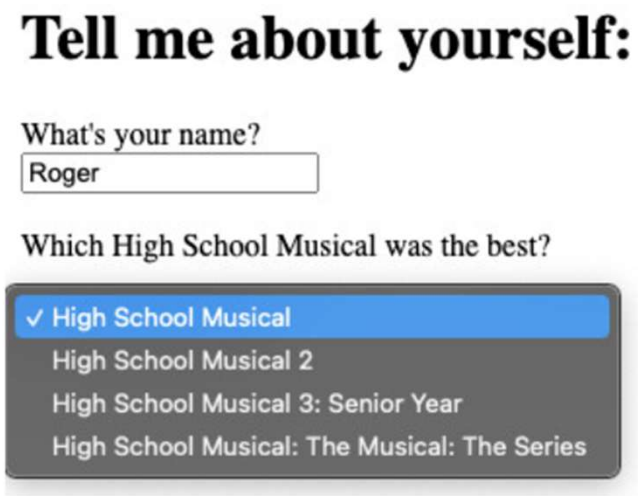
12.4 HTML Forms

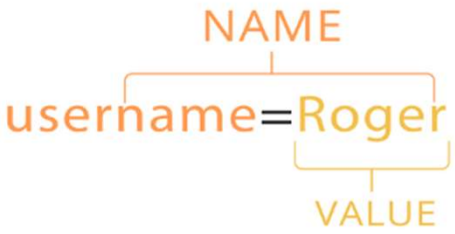
How do forms work?

The pages you've created so far have been great. However, the visitor could only passively receive the information you give — they couldn't share anything themselves.

Adding forms to your code allows the user to submit information to your server. For example, take this form:



Once Roger presses submit, the form immediately sends the server each of the input names alongside Roger's chosen values. To make sure the server can differentiate the various types of information, it's sent out from the browser in **name-value** pairs. Here's what it could look like:



> **Note: Don't change the name of a form control unless you're absolutely confident the server will be able to understand the change.*

So what happens to this data? Well, the server can then either process it using another programming language, like Java or PHP, or store it in a database somewhere.

On Roger's side, he receives a server-generated message based on the value he chose.

Thanks, Roger!

You voted for High School Musical

Adding forms

The main benefit of using forms is that they allow you to group individual input elements together. This way, instead of having to submit each input to the server separately, you can do it all at once.

> **Note: You can use JavaScript to submit separate inputs programmatically, as you'll see later on in this course. However, until then, HTML forms will be the main tool for forwarding user data to the server.*

As you saw in the video, forms can be created with a simple `<form>` element, within which all future inputs will be added. There are two main attributes that come along with it: `action` and `method`.

action attribute

The action attribute is mandatory for the `<form>` element to work.

> *Note: Technically, instead of the `action` attribute, you can also use a `formaction` attribute on the button itself. But for now, let's stick to the first one.*

The `action` attribute specifies the URL for the server which receives the submitted information from the form.

```
<form action="/action_page.php">
<h1> This is a form! </h1>
</form>
```

TRY IT OUT

method attribute

Another required addition to your form element is the `method` attribute. It specifies the HTTP method for transmitting your submitted data. There are two ways of doing so: GET and POST.

GET

Description: The form values are added to the end of `action` attribute's URL. You don't need to specify it as it's the default method. For example, the code snippet above uses the GET method.

When to use: When you're getting something from the server (rather than sending it forward to a database).

POST

Description: The form values are sent through HTTP methods.

When to use: If your form has a file upload function, contains confidential data, or you wish to send the information to a database.

Task: Go back to our code editor (you can use the *Try it Out* button below). Change the form's method to POST.

TRY IT OUT