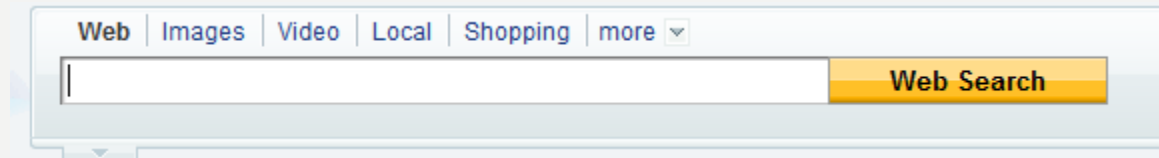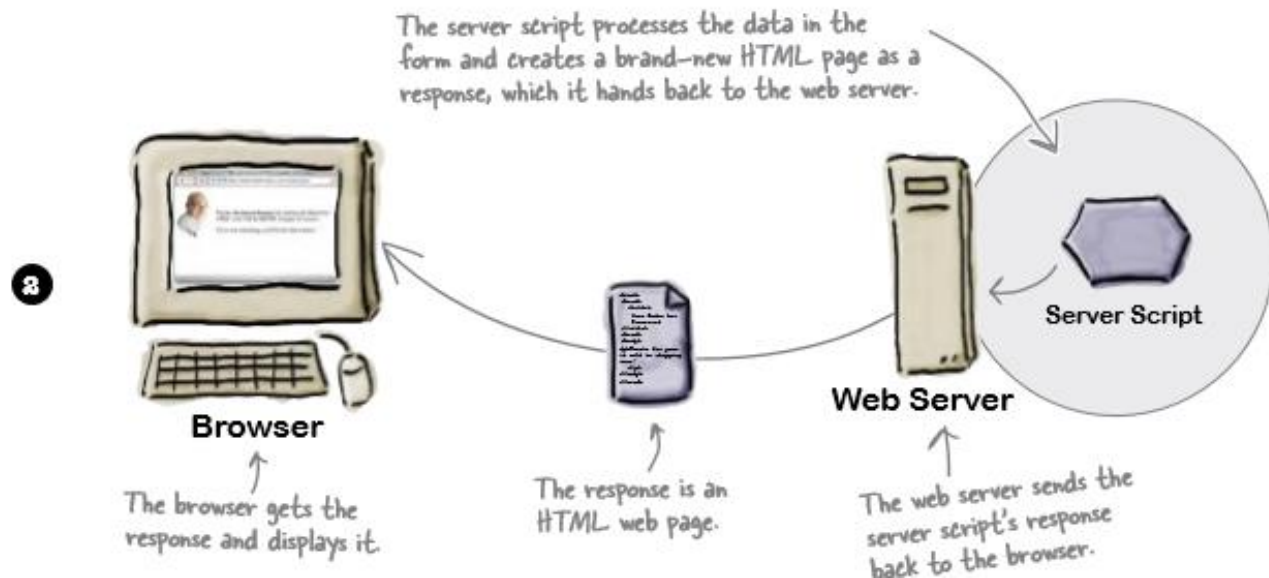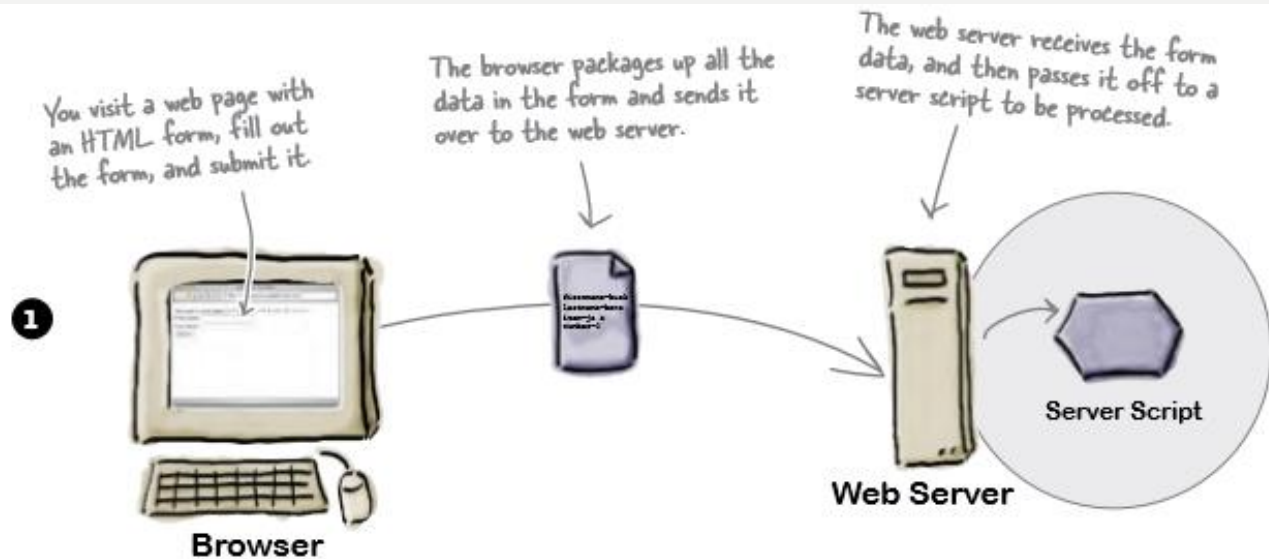# FORMS

OVERVIEW OF FORMS

- Forms are used all over the Web to
  - Accept information
  - Provide interactivity

- Types of forms:
  - Search form, Order form, Newsletter sign-up form, Survey form, Add to Cart form, and so on…

SERVER SIDE

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

Browser

Web Server

Server Script

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.

2

Browser

Web Server

Server Script

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.

3

HOW FORMS WORK IN THE BROWSER

The browser loads the page, it creates controls on the page that allow you to input various kinds of data.

You enter data, type a single line of text, click an option, checkbox control etc.

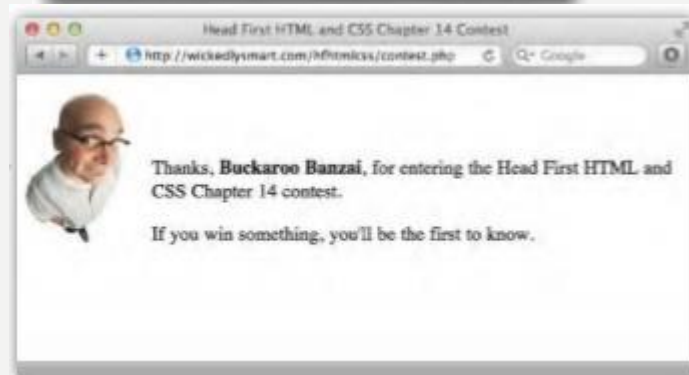You submit the form by clicking a submit button.

The server responds by receiving the data and processing it and presenting a new html page.

4

FORM ELEMENTS

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


    First name: <input type="text" name="firstname" value=""> <br>
    Last name: <input type="text" name="lastname" value=""> <br>
    <input type="submit">


   </p>
</form>
```

(A)
(B)
(C)
(D)

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

...and a bunch of elements nested inside it

5

FORM ELEMENTS

WHAT CAN GO ON A 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.

Name: Buckaroo Banzai

An <input> element with a type attribute of "text" creates a one-line control in the browser page.

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

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

WHAT CAN GO ON A FORM?

Radio buttons create a single control of several buttons of which only one can be chosen.

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."
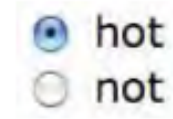
○ hot
○ not   The radio control allows only one of a set of choices.

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">
```

WHAT CAN GO ON A FORM?

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.

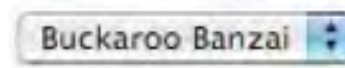Each checkbox has a different value.

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

WHAT CAN GO ON A FORM?

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.

Buckaroo Banzai ⏷

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

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

✓ Buckaroo Banzai
Perfect Tommy
Penny Priddy
New Jersey
John Parker

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

Just like the other form give the select element a name using the name att
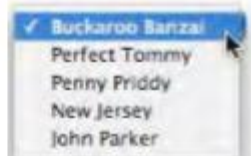
```
<select name="characters">
     <option value="Buckaroo">Buckaroo Banzai</option>
     <option value="Tommy">Perfect Tommy</option>
     <option value="Penny">Penny Priddy</option>
     <option value="Jersey">New Jersey</option>
     <option value="John">John Parker</option>
</select>
```

WHAT CAN GO ON A FORM?

HTML5 adds more input types.

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">
```

WHAT CAN GO ON A FORM?

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.

#ffc90e

Other...

`<input type="color">`

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

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

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">`

May   2012
Mon Tue Wed Thu Fri Sat Sun
10   1   2   3   4   5   6
7   8   9   10 11 12 13
14   15 16 17 18 19 20
21   22 23 24 25 26 27
28   29 30 31 1   2   3
4   5   6   7   8   9   10
                    Today

Like with color, if the date input isn't supported by the browser yet, you'll get a regular text input instead.
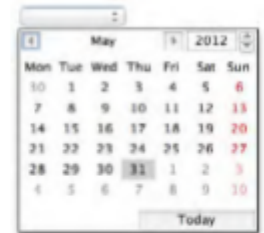
WHAT CAN GO ON A FORM?

Even with these specialised types, it's up to you to make sure you know what values the server script is expecting and use the right <input> type.

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

Not all browsers fully support these input types yet.

Watch it! The input types on these pages are new in HTML5, and while you can _use_ them in all web pages now, some may not _display_ as you see them here.

WHAT CAN GO ON A FORM?

Not every form element is an <input> element there are some others for typing more than one line of text:

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.

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 cols attribute tells the browser how many characters wide to make the text area.

```
<textarea name="comments" rows="10" cols="48"></textarea>
```
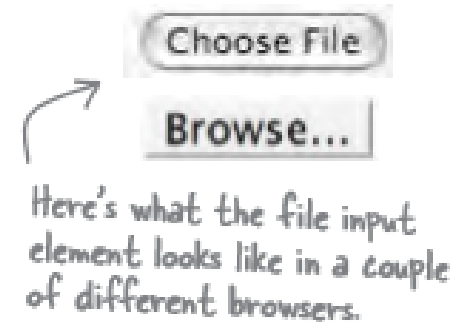
The rows attribute tells the browser how many characters tall to make the text area.

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.

WHAT MORE COULD GO INTO A FORM?

If you need to send an entire file you can set the input type to "**file**", this creates a control that allows you to choose a file and the contents when the form is submitted is sent to the server.

Choose File

Browse...

Here's what the file input element looks like in a couple of different browsers.

A **<fieldset>** element can be used to group together common elements.

It then can also make use of an element called **<legend>**.

Condiments

☑ Salt
☑ Pepper
☐ Garlic

Here's how the fieldset and legend look in one browser. You'll find that browsers display them differently.

WHAT MORE COULD GO ON A FORM?

The **placeholder** attribute can be used with most <input> types in a form, it gives a hint about the kind of content expected.

The **required** attribute forces the user to enter content. An error will show if the user leaves it blank.

If a form element includes the **autofocus** attribute then the cursor is in the form element when the form opens.

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!

Name: Buckaroo Banzai   Submit

⚠ Please fill out this field.
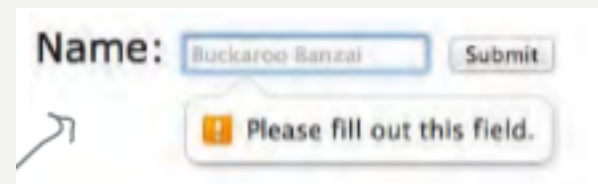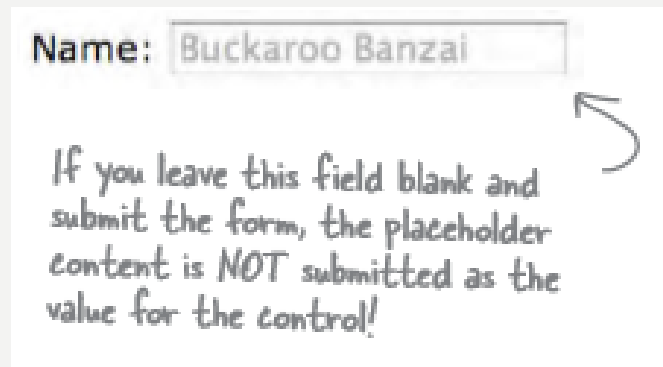
WHAT MORE COULD GO ON A FORM?

- We can also include a **Pattern**. This allows us to specify the type of data, and the format required.

    - pattern="[a-zA-Z0-9]": All alphanumeric characters.

    - pattern="[A-Za-z]{3}" : Three letter code.

    - pattern = "[A-Z][a-z]+([A-Z][a-z]+)*": Name (with optional second name, and more names...).

    - pattern ="\S{6,}": At least 6 characters (excluding space, tab, newline, carriage return, form feed).

    - pattern="\d{4,8}": Matches at least 4 digits but no more than 8 digits.

- Add a **title** attribute to add an error message and it is displayed when the cursor moves over the respective form element.

WHAT MORE COULD GO ON A 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">`
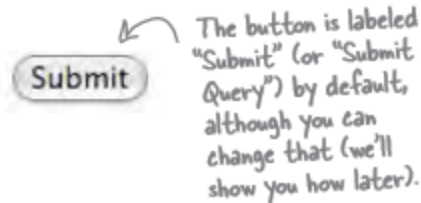
For a submit button, specify "submit" as the <input> element's type.

`<input type="reset">`

Defines a button on a form that will return all fields to their default values.

ACCESSIBILITY

- So far we have been labelling our form using text. We should really be using the **<label>** element to mark up these labels.

- The **<label>** element provides further information about the structure of your page, allows you to style your labels using CSS more easily, and helps screen readers for the visually impaired to correctly identify form elements.

- Labels don't look any different from normal text. But they can make a big difference when it comes to accessibility.

To use a <label> element, first add an
id attribute to your form element.

```
<input type="radio" name="hotornot" value="hot" id="hot">
<label for="hot">hot</label>

<input type="radio" name="hotornot" value="not" id="not">
<label for="not">not</label>
```

Then add a <label> and set its "for"
attribute to the corresponding id.

19

FORM LAYOUT

- Wrap labels and controls inside a <div> (or <fieldset>) with the Bootstrap **form-group** class applied for optimum spacing.
- Use Bootstrap's **form-control** class against textual <input>, <textarea>, and <select> elements to apply a width of 100% by default.

```
<div class="form-group">
  <label for="firstname ">First Name</label>
  <input type="text" class="form-control" id="firstname "
    name=" firstname ">
 </div>
```

First Name

FORM LAYOUT

- Default *checkboxes* and *radio buttons* are improved upon with the help of **form-check,** a single class for both input types that improves the layout and behavior of their HTML elements.

- Use the class **form-check-label** with the *label* element.

```
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" value="" checked>
    Option one is this and that - be sure to include why it's great
  </label>
</div>
```

☑ Option one is this and that—be sure to include why it's great

FORM LAYOUT

- You can use Bootstrap's grid classes to create horizontal forms. Just specify how many columns each element should span.

- Specifically, add a **row** class with the **form-group** and a **col-*-** or **col-*** class for each column.

- You should also add Bootstrap's **col-form-label** class to the *label* element.

- You may use **col-form-label-sm** or **col-form-label-lg** to your *label* elements to correctly follow the size of **form-control-lg** and **form-control-sm**.

FORM LAYOUT

```html
<div class="form-group row">
  <label for= "firstname" class="col-sm-2 col-form-label-sm">
    First Name</label>
  <div class="col-sm-10">
    <input type="text" class="form-control" id="firstname"
      name="firstname">
  </div>
</div>
```


First Name

MORE ON HOW THE <FORM> ELEMENT WORKS

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.

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

Everything inside your form goes here...

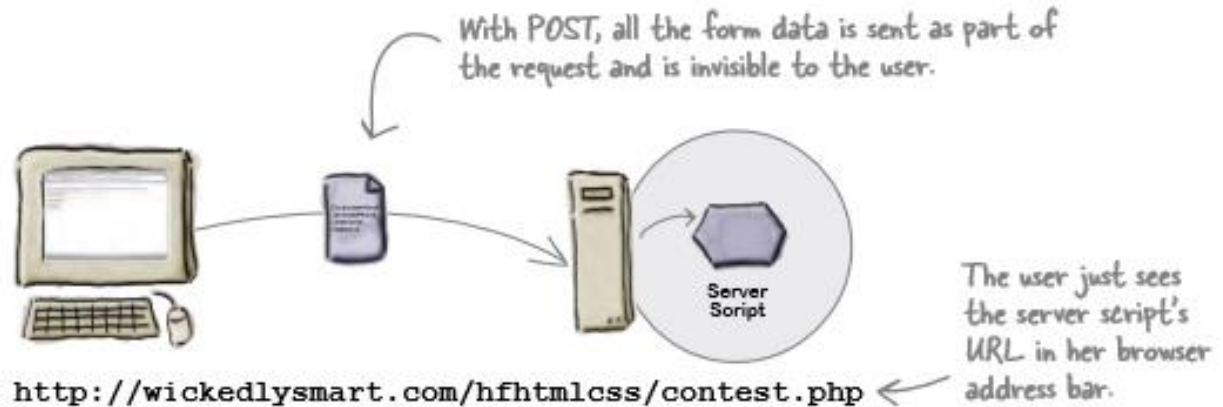`</form>` ...and the closing tag ends the form.
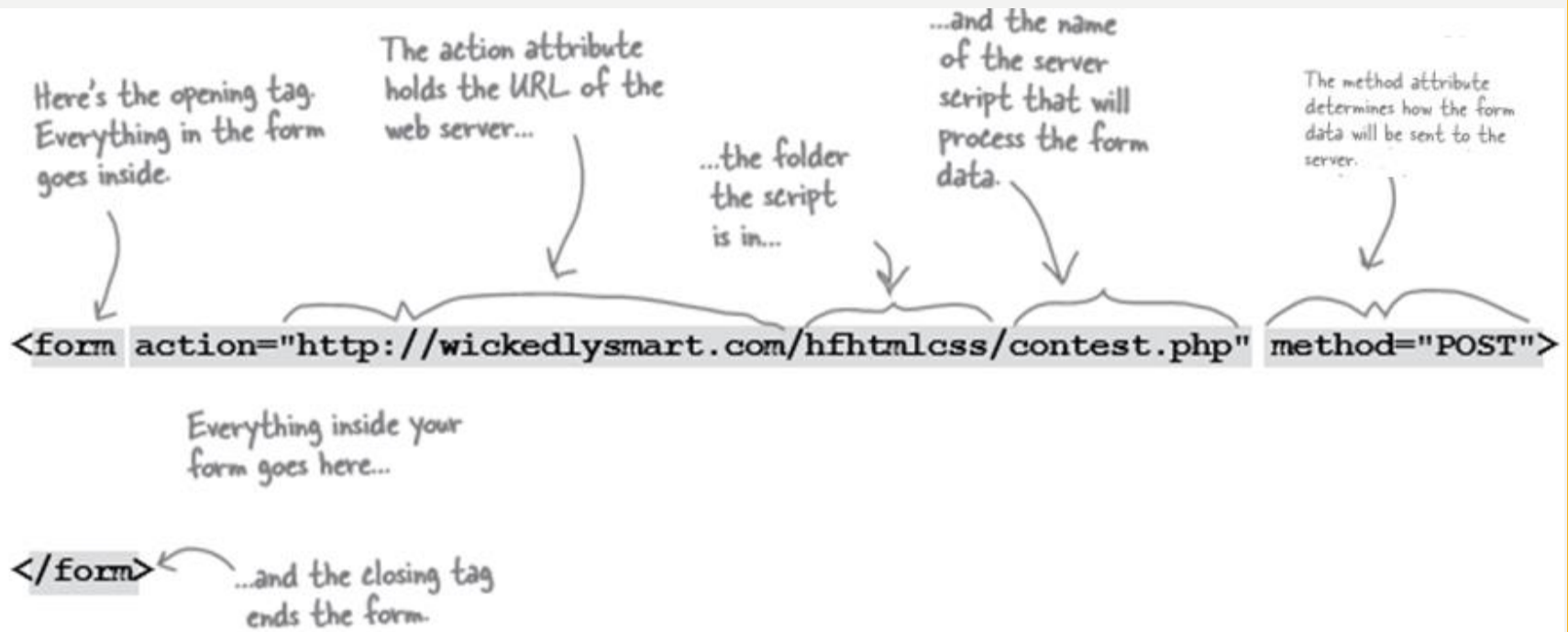
SENDING THE DATA TO THE SERVER

There are two primary methods the browser uses: POST and GET.

POST and **GET** accomplish the same thing—getting your form data from the browser to a server—but in two different ways. **POST** packages up your form variables and sends them behind the scenes to your server, while **GET** also packages up your form variables, but appends them on the end of the URL before it sends a request to the server.

SENDING THE DATA TO THE SERVER

With POST, all the form data is sent as part of the request and is invisible to the user.

POST

http://wickedlysmart.com/hfhtmlcss/contest.php

The user just sees the server script's URL in her browser address bar.

With GET, the form data is added to the URL itself, so the user sees the form data.

GET

Notice the form data added on to the end of the URL. This is what the user sees in the address bar.

http://wickedlysmart.com/hfhtmlcss/contest.php?firstname=buckaroo&lastname=banzai