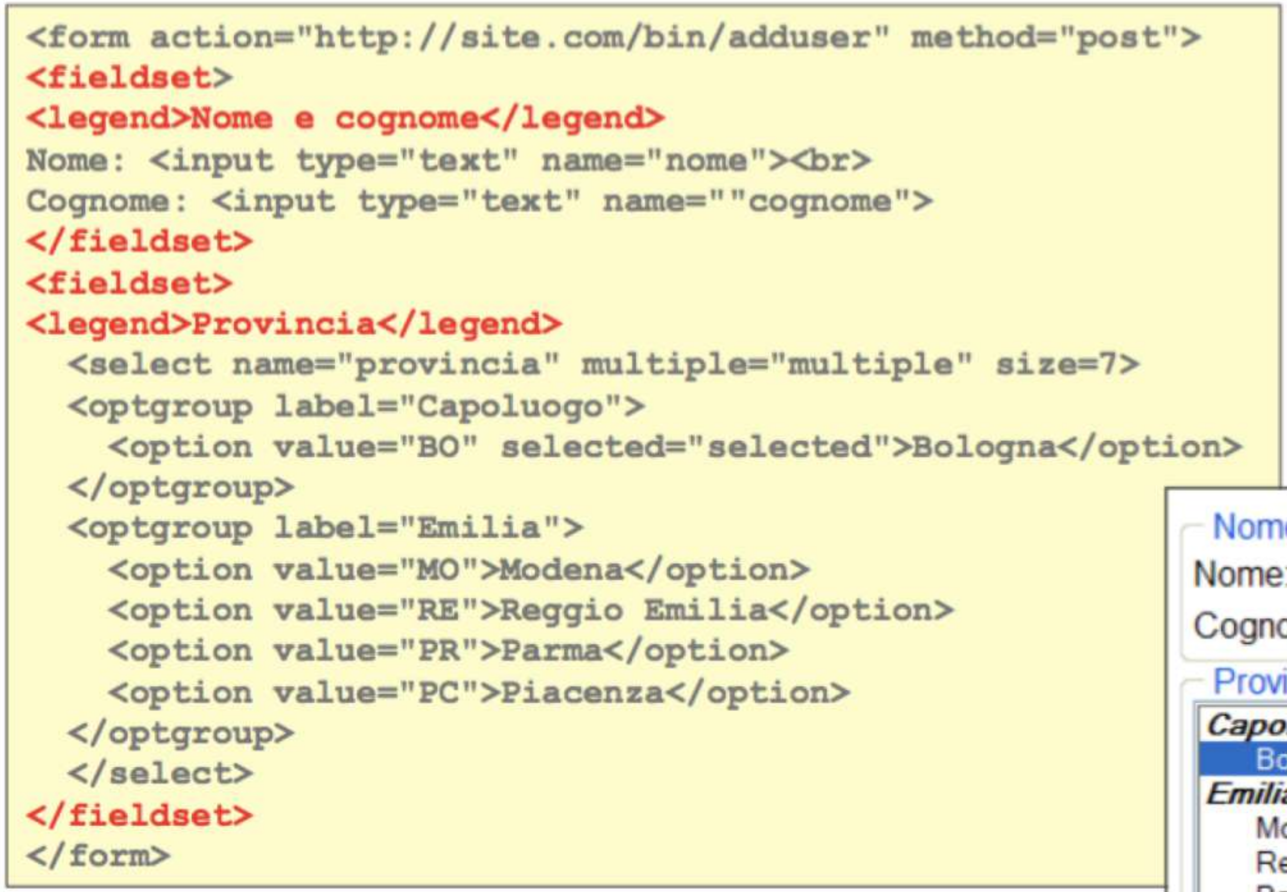
per il



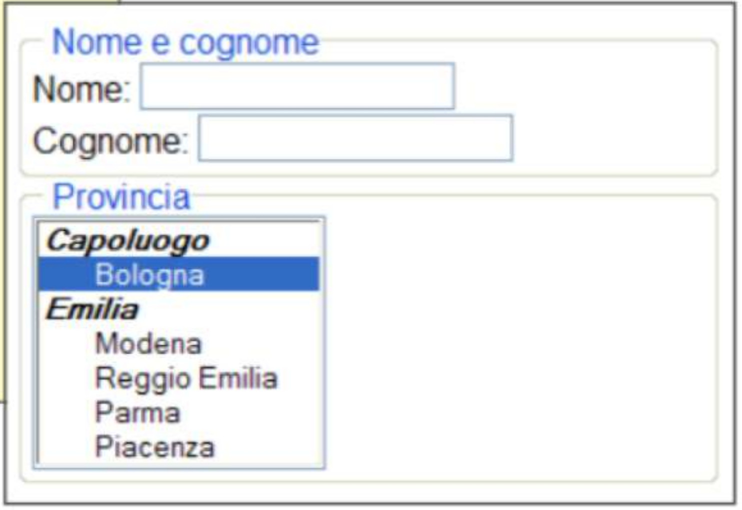
```
<form action="http://site.com/bin/adduser" method="post"
   enctype="multipart/form-data" >
   <p>
      <input type="file" name="attach">
   </p>
</form>
```

Organizzare form complessi

- Con il tag **<fieldset>** si possono creare gruppi di campi a cui è possibile attribuire un nome utilizzando il tag **<legend>**



```
<form action="http://site.com/bin/adduser" method="post">
<fieldset>
<legend>Nome e cognome</legend>
Nome: <input type="text" name="nome"><br>
Cognome: <input type="text" name=""cognome">
</fieldset>
<fieldset>
<legend>Provincia</legend>
  <select name="provincia" multiple="multiple" size=7>
  <optgroup label="Capoluogo">
    <option value="BO" selected="selected">Bologna</option>
  </optgroup>
  <optgroup label="Emilia">
    <option value="MO">Modena</option>
    <option value="RE">Reggio Emilia</option>
    <option value="PR">Parma</option>
    <option value="PC">Piacenza</option>
  </optgroup>
  </select>
</fieldset>
</form>
```



Collegare le etichette ai controlli

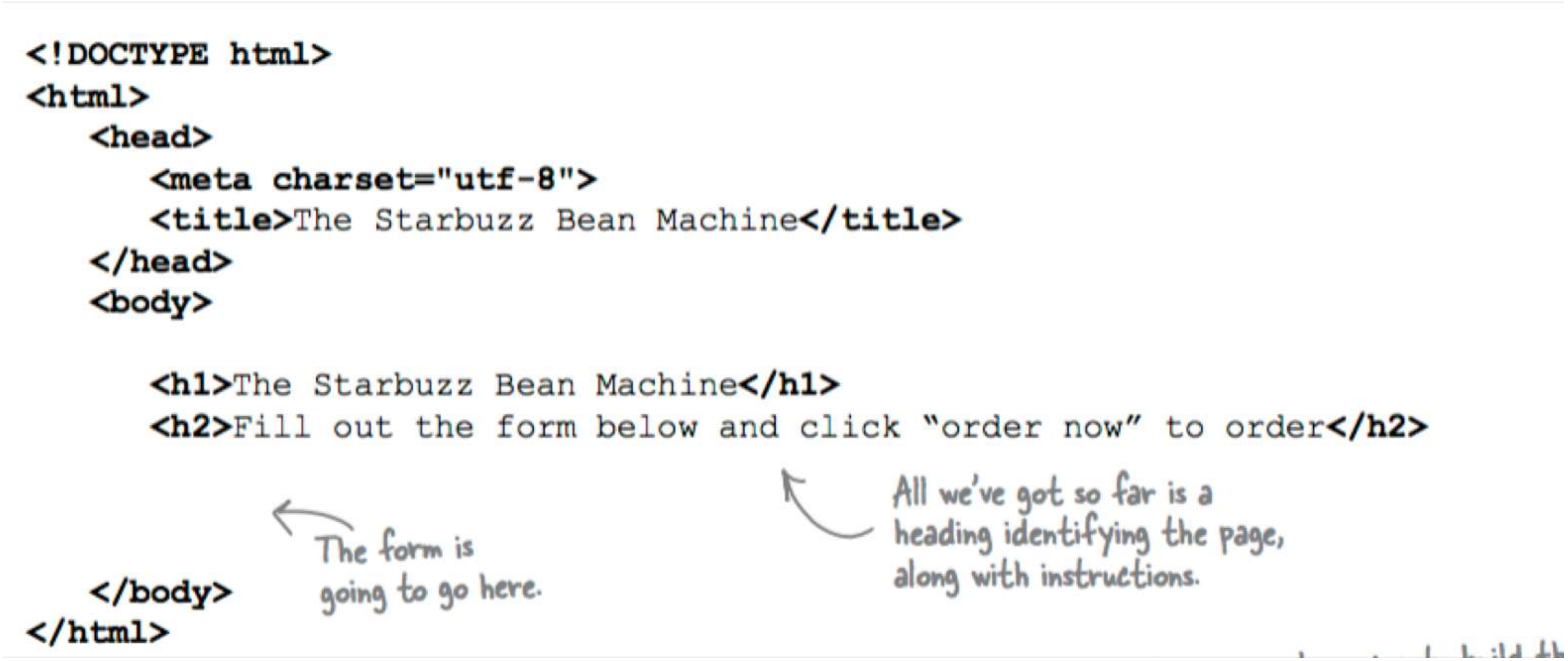
- Il tag **<label>** permette di associare un’etichetta ad un qualunque controllo di un form
  - L’associazione può essere fatta in forma implicita inserendo il controllo nell’elemento label
  - In forma esplicita tramite l’attributo **for** che deve corrispondere all’attributo **id** del controllo

```
<form action=“…”>
 <label>Nome: <input type="text“ id=“nome”></label><br>
 <label>Cognome: <input type="text" id=“cognome"></label><br>
</form>
```

```
<form action=“…”>
 <label for=“nome">Nome: </label>
 <input type="text" id=“nome"><br>
 <label for=“cognome">Cognome: </label>
 <input type="text" id=“cognome"><br>
</form>
```

Exercise

- Look at “L05 code/starbuzz” folder, and



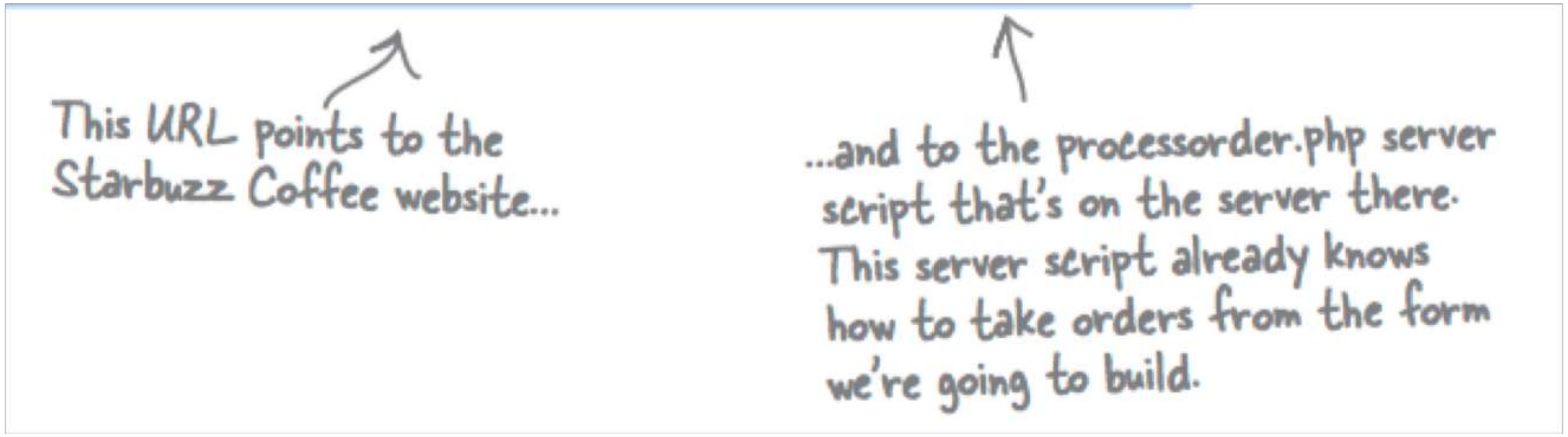
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>The Starbuzz Bean Machine</title>
   </head>
   <body>

      <h1>The Starbuzz Bean Machine</h1>
      <h2>Fill out the form below and click “order now” to order</h2>



   </body>
</html>
```

The form is going to go here.

All we've got so far is a heading identifying the page, along with instructions.

The URL of the server script

- **http://starbuzzcoffee.com/processorder.php**


This URL points to the Starbuzz Coffee website...

...and to the processorder.php server script that's on the server there. This server script already knows how to take orders from the form we're going to build.

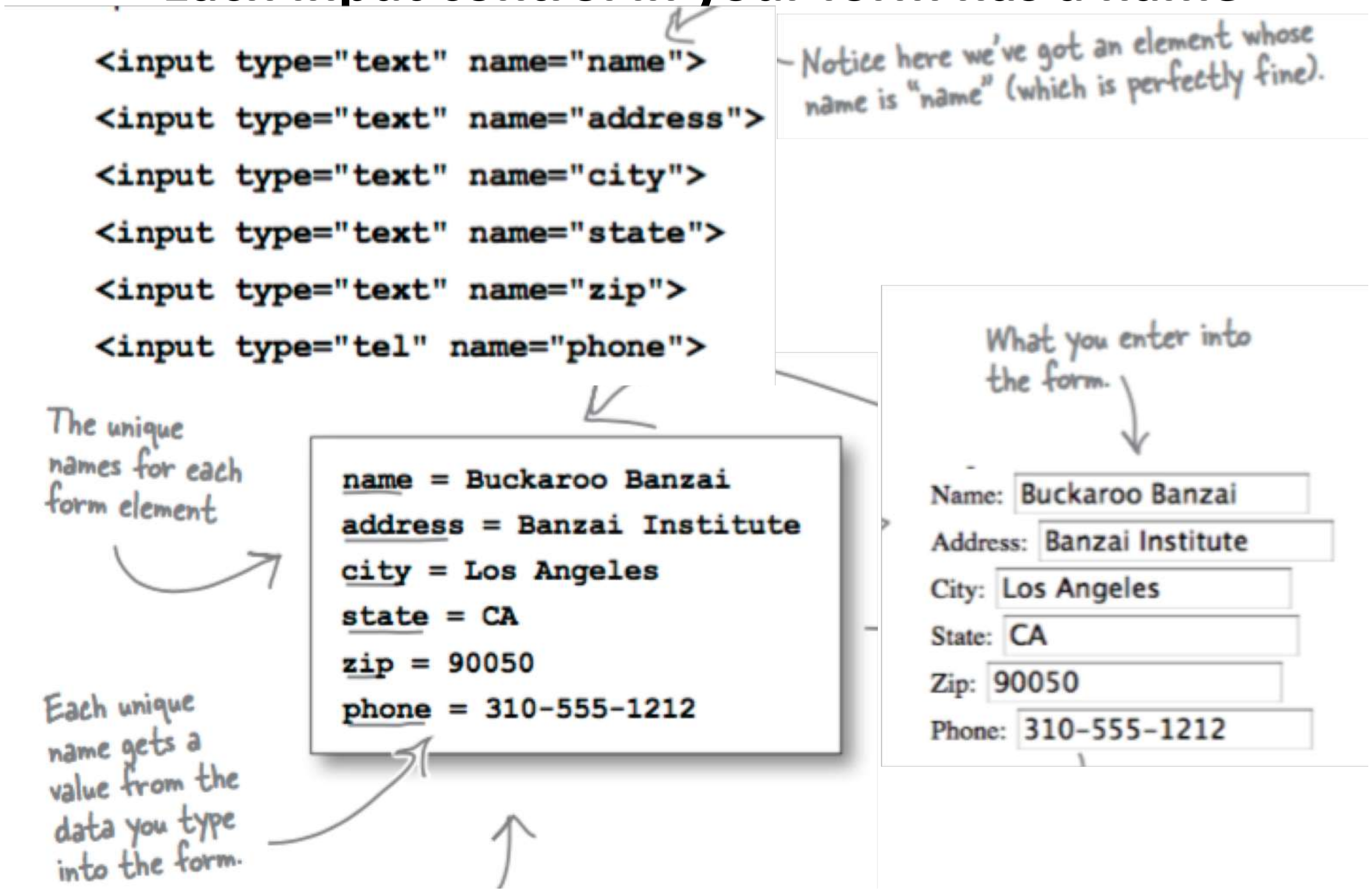
Adding the form element

- The URL of the server script that will process your form has tc be plugged into the action attribute of your <form> element

```
<body>
     <h1>The Starbuzz Bean Machine</h1>
     <h2>Fill out the form below and click “order now” to
order</h2>
     <form
action=“http://starbuzzcoffee.com/processorder.php”
method="POST">
     …
      </form>
</body>
```

Form element names

- **Each input control in your form has a name**



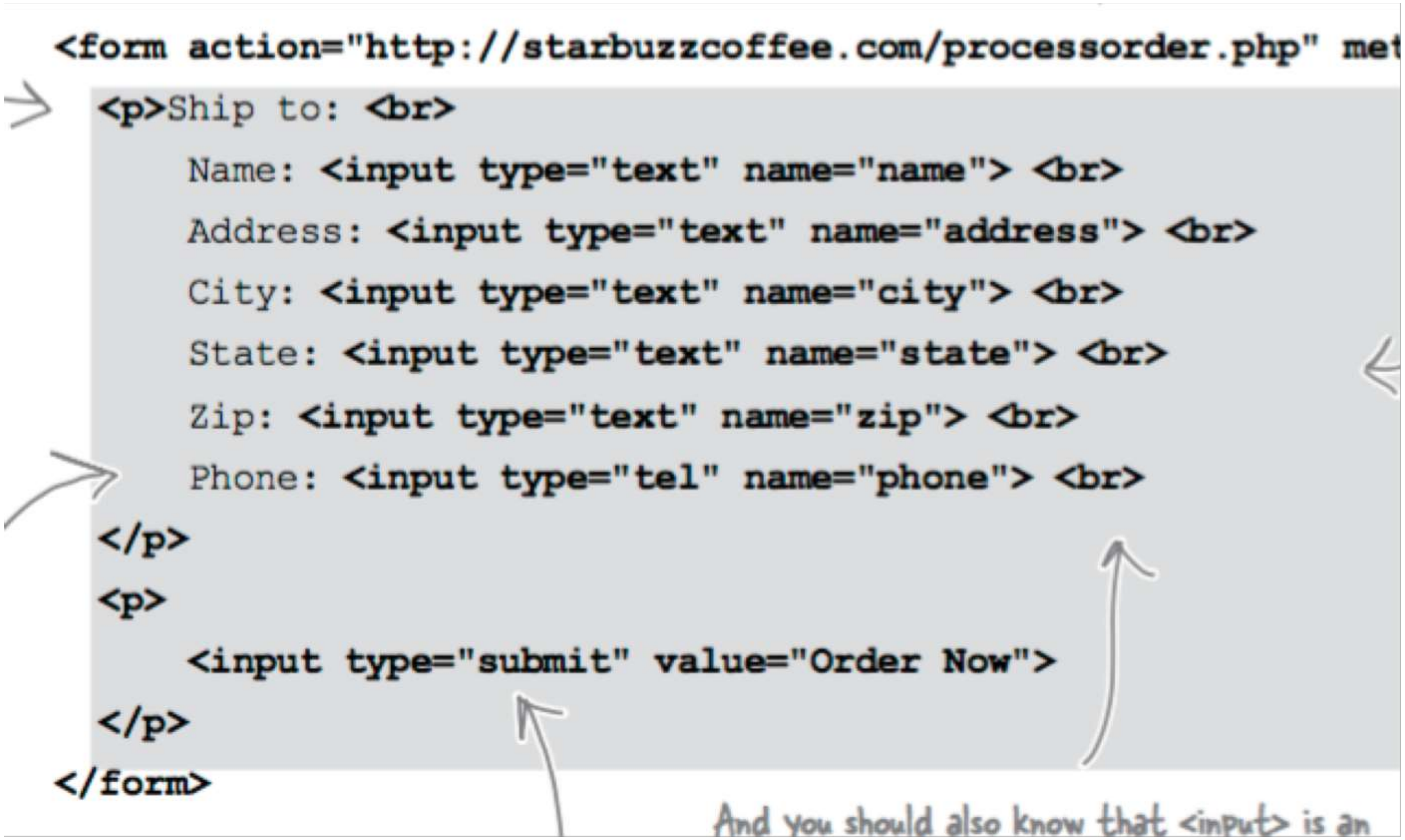
```
<input type="text" name="name">
<input type="text" name="address">
<input type="text" name="city">
<input type="text" name="state">
<input type="text" name="zip">
<input type="tel" name="phone">
```

```
name = Buckaroo Banzai
address = Banzai Institute
city = Los Angeles
state = CA
zip = 90050
phone = 310-555-1212
```

Name:	Buckaroo Banzai
Address:	Banzai Institute
City:	Los Angeles
State:	CA
Zip:	90050
Phone:	310-555-1212

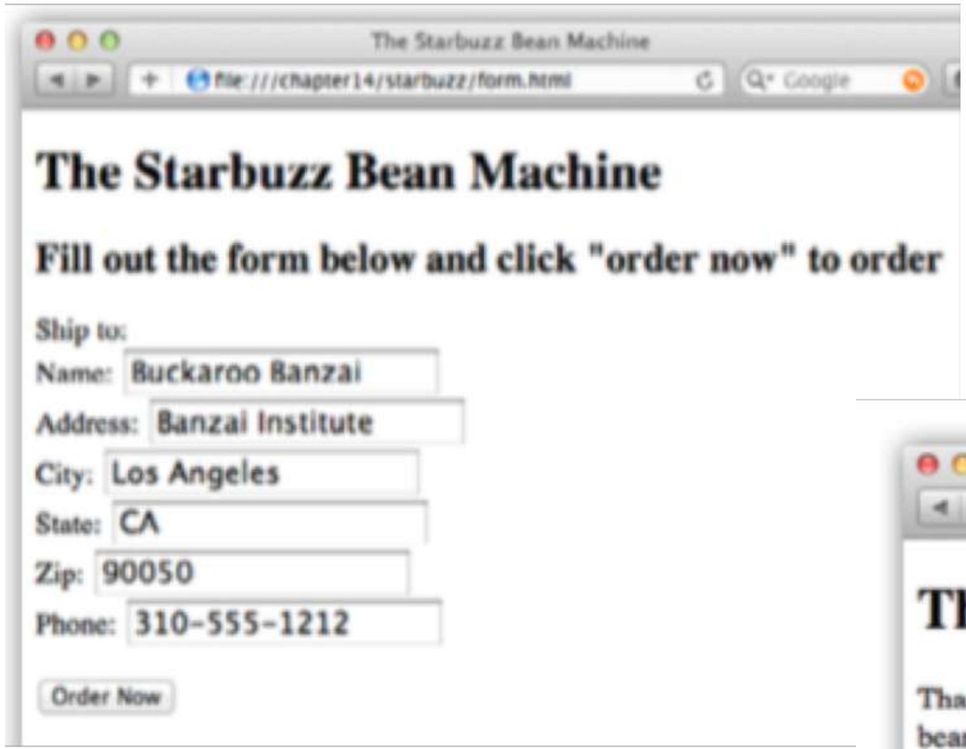
Notice here we've got an element whose name is "name" (which is perfectly fine).
What you enter into the form.
The unique names for each form element
Each unique name gets a value from the data you type into the form.

Insert the inputs in the HTML



```
<form action="http://starbuzzcoffee.com/processorder.php" met
  <p>Ship to: <br>
      Name: <input type="text" name="name"> <br>
      Address: <input type="text" name="address"> <br>
      City: <input type="text" name="city"> <br>
      State: <input type="text" name="state"> <br>
      Zip: <input type="text" name="zip"> <br>
      Phone: <input type="tel" name="phone"> <br>
  </p>
  <p>
      <input type="submit" value="Order Now">
  </p>
</form>
```

And you should also know that <input> is an



The Starbuzz Bean Machine

file:///chapter14/starbuzz/form.html

The Starbuzz Bean Machine

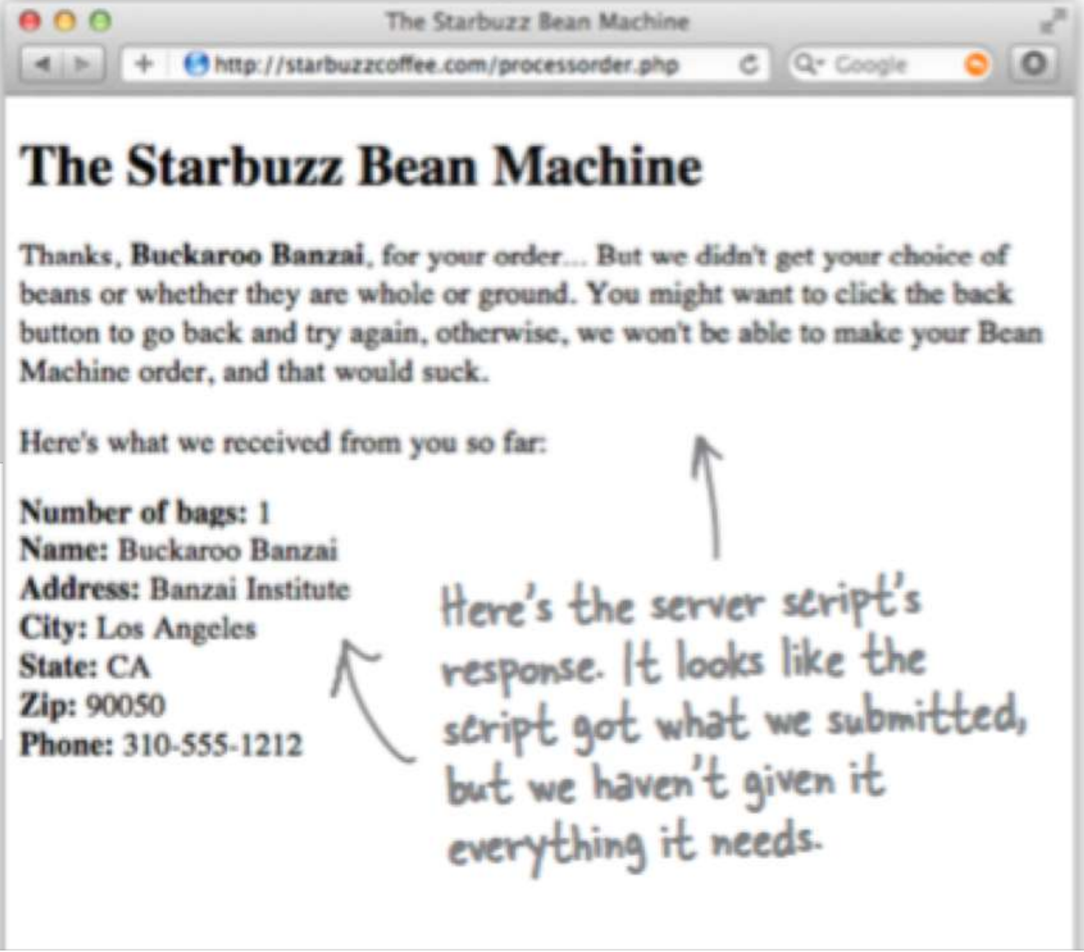
Fill out the form below and click "order now" to order

Ship to:

Name:	Buckaroo Banzai
Address:	Banzai Institute
City:	Los Angeles
State:	CA
Zip:	90050
Phone:	310-555-1212

Here's the form.

the address bar?



The Starbuzz Bean Machine

http://starbuzzcoffee.com/processorder.php

The Starbuzz Bean Machine

Thanks, **Buckaroo Banzai**, for your order... But we didn't get your choice of beans or whether they are whole or ground. You might want to click the back button to go back and try again, otherwise, we won't be able to make your Bean Machine order, and that would suck.

Here's what we received from you so far:

Number of bags: 1
Name: Buckaroo Banzai
Address: Banzai Institute
City: Los Angeles
State: CA
Zip: 90050
Phone: 310-555-1212

And here's the response after submitting the form.

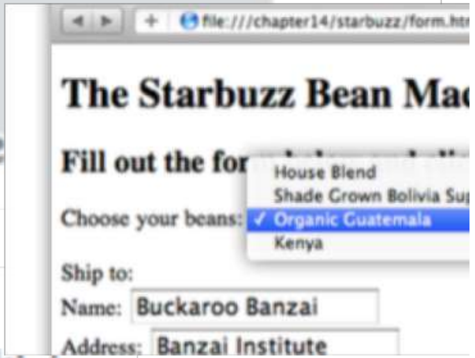
Here's the server script's response. It looks like the script got what we submitted, but we haven't given it everything it needs.

Adding the select element

```
<form action="http://starbuzzcoffee.com/processorder.php" method="post">

  <p>
    Choose your beans:
    <select name="beans">
      <option value="House Blend">House Blend</option>
      <option value="Bolivia">Shade Grown Bolivia Supremo</option>
      <option value="Guatemala">Organic Guatemala</option>
      <option value="Kenya">Kenya</option>
    </select>
  </p>
```

Here's our brand-new <select> element. It gets a unique name too.

Inside, we put each <option> element, one per choice of coffee.


file:///chapter14/starbuzz/form.htm

The Starbuzz Bean Mac

Fill out the for

Choose your beans:

Ship to:

Name: Buckaroo Banzai

Address: Banzai Institute

Each option has a value.

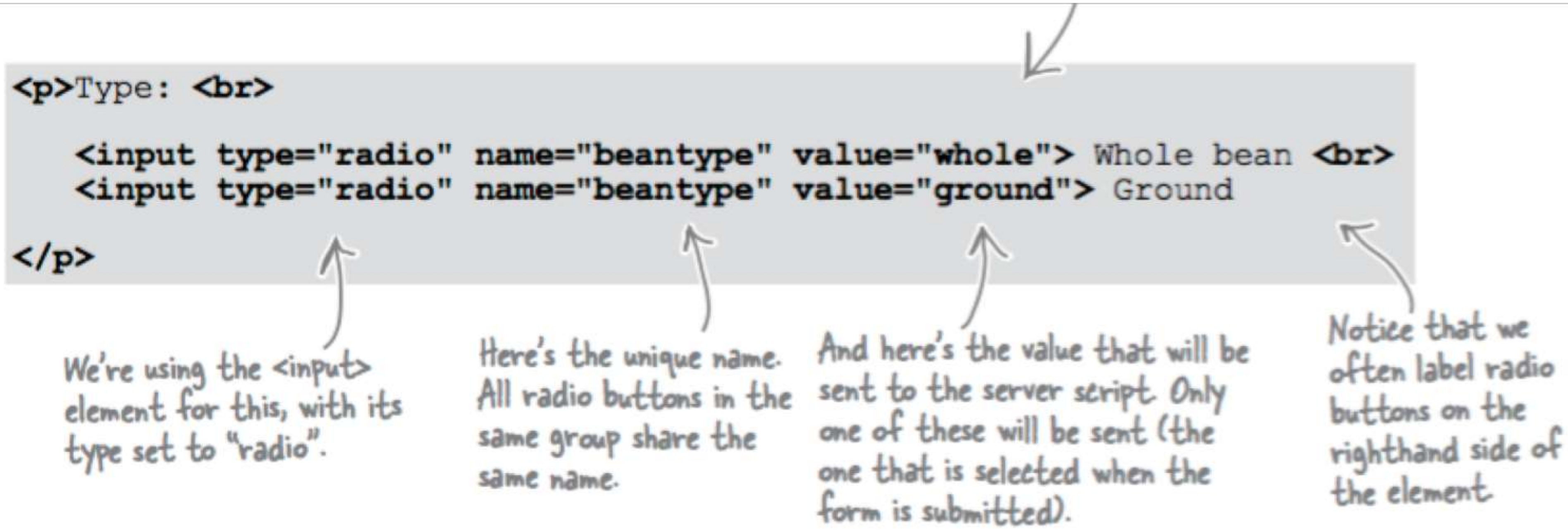
The content of the element is used as the label in the drop-down menu.

```
<option value="Guatemala">Organic Guatemala</option>
```

When the browser packages up the names and values of the form elements, it uses the name of the <select> element along with the value of the chosen option.

In this case, the browser would send the server beans = "Guatemala".

Radio buttons



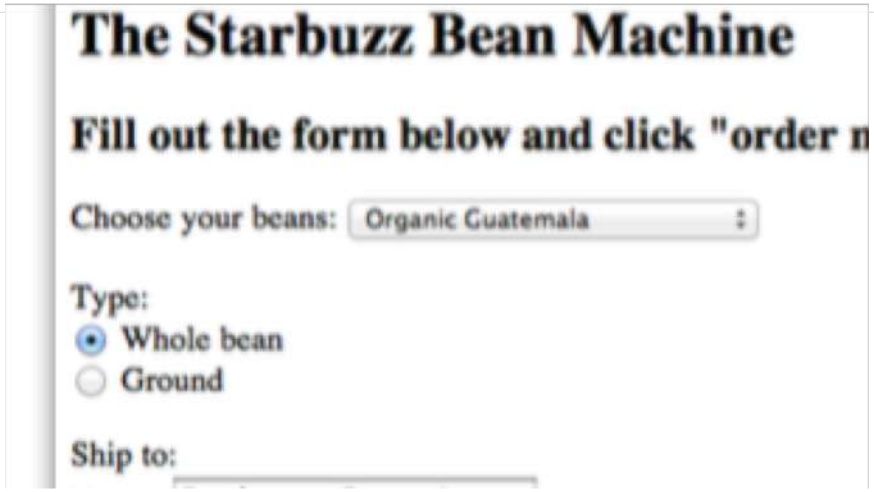
```
<p>Type: <br>

   <input type="radio" name="beantype" value="whole"> Whole bean <br>
   <input type="radio" name="beantype" value="ground"> Ground

</p>
```

We're using the <input> element for this, with its type set to "radio".

Here's the unique name. All radio buttons in the same group share the same name.

And here's the value that will be sent to the server script. Only one of these will be sent (the one that is selected when the form is submitted).

Notice that we often label radio buttons on the righthand side of the element.



The Starbuzz Bean Machine

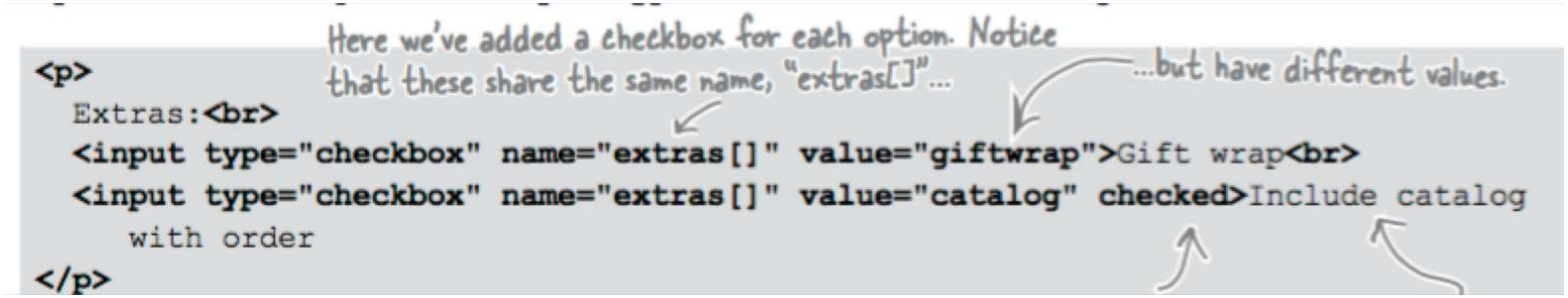
Fill out the form below and click "order n

Choose your beans: Organic Guatemala

Type:
[✓ Whole bean]
Ground

Ship to:

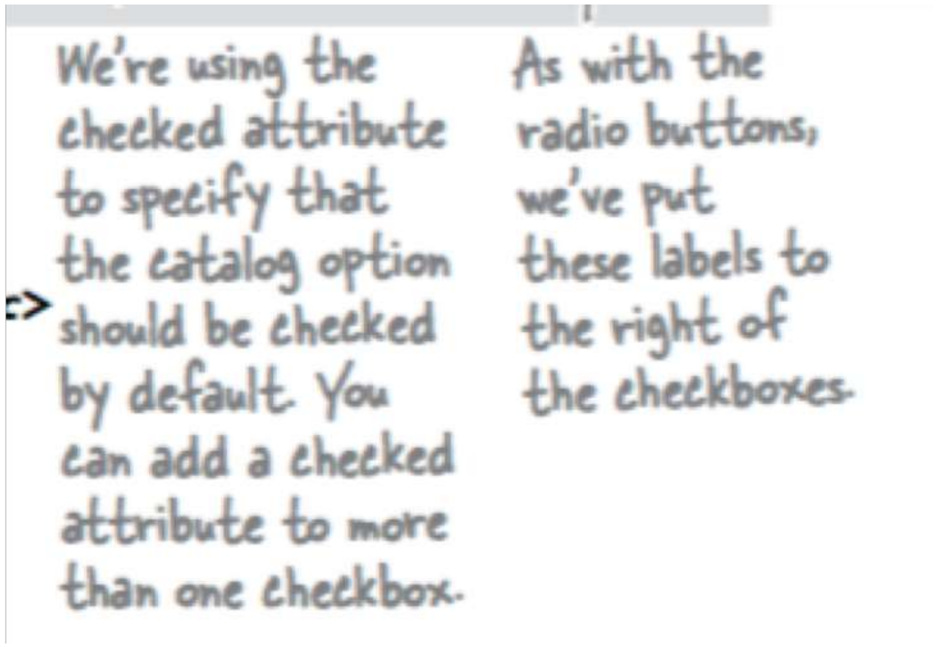
Check boxes



Here we've added a checkbox for each option. Notice that these share the same name, "extras[]"...

...but have different values.

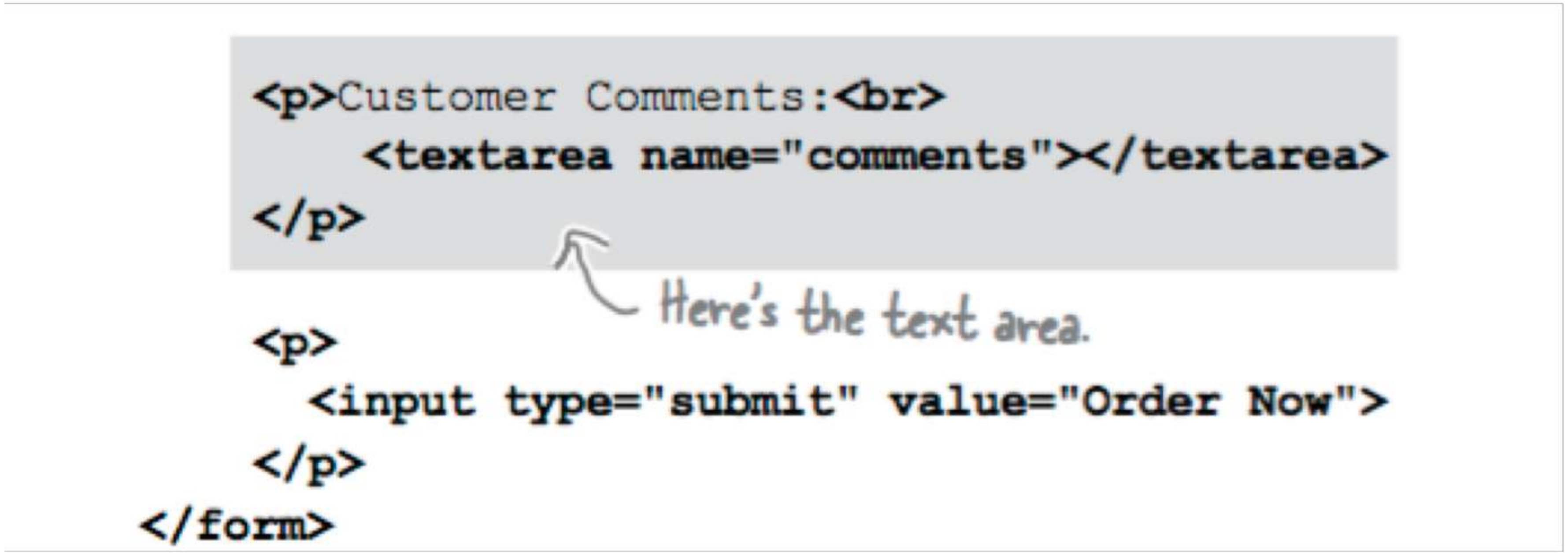
```
<p>
  Extras:<br>
  <input type="checkbox" name="extras[]" value="giftwrap">Gift wrap<br>
  <input type="checkbox" name="extras[]" value="catalog" checked>Include catalog
     with order
</p>
```



We're using the checked attribute to specify that the catalog option should be checked by default. You can add a checked attribute to more than one checkbox.

As with the radio buttons, we've put these labels to the right of the checkboxes.

Text Area

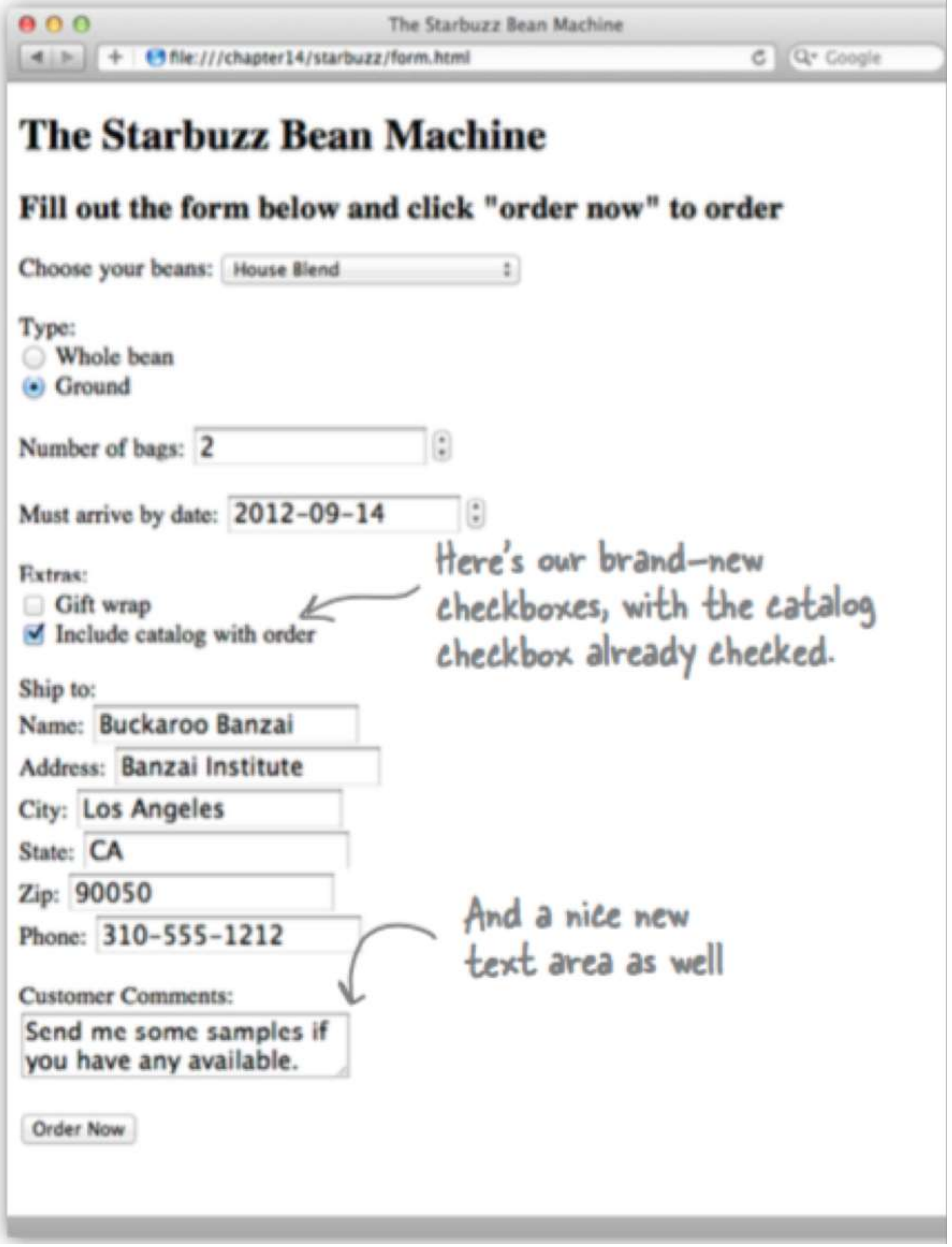


```
<p>Customer Comments:<br>
    <textarea name="comments"></textarea>
</p>

<p>
  <input type="submit" value="Order Now">
</p>
</form>
```

Number and data types

```
<p>
    Number of bags:
    <input type="number" name="bags"
min="1”max="10">
</p>
<p>
    Must arrive by date:
    <input type="date" name="date">
</p>
```

Not supported by all the browsers!



The Starbuzz Bean Machine

file:///chapter14/starbuzz/form.html

The Starbuzz Bean Machine

Fill out the form below and click "order now" to order

Choose your beans: House Blend

Type:
- ( ) Whole bean
- (•) Ground

Number of bags: 2

Must arrive by date: 2012-09-14

Extras:
- [ ] Gift wrap
- [✓] Include catalog with order

Ship to:
Name: Buckaroo Banzai
Address: Banzai Institute
City: Los Angeles
State: CA
Zip: 90050
Phone: 310-555-1212

Customer Comments:
Send me some samples if you have any available.

Order Now

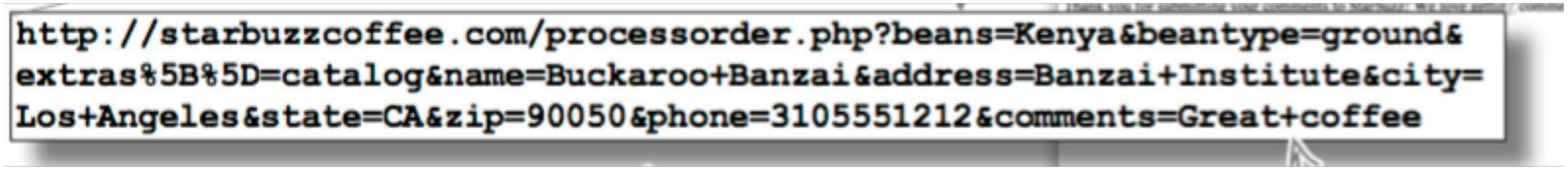
Here's our brand-new checkboxes, with the catalog checkbox already checked.
And a nice new text area as well

get

- Open up your “form.html” file and make the following small change:

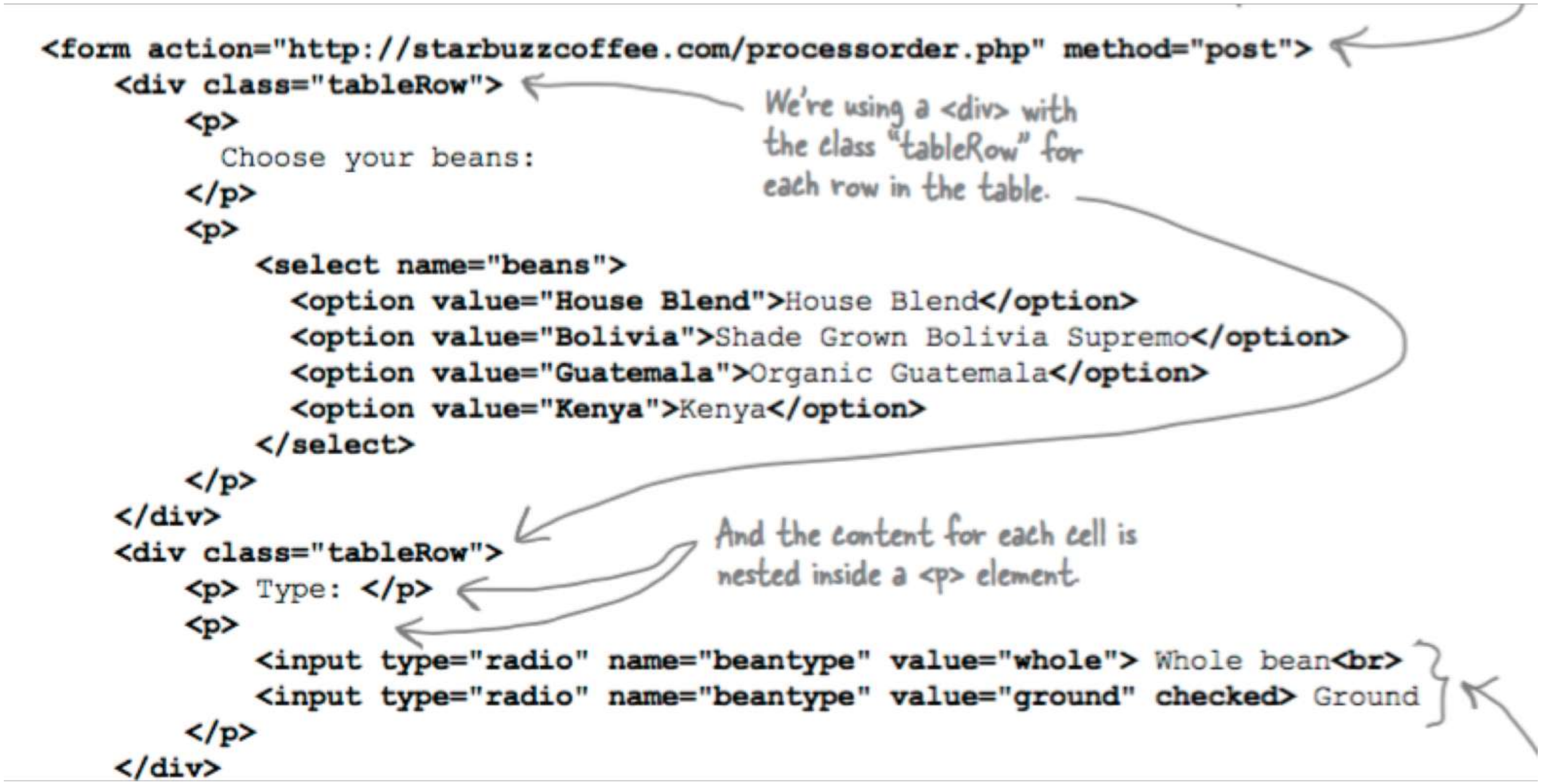
<form action="http://starbuzzcoffee.com/processorder.php" method="**GET**">

- This is the URL of your browser:



```
http://starbuzzcoffee.com/processorder.php?beans=Kenya&beantype=ground&
extras%5B%5D=catalog&name=Buckaroo+Banzai&address=Banzai+Institute&city=
Los+Angeles&state=CA&zip=90050&phone=3105551212&comments=Great+coffee
```

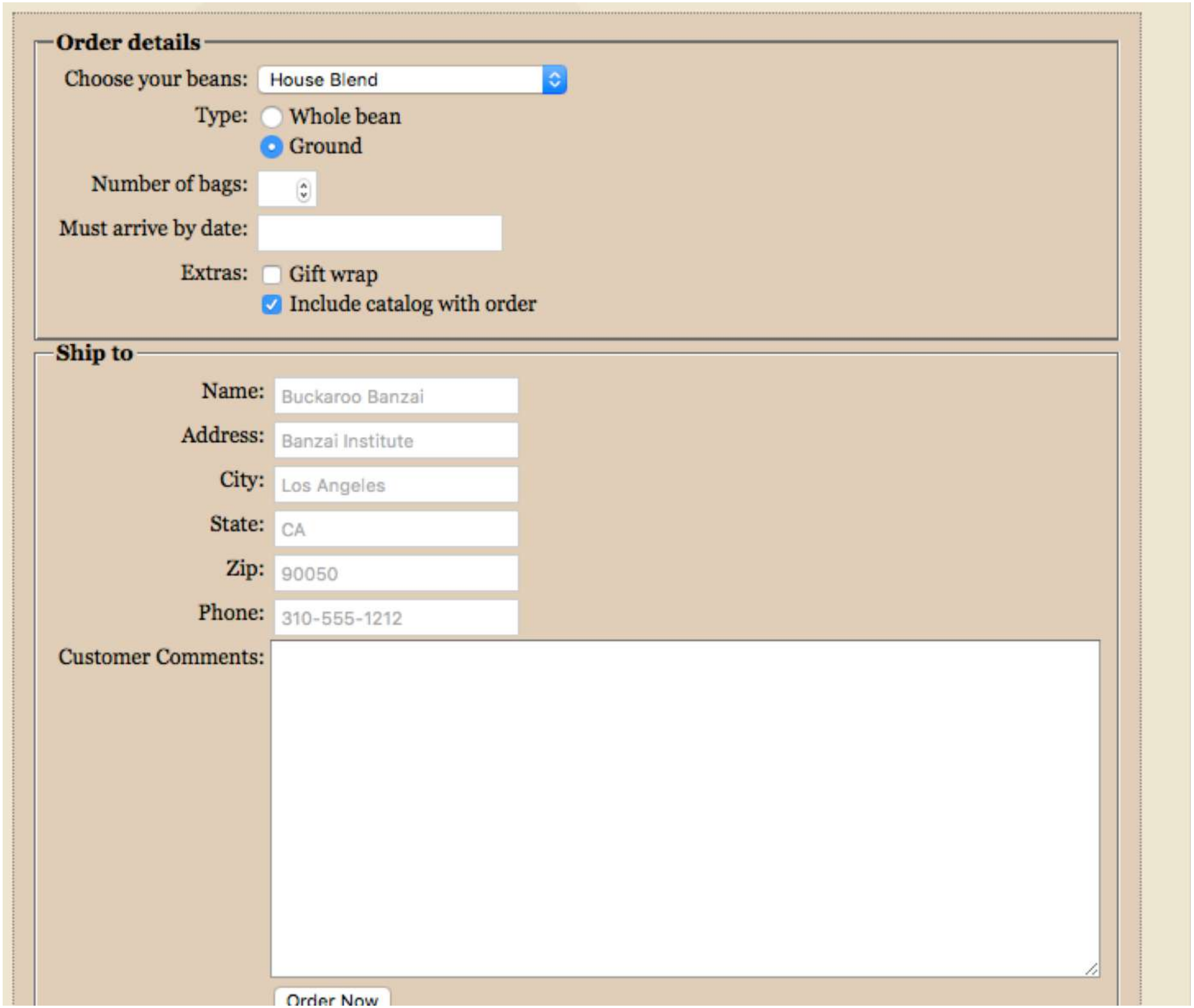
Getting the form elements into a Table



```
<form action="http://starbuzzcoffee.com/processorder.php" method="post">
    <div class="tableRow">
        <p>
          Choose your beans:
        </p>
        <p>
            <select name="beans">
              <option value="House Blend">House Blend</option>
              <option value="Bolivia">Shade Grown Bolivia Supremo</option>
              <option value="Guatemala">Organic Guatemala</option>
              <option value="Kenya">Kenya</option>
            </select>
        </p>
    </div>
    <div class="tableRow">
        <p> Type: </p>
        <p>
            <input type="radio" name="beantype" value="whole"> Whole bean<br>
            <input type="radio" name="beantype" value="ground" checked> Ground
        </p>
    </div>
```

We're using a <div> with the class "tableRow" for each row in the table.

And the content for each cell is nested inside a <p> element.



Order details

Choose your beans:	House Blend
Type:	○ Whole bean ● Ground
Number of bags:	
Must arrive by date:	
Extras:	☐ Gift wrap ✓ Include catalog with order

Ship to

Name:	Buckaroo Banzai
Address:	Banzai Institute
City:	Los Angeles
State:	CA
Zip:	90050
Phone:	310-555-1212
Customer Comments:	

```
<div class="tableRow">
    <p> Number of bags: </p>
    <p> <input type="number" name="bags" min="1" max="10"> </p>
</div>
<div class="tableRow label">
    <p> Must arrive by date: </p>
    <p> <input type="date" name="date"> </p>
</div>
<div class="tableRow">
    <p> Extras: </p>
    <p>
        <input type="checkbox" name="extras[]" value="giftwrap"> Gift wrap<br>
        <input type="checkbox" name="extras[]" value="catalog" checked>
        Include catalog with order
    </p>
</div>
```

For the bean selection menu, the "beantype" radio buttons, and the "extras" checkboxes, we put all the form elements for each menu in one data cell.

```
<div class="tableRow">
    <p class="heading"> Ship to </p>
    <p></p>
</div>
<div class="tableRow">
    <p> Name: </p>
    <p> <input type="text" name="name" value=""> </p>
</div>
<div class="tableRow">
    <p> Address: </p>
    <p> <input type="text" name="address" value=""> </p>
</div>
```

Notice that we've also got an empty cell in the right column, so we can just put an empty <p> element here.

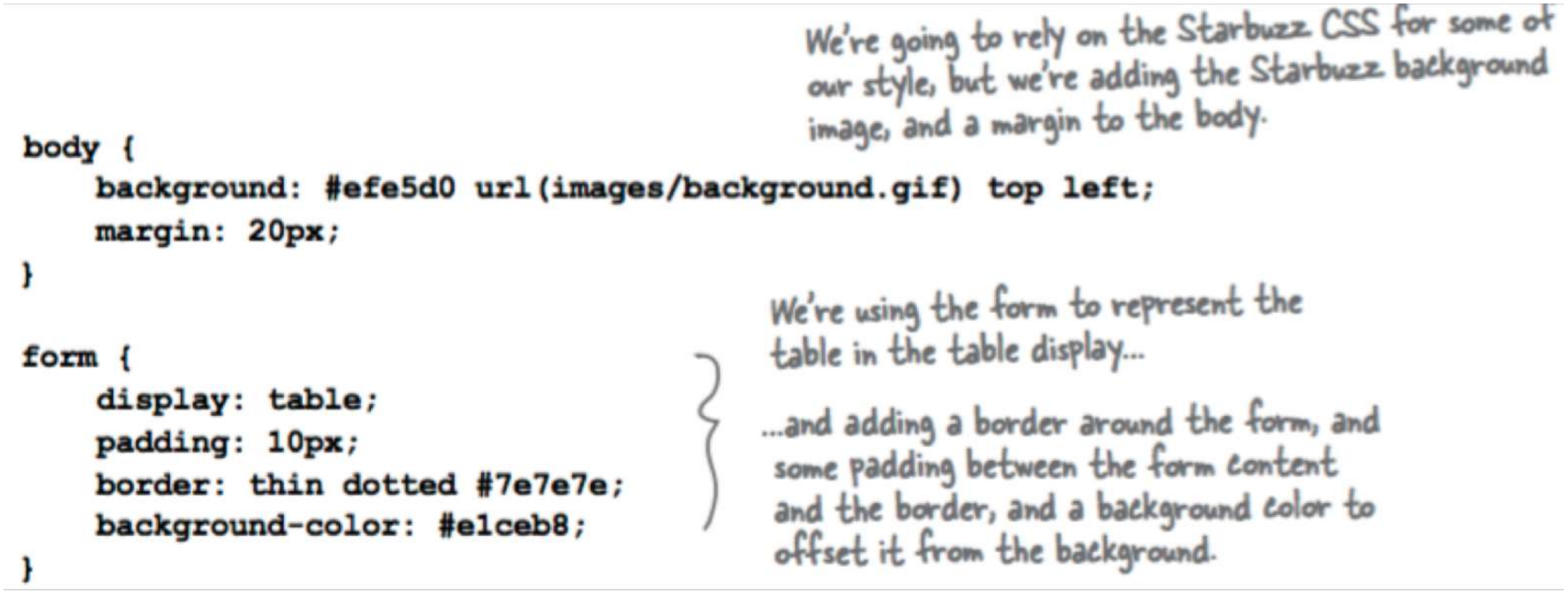
All the rows are straightforward: a "tableRow" <div> for the row, and each cell in a <p>.

```
<div class="tableRow">
    <p> City: </p>
    <p> <input type="text" name="city" value=""> </p>
</div>
<div class="tableRow">
    <p> State: </p>
    <p> <input type="text" name="state" value=""> </p>
</div>
<div class="tableRow">
    <p> Zip: </p>
    <p> <input type="text" name="zip" value=""> </p>
</div>
```

```
    <div class="tableRow">
        <p> Phone: </p>
        <p> <input type="tel" name="phone" value=""> </p>
    </div>
    <div class="tableRow">
        <p> Customer Comments: </p>
        <p>
            <textarea name="comments" rows="10" cols="48"></textarea>
        </p>
    </div>
    <div class="tableRow">
        <p></p>
        <p> <input type="submit" value="Order Now"> </p>
    </div>
</form>
```

And for the last row, we've got an empty cell in the left column, so again, we can use an empty <p> element for that.

Styling the form with CSS



We're going to rely on the Starbuzz CSS for some of our style, but we're adding the Starbuzz background image, and a margin to the body.

```
body {
    background: #efe5d0 url(images/background.gif) top left;
    margin: 20px;
}

form {
    display: table;
    padding: 10px;
    border: thin dotted #7e7e7e;
    background-color: #e1ceb8;
}
```

We're using the form to represent the table in the table display...

...and adding a border around the form, and some padding between the form content and the border, and a background color to offset it from the background.

```
form textarea {
    width: 500px;
    height: 200px;
}
```

We're making the textarea control in the form bigger, so there's more room for comments by setting its width and height.

```
div.tableRow {
    display: table-row;
}
```

Each "tableRow" <div> acts as a row in the table display layout.

```
div.tableRow p {
    display: table-cell;
    vertical-align: top;
    padding: 3px;
}
```

Each <p> element that is nested in a "tableRow" <div> is a table cell. We vertically align the content in each <p> so the content in each row lines up at the top of the cells. And we're adding a bit of padding here too, to add space between the rows.

```
div.tableRow p:first-child {
    text-align: right;
}
```

This rule uses the first-child pseudo-element on the selector for <p> elements nested inside "tableRow" <div>s. This means the first <p> element in each row is aligned to the right, so they all line up vertically against the right side of the column.

```
p.heading {
    font-weight: bold;
}
```

And for any <p> elements with the class "heading", we bold the text so it looks like a heading. We use this in the "Ship to" cell.

Password

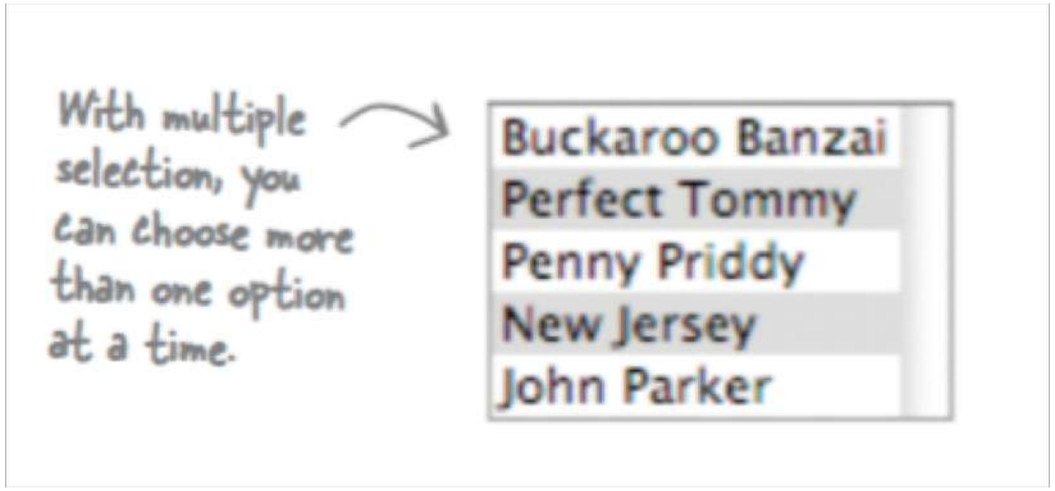
- The password <input> element works just like the text <input> element, except that the text you type is masked.

<input type="password" name="secret">

Multiple selection

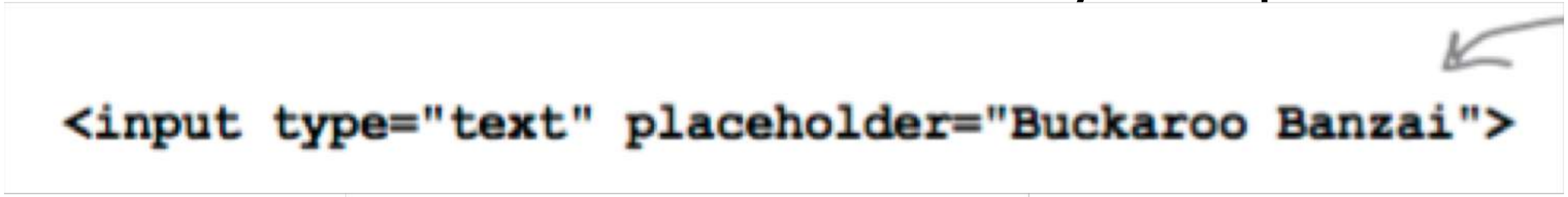
```
<select name="characters" multiple>
    <option value="Buckaroo">Buckaroo Banzai</option>
    <option value="Tommy">Perfect Tommy</option>
    <option value="Penny Priddy">Penny</option>
    <option value="New Jersey">Jersey</option>
    <option value="John Parker">John</option>
</select>
```

Just add the attribute multiple to turn a single selection menu into a multiple selection menu.

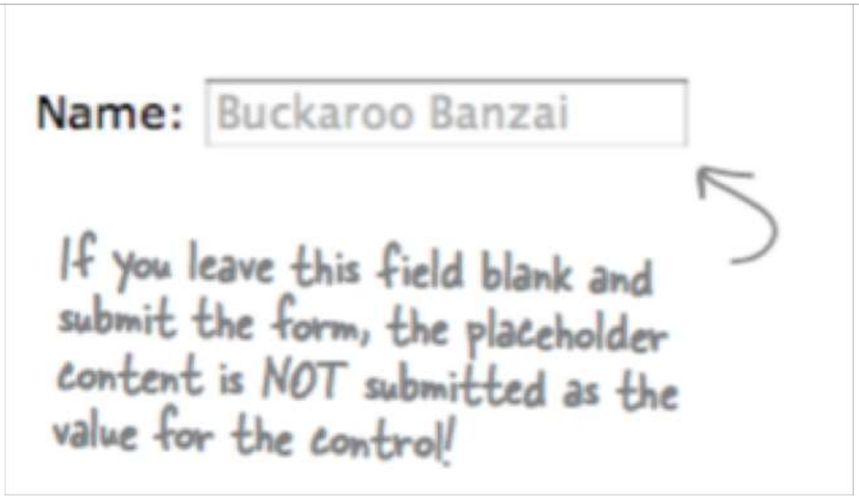

With multiple selection, you can choose more than one option at a time.

Placeholder

- You can use the placeholder attribute with most of the <input> types in a form to give the person who's filling out the form a hint about the kind of content you expect



```
<input type="text" placeholder="Buckaroo Banzai">
```



Name: Buckaroo Banzai

If you leave this field blank and submit the form, the placeholder content is NOT submitted as the value for the control!

Required and other input attributes

- It indicates that a field is required, so you shouldn’t submit the form without specifying a value for the controls that have this attribute set.

  <input type="text" placeholder="Buckaroo Banzai" required>
- *Not supported by all browsers!*

- Other attributes:

<input type="text" name="firstname" **maxlength**="10">

<input type="text" name="firstname" value="John" **readonly**>

<input type="text" name="firstname" value="John" **disabled**>

<input type="email" name="email" **autocomplete="off"**>

<input type="number" name="points" min=0 max=30 **step**="3">