

Office of Digital Humanities

[Back](#)   [BYU LiveCode Lessons Gateway](#)

## Understanding HTML Forms

### GET and POST Methods

#### The `form` tag

It isn't too difficult to read a form tag in an HTML document to figure out how to form a valid argument list for submitting form data with a GET or POST method. First you want to examine the source code of the web page that has the form on it. The form tags look something like this:

```
<form action="http://url.here.com/" method = "POST" (or "GET")>
  - input elements go here -
</form>
```

The `action=` attribute tells you the URL to send the argument list to, and `method=` tells you whether to use GET or POST. The various input element types within the form will tell you what kinds of information you need to send to the URL to form a valid request. The most used tag within forms is the `<input>` tag. The type of input is specified with the `type` attribute.

#### Some common input types:

Name: `<input type="text" name="fullname" />`

Looks like this:

Name:

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

Looks like this:

☐ Male  
☐ Female

```
-----
<input type="checkbox" name="agree" value="true"> I agree to these terms.</input>
```

Looks like this:

☐ I agree to these terms

**Note:** When a checkbox input is checked, a `name=value` pair is sent with the argument list. However, if it is not checked, no `name=value` pair is sent for the input.

```
-----
<select name="color">
  <option value="blue">blue</option>
```

```
<option value="yellow">yellow</option>
<option value="red">red</option>
<option value="green">green</option>
</select>
```

Looks like this:



### Understanding name=value pairs

The key to finding the components you need to construct argument lists is to look for `name=""` and `value=""` elements inside form input tags to get `name=value` pairs. In fact the whole purpose of an HTML form is to create strings of `name=value` pairs. These essentially are made into arrays by the host server when it receives the data. Remember, an [array](#) is made up of a "key" (corresponding to the "name" attributes in the form) and the actual data (corresponding to the "value" attributes in the form).

So when trying to figure out what to GET or POST by reading an HTML form, look for the keywords `name="xxxx"` and `value="yyyy"`. That will tell you what names and values to put into your argument list for either GET or POST.

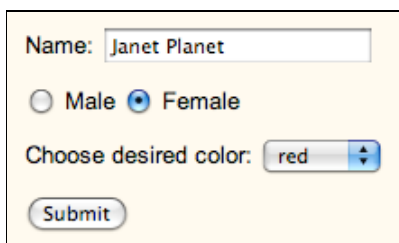
An arg list is always in the form:

```
name1=value1&name2=value2&name3=value3
```

So if you wanted to pass on the `name=value` pair for the radio button input group shown above, you would look at the `input type="radio"` tag. You see `name="gender"` and `value="male"` or `value="female"`. Whichever button the user chooses will be the value that gets submitted with the arguments list. So if the user clicked on Female you would find this string somewhere in the argument list string for either GET or POST:

```
gender=female
```

Let's consider a more complete example. Assume a user filled out the following form and clicked Submit:



The argument list string generated might look like this:

```
fullname=Janet+Planet&gender=female&color=red
```

You can look at virtually any form element in an HTML source and, using this type of analysis, determine exactly what kind of argument list you need to construct in your LiveCode stack to send a valid argument string to a web service.

---

[Go Back](#)

---

[Back](#)   [BYU LiveCode Lessons Gateway](#)

Maintained by Devin Asay.

Copyright © 2005 Brigham Young University.

This page last updated on March 09, 2020 17:52:07.