

Forms

ART326

What is an HTML Form?

On the web, a form makes it possible for users to **pass data to a server**.

HTML forms provides a set of controls that make it possible to collect information from visitors.

What are some different types of forms?

Examples

Making a Purchase

Signing up for a Mailing List

Requesting Product Information

Contact Form

Donation Form

Search Boxes

Online Shopping Interfaces



How Forms Work

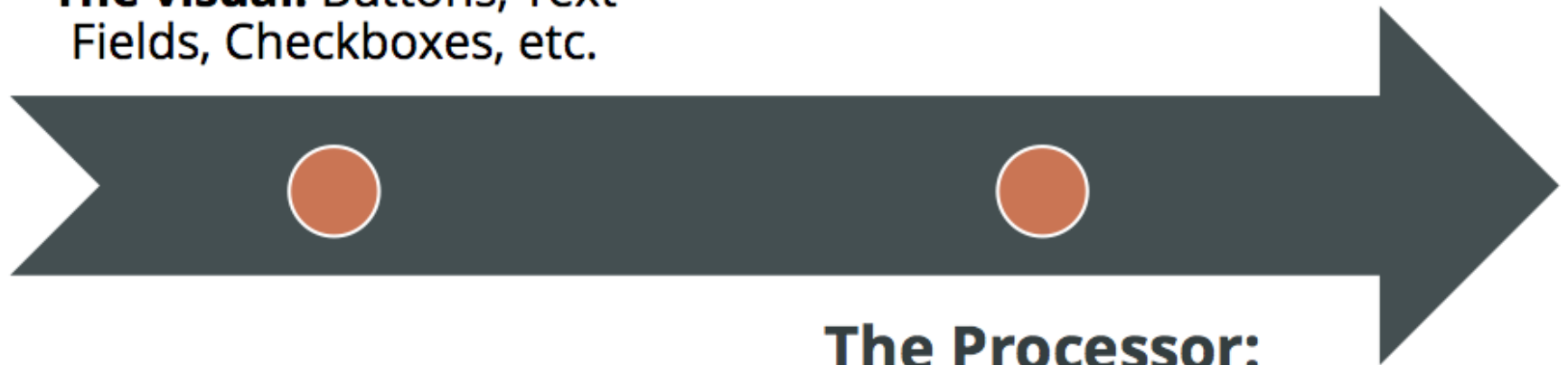
01. **Browser reads** your markup and renders form control elements on page.
02. Users **enter** info into fields and clicks: 
03. Browser **collects** info and encodes it.
04. Browsers **sends** it to the processor on the server.
05. Processor accepts info and **processes** it (however it was written to do so).
06. The web application returns a **response** and sends it back to the browser where it is displayed.

Simply, data is entered, collected, sent, processed, and then a response is displayed.

Form Components

The Form Controls:

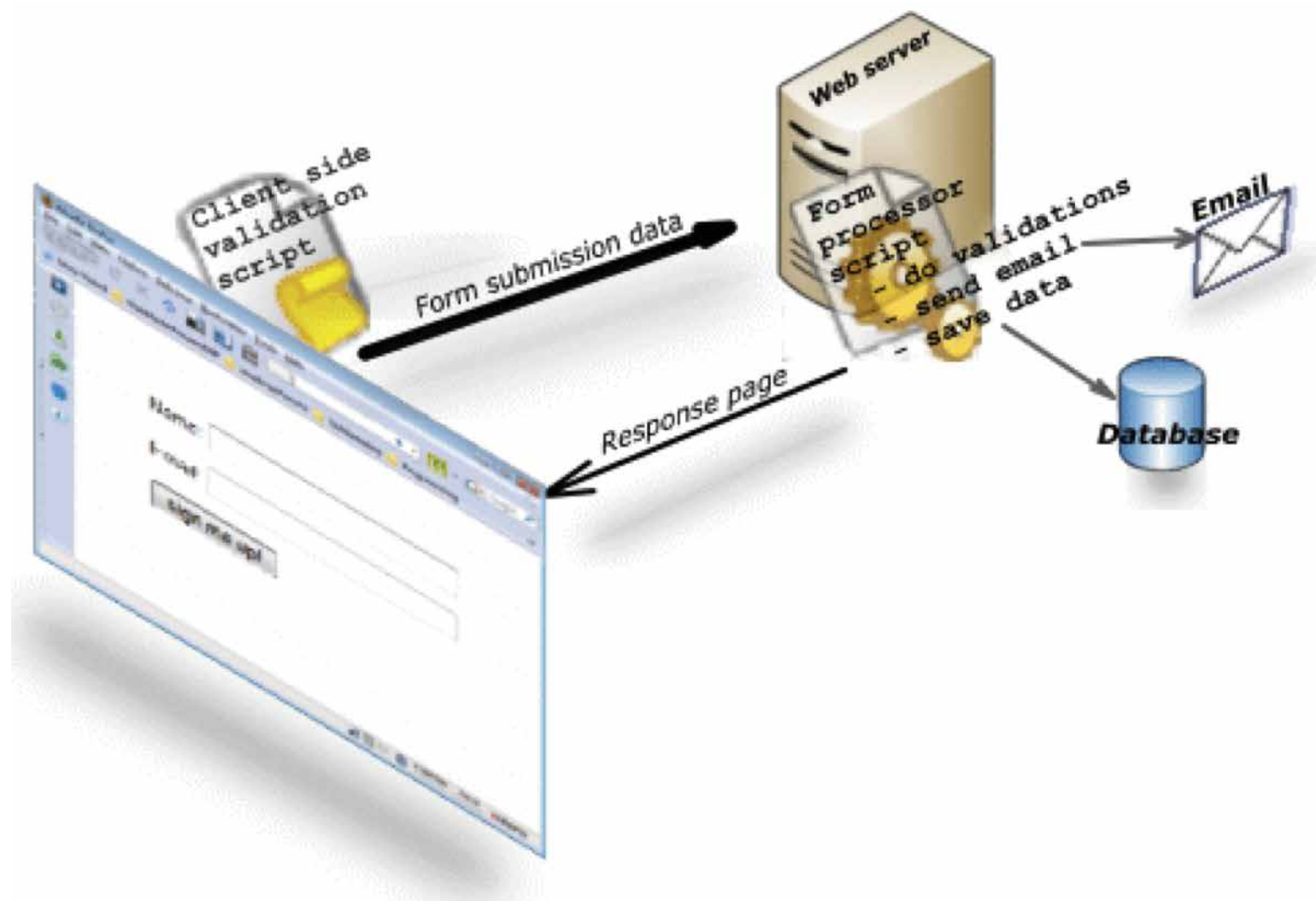
The visual: Buttons, Text Fields, Checkboxes, etc.



The Processor:

Make it work: An application or script that processes the info collected by the form, and returns a response.

The Steps of a Form Transaction



The <form> tag

<form>Where the magic happens.</form>

A **container** for all form content.

May also contain block-level elements (<p>).

May not contain another form.

In order for the form to actually do something, attributes, such as **action** and **method**, must be set.

The action attribute

Describes the **processing page**, location (URL) of the application or script.

In the text, you used a .php page to process.

PHP is an open source scripting language most commonly used with the Apache Web Server.

Most of the time, you will work with a web developer or server administrator who will provide the name and location of the program.

The method attribute

Specifies *how* the info should be sent to the server.

Two methods:

POST Sends a separate server request.

Only server sees data. Good for **private** info.

No character limit.

GET Encoded data gets **added to the URL** sent to the server.

Good for bookmarking form data, such as search results.

Not good for private information.

256 character limit.

Form Controls



Form Controls

A form element, or control, allows users to enter info or choose options (entry fields, buttons, menus, etc.).

Controls for Adding Text

text input: Single line of text (i.e. email, name)

First name:

Last name:

password input: Single line of masked text

Password:

textarea: Multi-line of text (i.e. messages, comments)

Enter your comment.

|

text input

First name:

Last name:

First name: `<input type="text" name="firstname">` `<br>`

Last name: `<input type="text" name="lastname">`

The **<input>** element is used for several different form controls.

When the type attribute has a value of **"text"** it creates a single line text input.

The **name attribute** tells the server the name of the data.

password input

Password:

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

The **<input>** element is used for several different form controls.

When the type attribute has a value of **"password"** it creates a single line text input that it masks.

The **name attribute** tells the server the name of the data.

textarea



`<textarea name="comments">`
Enter your comment.`</textarea>`

The **<textarea>** element creates a multi-line text area.

Any text in between will appear in the text box to the user.

The **name attribute** tells the server the name of the data.

Controls for Making Choices

radio buttons: One selection *must* be made.

☐ Male
☐ Female

checkboxes: Multiple selections can be made.

☐ I have a bike
☐ I have a car

drop-down boxes: One selection *must* be made from a list.

Volvo ▼
Volvo
Saab
Opel
Audi

radio buttons

☐ Male
☐ Female

```
<input type="radio" name="gender" value="male">Male<br>
```

```
<input type="radio" name="gender" value="female">Female
```

The **<input>** element is used for several different form controls.

When the **type** attribute has a value of **"radio"** it creates a radio button for one selection.

The **name** attribute tells the server the name of the data. Same for all buttons.

The **<value>** attribute indicates the selected option.

checkboxes

☐ I have a bike
☐ I have a car

`<input type="checkbox" name="vehicle" value="bike">`I have a bike

`<input type="checkbox" name="vehicle" value="car">`I have a car

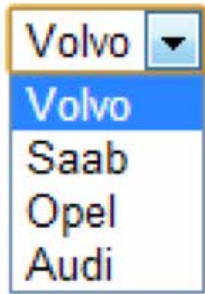
The **<input>** element is used for several different form controls.

When the **type** attribute has a value of **"checkbox"** it creates a button that allows multiple selections.

The **name** attribute tells the server the name of the data. Same for all buttons.

When checked, the **<value>** attribute indicates the selected option.

drop-down boxes



```
<select name="cars">
```

```
<option value="valvo">Valvo</option>
```

```
<option value="saab">Saab</option>
```

```
<option value="opel">Opel</option>
```

```
<option value="audi">Audi</option></select>
```

The **<select>** element is used to create a drop-down list box.

The **name** attribute tells the server the name of the data.

The **<option>** element indicates the options the user can select from.

The **<value>** attribute indicates the selected option.

Controls for Submitting

submit button: Submits data from your form to another page on the server.

Username:

image button: Same as above, but replaces button with an image.

Username:

submit button

Username:

```
<form name="input" action="html_form_action.php" method="get">
```

```
Username: <input type="text" name="user">
```

```
<input type="submit" value="Submit"></form>
```

When the **type** attribute has a value of **"submit"**, it is used to send a form to the server.

The **name** attribute can be used, but is not necessary.

The **value** attribute is used to control the text that appears on the button.

image button

Username:

```
<form name="input" action="html_form_action.php" method="get">
```

```
Username: <input type="text" name="user">
```

```
<input type="image" src="images.jpg" width="100" height="20"></form>
```

When the **type** attribute has a value of **"image"**, it allows you to use an image for the submit button.

The **name** attribute can be used, but is not necessary.

The **src**, **width**, **height**, and **alt**, attributes work the same as the `<img>` element.

Labelling



Why Use Labels?

Labels identify the purpose of each form control.

By using labels, the form is accessible to the visually impaired.

Some labels we will use include:

`<label>`

`<fieldset>`

`<legend>`

The diagram illustrates a sign-up form with various HTML labels. A `<fieldset>` label points to the entire form container. A `<legend>` label points to the 'Sign-up Form' title. An `<input>` label points to the first text input field. A `<label>` label points to the 'Name' text label. Another `<label>` label points to the 'Newsletter?' text label. A `<button>` label points to the 'Send' button. The form includes input fields for 'Name', 'Email', and 'Psw', and radio buttons for 'Free Subscription' and 'Professional'.

The <label> tag

Labels a form control.

One per form control.

Two ways to use:

Nested: Label wraps around the text description and the form control.

Separate (for): Label kept separate. Uses “for” attribute to indicate which form control it is labeling.

Nested vs. Separate

Nested: Label **wraps around** the text description and the form control.

```
<label> Age: <input type="text" name="age" /></label>
```

Separate (for): Label kept separate. Uses **“for” attribute** to indicate which form control it is labeling.

```
<label for="year"> Age: </label>
```

```
<input type="text" name="age" id="year" />
```

Try It: http://www.w3schools.com/tags/tryit.asp?filename=tryhtml_label

The <fieldset> tag

Groups form elements together.

Helpful with longer forms.

Most browsers show a line around the edge of the fieldset. Can be adjusted using CSS.

My Radio buttons

- ☐ Option 0
- ☐ Option 1
- ☐ Option 2
- ☐ Option 3
- ☐ Option 4

**<fieldset>Wraps around
grouped form elements.</
fieldset>**

The <legend> tag

Comes directly after the opening of the <fieldset> tag.

Contains a caption which identifies the purposes of that control group.

My Radio buttons

☐ Option 0
☐ Option 1
☐ Option 2
☐ Option 3
☐ Option 4

```
<fieldset>  
<legend>My Radio  
buttons</legend>  
</fieldset>
```

Sending Info



Receiving Info

Processing Info

Storage Info

Name/Value

*If forms collect data,
how is it sent to the server?*



All information is sent in name/value pairs.

The **name attribute** names the control.

Data entered/selected by the user, is the value.

All form control elements (except Submit and Reset) must include a name so that the form processor can sort the info.

TIP: You should never change the name of a form control unless you are certain the server code will understand it.

Name/Value Example

In this example, the **name** is “comment” and the **value** is “Would you like to comment?”:

```
<textarea name =“comment”>Would you like to  
comment?</textarea>
```

PHP Form Example

Contact Form

Subject :

Detail :

Name :

Email :

action="send_contact.php"

```
<?php
$subject = "$subject";
$message = "$detail";
$mail_from = "$customer_mail";
$header = "from: $name <$mail_from>";
// Enter your email address
$to = 'someone@somewhere.com';
$send_contact = mail($to, $subject, $message, $header);
```

Domains + Hosting

ART326

Servers



Image Source: oactechnology.com

The **software** on computers that “serves up” documents upon request.

Requests for info are made to other computers that host the website pages.

Makes communication between computers possible.

MOST POPULAR WEB SERVERS:

Apache, Microsoft + Nginx

The Server does the Servin'...



Every computer on the Internet is assigned a unique numeric **IP (Internet Protocol) Address**, can be thought of like a phone number for that computer.

oreilly.com is 208.201.239.37

The Domain Name System (DNS), the “phone book,” was developed to allow us to refer to that server by its **domain name**. Translates domain names into IP addresses.

Domain Name

Your **domain name** is the name that points to your site.
Associated with a numeric IP address (74.220.195.31).

Ex. cassreese.com

To register this name, you must find a **registrar**, and typically pay an annual fee.

The person that registers a domain name is called the **domain name registrant**.

As millions of names have already been registered, you have to check if the name you want is available.

Domain Name Suffixes

Every domain name has a suffix:

.gov	Government agencies
.edu	Educational institutions
.org	Organizations (nonprofit)
.mil	Military
.com	Commercial Business
.net	Network organizations
.ca	Canada
.th	Thailand

Web Hosting

Many sites that offer domain registration, also offer hosting.

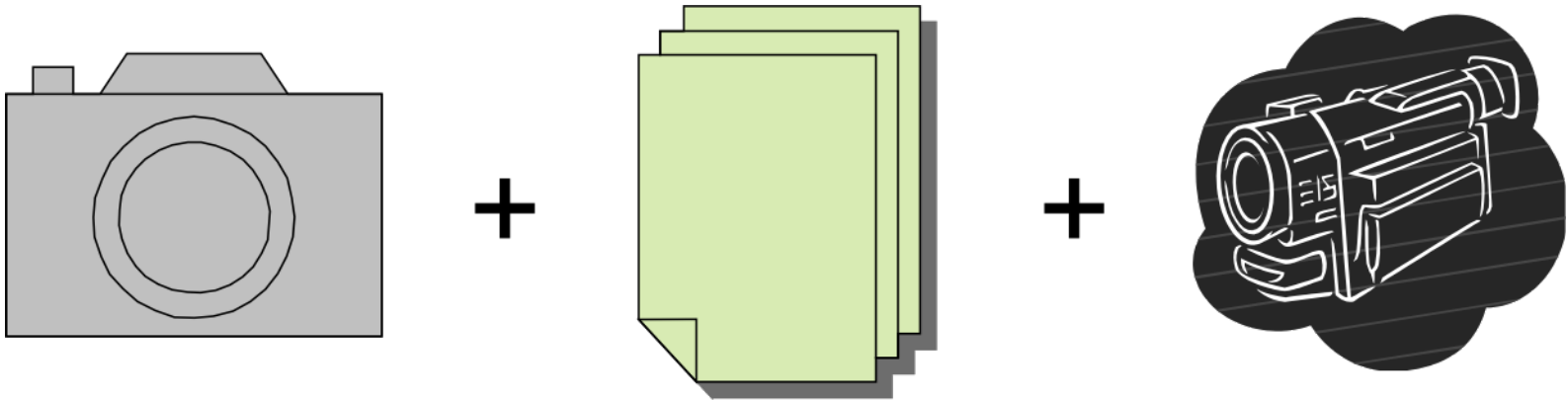
Web hosting companies own servers that you can purchase space from.

In addition to **space**, you are also provided **with an IP address**.

Most companies purchase space from a large hosting company, as it is cheaper and more reliable.

Disk Space

The total size of your everything on your site.



Bandwidth

The **amount of data requested** by the visitors of your site.

If 6 people visit your site, the bandwidth would be 6 times your used disk space.



Backups



The storing of your site on another backup server.

Each company has your back in a different way:

Some allow you access to former site backups (in the case you break your site).

Some only create backups in the case that their server breaks.

Whols Database

An Internet utility that returns information about a domain name or IP address.

For example, if you enter a domain name such as microsoft.com, whois will return the name and address of the domain's owner (in this case, Microsoft Corporation).

Registrants can opt to have their information kept private for an additional fee. **I recommend this.**

More information: <http://www.networksolutions.com/education/you-need-private-registration/>

Email Accounts

Most hosting companies offer an email server with your package.

The mailbox size and number may vary.

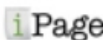
Example: `cass@cassreese.com`

The Process

01. **Buy the domain name.**
02. **Find a host.** I recommend using the same company as your registrar. In the case that you do not, you will have to manage the DNS (Domain Name System). Here is an article that helps this process: <http://coding.smashingmagazine.com/2011/05/25/introduction-to-dns-explaining-the-dreaded-dns-delay/>

Registrars / Hosting Companies

There are companies that will provide a free domain name if you buy their hosting.

#1	#3	#4	#5	#6	#7	#8	#9	#10	#11
 Just Host \$2.50	 iPage \$2.25	 Arvixе \$4.00	 MidPhase \$7.99	 HostGator \$3.96	 AN Hosting \$3.95	 Network Solutions \$3.99	 Globat \$4.44	 WestHost \$4.00	 FatCow \$3.15
#12	#13	#14	#15	#16	#17	#18	#19	#20	
 BlueHost \$3.95 REVIEW	 GreenGeeks \$3.96 REVIEW	 Lunarpages \$8.95 REVIEW	 myhosting \$4.95 REVIEW	 Hostway \$6.95 REVIEW	 Doteasy \$4.95 REVIEW	 HostRocket \$5.99 REVIEW	 DreamHost \$8.95 REVIEW	 HostMonster \$4.95 REVIEW	

<http://www.whoishostingthis.com/compare/>

<http://web-hosting-review.toptenreviews.com>

The Process: Launching Your Site

01. Login to FTP. Enter the server name, login, and password.
02. Navigate your folders. Find the root folder on your server.
03. Upload Site.
04. Browser Check. Open your site in a browser. Voila!

The Control Panel

User interfaces that allows you to manage your website.

Linux: Cpanel

Window: Plesk

<https://www.youtube.com/watch?v=eva6DYOA6kQ>

