

Chapter 12 — Forms and Input Elements

One of the most powerful features of HTML is the ability for users to fill in information on forms and to have the data sent to remote servers to process. This chapter will introduce some of the most common form elements and how to use them to create pages to collect information.

Learning Outcomes for this Chapter

Upon completion of the reading of this chapter and assignments found at the end, a student should be able to:

- Recall the two modes (get and post) and their differences,
- create input boxes of various types,
- use drop downs and radio buttons on a page,
- associate labels to fields,
- submit data to a server and see the result.

Form Tag

Have you ever filled out information on a webpage, such as your address when shopping online or your name while signing up for an online membership? These areas are called forms, and are simple to add to any HTML document.¹³¹

Forms are added inside of the `<form> . . . </form>` tag which are usually nested within the `<body> . . . </body>` of a document. Each form tag includes an **action** and a **method** attribute, which dictates how the form will function and where the information will be sent to.

The **action** attribute tells the webpage where to send the form information. It will contain the URL of a program on the web server that will process the data. These programs on the web server are known as Common Gateway Interface scripts or CGIs for short.¹³² A CGI may be written in either a compiled language (c, c++, java, etc) or in a scripting language (php, javaScript, ...).

The **method** attribute in a form communicates to the web browser how to send the data. This can either be done using **method="get"** or **method="post"**. The **get** method sends the data to the action URL as a full package, and will appear as such in the address bar:

<http://www.syw21.org/reflect.php?username=Jim&age=53&title=Associate Professor>

¹³¹ <https://html.spec.whatwg.org/multipage/forms.html>

¹³² http://en.wikipedia.org/wiki/Common_Gateway_Interface



The diagram illustrates data sent to a server. It is divided into two sections: 'GET Data' and 'POST Data'. The 'GET Data' section contains a table with two columns: 'Key' and 'Value'. The rows are: 'username' with value 'Jim', 'age' with value '53', and 'title' with value 'Associate Professor'. The 'POST Data' section contains a table with two columns: 'Key' and 'Value', but it is currently empty.

Illustration 40: Data Sent to a Server

In the above example, the data following the `?` is in **key=value** pairs, separated by an **&**. This method is quick, but is also limited in the amount and type of data that can be sent. In contrast, the **post** method sends the data to the server as a completely separate stream and can upload a large quantity of data. The type of method that you use will entirely depend on the type of server and the application.

There are many elements that can be used inside of a form, which will be explained below. Keep in mind as you are reading that all form data controls must include the **name** attribute, which functions similarly to an **id**, and is used to identify the information being sent to the CGI.¹³³

The Input Tag

The `<input>` tag is a phrasing element that is used by the user to input information, such as their name or email address.¹³⁴ As with any form element, there are a few attributes that need to be included. An example input tag could be written like:

```
<input type="text" name="valuenam" value="type something here">
```

The **type** attribute specifies what kind of data will be input into the form element, and can be specified in a multitude of different ways. Here is a list of some of the most common types of form inputs:

- **type="text"** — This **type** specifies that this tag is a text entry field. It will display a general box for the entry of a single line of text.
 - may use the **maxlength** attribute to specify the maximum number of characters to

¹³³ <https://html.spec.whatwg.org/multipage/form-control-infrastructure.html>

¹³⁴ <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>



- allow in the field (also works for input types **search**, **tel**, **url**, **email**, and **password**)
- may use **size** attribute to specify how many characters long the box will be displayed as (also works for input types **search**, **tel**, **url**, **email**, and **password**)
- **type="email"** — This type of text field is used specifically for email addresses. Using this attribute will ensure that the address is properly formatted with valid characters, and will inform the user when it is not so that they may correct it. ¹³⁵
- **type="url"** — This type will ensure that a properly formatted URL is entered in the input field. This field type does not validate that the URL exists just that the format is proper. ¹³⁶
- **type="tel"** — This form input type is used for telephone numbers. Some browsers will display numeric keypads for ease of use. ¹³⁷
- **type="password"** — This is a special type for passwords. Anything typed into this form element will be displayed a a dot or an asterisk, ensuring privacy.
- **type="number"** — A field of this type will only allow the user to enter numbers. You may specify the **min**, **max**, and **step** attributes to enforce limits on the numbers being entered. ¹³⁸
 - If you wanted the user to enter a whole number from 0 to 100 the field might look like:
`<input type="number" name="score" min=0 max=100 step=1>.`
 - **type="range"** — This field will display a slide-bar that the user can interact with to select a number. Though it looks nice, it can be inaccurate. You may use the **min**, **max**, and **step** attributes to change the behavior of the slider. ¹³⁹
 - Let's say you want the user to enter a number from 0 to 1 with one decimal point, the field might look like:
`<input type="range" name="somenumber" min=0 max=1 step=.1>.`

135 [https://html.spec.whatwg.org/multipage/input.html#e-mail-state-\(type=email\)](https://html.spec.whatwg.org/multipage/input.html#e-mail-state-(type=email))

136 [https://html.spec.whatwg.org/multipage/input.html#url-state-\(type=url\)](https://html.spec.whatwg.org/multipage/input.html#url-state-(type=url))

137 [https://html.spec.whatwg.org/multipage/input.html#telephone-state-\(type=tel\)](https://html.spec.whatwg.org/multipage/input.html#telephone-state-(type=tel))

138 [https://html.spec.whatwg.org/multipage/input.html#number-state-\(type=number\)](https://html.spec.whatwg.org/multipage/input.html#number-state-(type=number))

139 [https://html.spec.whatwg.org/multipage/input.html#range-state-\(type=range\)](https://html.spec.whatwg.org/multipage/input.html#range-state-(type=range))



The step determines the interval of the available numbers. For example, a step of 5 will allow the user to select only every 5th number.

- **type="date"** — This field type can be used to enter a date. In many browsers, a data selection dialog will pop up to assist in quick and accurate entry.
- **type="checkbox"** — This will generate a box that the user can check or not. Checkboxes are independent and clicking on one will not usually effect another on the page.
- **type="radio"** — A radio similar to the checkbox, however only one option in a radio can be selected at a time. Each radio button in a group will have the same **name** attribute but will have a unique **value** that will be used by the server to see what was selected. Radios may also have a unique **id** if needed for scripting or styling.
- **type="hidden"** — A hidden field is special. It will send data to the server but will not be displayed on the web browser. You need to specify a **value** attribute that contains the data to be sent.
- **type="submit"** — Using the submit input type will generate a button that will send the completed form to the server.
- **type="reset"** — The reset type will generate a button that will clear every field and return them to their original state

There are several other different attributes that can be applied to the `<input>` tag to make them behave in special ways: ¹⁴⁰

- **maxlength** — This attribute can be used to specify the maximum amount of characters allowed in an input field. For example, if you have an input box for a telephone number, you may want to limit the number of characters to 10 in order to prevent the user from accidentally adding too many numbers.
- **sizerequired** — This attribute will specify the length of an input box by how many characters are required.
- **disabled** — This attribute will disable an input box, not allowing a user to interact with it. The primary use of this attribute is to disable inputs on conditions, such as preventing users from entering their phone number if they have previously selected that they don't have a phone.
- **readonly** — This attribute will prevent a user from changing the value of an input.

140 <https://html.spec.whatwg.org/multipage/input.html#common-input-element-attributes>



The Label Tag

The `<label>...</label>` tag is a container for `<input>` tags and can be used to associate a prompt to the input.¹⁴¹ For example:

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

The `<label>...</label>` tag can be useful for organization and for creating CSS.

If the `<input>` tag is defined outside the `<label>...</label>`, you should include the `for` attribute to associate the input element and the label's text. It is written like:

```
<label for="namefield">Name:</label><input type="text" id="namefield"  
name="name">
```

Keep in mind that using the `for` attribute requires the `id` attribute to be included in the form element.

Let's suppose that we need to develop a simple registration page for our company picnic, next week. The server developer would like us to send the data to <http://www.syw2l.org/reflect.php> as get data and has given us the following field names, types, and lengths:

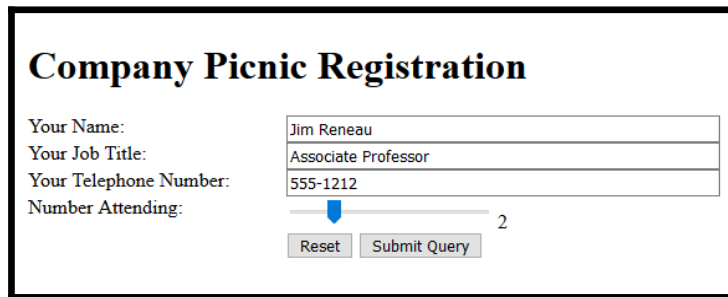
Field Name	Type	Length or Range
Name	Text	50
Title	Text	50
Telephone	Telephone Number	15
NumberAttending	Number	0-10

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Company Picnic</title>  
    <style>  
      form {  
        display: grid;
```

¹⁴¹ <https://html.spec.whatwg.org/multipage/forms.html#the-label-element>



```
        grid-template-columns: 13em 1fr;
    }
</style>
</head>
<body>
  <h1>Company Picnic Registration</h1>
  <form method="get" action="http://www.syw2l.org/reflect.php">
    <label for="formName">Your Name:</label>
    <input id="formName" name="Name" type="text"
      size=50 maxlength=50>
    <label for="formTitle">Your Job Title:</label>
    <input id="formTitle" name="Title" type="text"
      size=50 maxlength=50>
    <label for="formTele">Your Telephone Number:</label>
    <input id="formTele" name="Telephone" type="tel"
      size=15 maxlength=15>
    <label for="formAttend">Number Attending:</label>
    <div>
      <input id="formAttend" name="NumberAttending" type="range"
        min=0 max=10 step=1 value=1
        oninput="formAttendShow.value = formAttend.value">
      <output id="formAttendShow">1</output>
    </div>
    <div></div>
    <div id="buttons">
      <input type="reset">
      <input type="submit">
    </div>
  </form>
</body>
</html>
```



Company Picnic Registration

Your Name: Jim Reneau

Your Job Title: Associate Professor

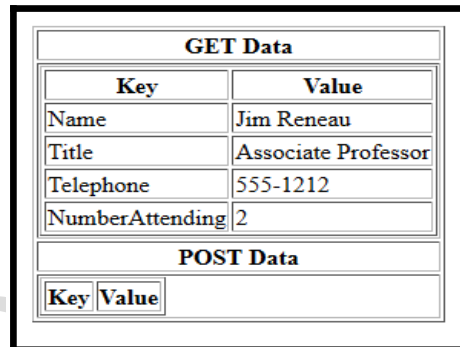
Your Telephone Number: 555-1212

Number Attending: 2

Reset Submit Query

Illustration 41: Company Picnic - Registration Form





GET Data	
Key	Value
Name	Jim Reneau
Title	Associate Professor
Telephone	555-1212
NumberAttending	2

POST Data	
Key	Value

Illustration 42: Company Picnic -
Registration Form Data

In the previous example a couple of tricks were one. The two buttons and the range input and output were combined into `<div>...</div>`s. This was done so that they would be a single grid item in the two column grid layout. Also you will see

`oninput="formAttendShow.value = formAttend.value"` on the range input. The string following `oninput` is actually a single JavaScript statement that takes the value of the slider (`formAttend`) and assigns it to the output element (`formAttendShow`) so that the user can see the range selector and see the number selected.

Other form Elements

- `<select>...</select>` — This form control allows the user to select a choice, which is usually displayed as a drop-down list with zero or more options.^{142 143}
 - Options within a select written with the `<option>...</option>` tag, like:
`<option value="g">green</option>`
In this example, the "g" is the value that will be sent to the server. It is possible to group options, however it is an advanced topic that will not be covered in this volume.¹⁴⁴
- `<fieldset>...</fieldset>` — Is used to group form elements together and enhances the organization of your webpage.¹⁴⁵ By default the field-set will be displayed surrounded by a border. Optionally you can add a title using the

142 <https://html.spec.whatwg.org/multipage/form-elements.html#the-select-element>

143 <https://html.spec.whatwg.org/multipage/form-elements.html#the-option-element>

144 <https://html.spec.whatwg.org/multipage/form-elements.html#the-optgroup-element>

145 <https://html.spec.whatwg.org/multipage/form-elements.html#the-fieldset-element>



`<legend>...</legend>` tag.¹⁴⁶

- `<textarea></textarea>` — This tag is used to create a multi-line text entry box, such as a comment section.¹⁴⁷ When defining it you should include the `rows` and `cols` attributes to define the size on the screen. For example:

```
<textarea name="message" rows=3 cols=60>Default  
Text</textarea>
```

Here is an example of form and input elements being used properly in an HTML document:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Using the Reflect Server Script</title>  
    <link href="usingreflect.css" rel="stylesheet" type="text/css">  
  </head>  
  <body>  
    <section>  
      <h2>Simple Form</h2>  
      <form action="http://www.syw2l.org/reflect.php" method="get">  
        <label for="usernamefield">Your Name:</label>  
        <input type="text" id="usernamefield" name="username" size=50  
          maxlength=50 value="Your user name">  
        <br>  
        <label for="agefield">Your Age:</label>  
        <input type="number" id="agefield" name="age" min=13  
          max=120 step=1>  
        <br>  
        <label for="weightfield">Your Weight (Kg):</label>  
        <input type="range" name="weight" id="weightfield" min=0  
          max=400 step=10 value=50  
        oninput="weightfieldout.value = weightfield.value">  
        <output id="weightfieldout">50</output>  
        <br>  
        <label for="humanfield">Check if you are a human:</label>  
        <input type="checkbox" id="humanfield" name="human">  
        <br>  
        <label for="colorselect">Your favorite color:</label>  
        <select id="colorselect" name="color">  
          <option value="r">Red</option>  
          <option value="g">Green</option>  
          <option value="b">Blue</option>  
          <option value="p">Pink</option>  
        </select>
```

146 <https://html.spec.whatwg.org/multipage/form-elements.html#the-legend-element>

147 <https://html.spec.whatwg.org/multipage/form-elements.html#the-textarea-element>




```
<br>
<fieldset>
  <legend>Are you a student:</legend>
  <input type="radio" name="student" id="student_k12"
value="k12">
  <label for="student_k12">K-12</label>
  <input type="radio" name="student" id="student_c"
value="c">
  <label for="student_c">College (any level)</label>
  <input type="radio" name="student" id="student_n"
value="n">
  <label for="student_n">Not currently</label>
</fieldset>
<br>
<label for="commenttext">Comments:</label>
<textarea id="commenttext" name="comment" rows=5 cols=40>
</textarea>
<br>
<input type="hidden" name="author" value="J.M.Reneau">
<input type="submit">
</form>
</section>
<footer>
  <p><small>&copy; 2020 The SYW2L Project
  <a href="http://www.syw2l.org">http://www.syw2l.org</a>
  </small></p>
</footer>
</body>
</html>
```

```
/* usingreflect.css - Sample CSS Sheet for formatting a Form */

label {
  display: inline-block;
  min-width: 10em;
}

fieldset {
  margin-left: 10em;
  display: grid;
  grid-template-columns: 40px 1fr;
}
```



Simple Form

Your Name:

Your Age:

Your Weight (Kg): 90

Check if you are a human:

Your favorite color:

Are you a student:

- K-12
- College (any level)
- Not currently

Comments:

© 2020 The SYW2L Project <http://www.syw2l.org>

Illustration 43: Sample Form with Elements

GET Data	
Key	Value
username	jreneau
age	50
weight	90
color	g
student	n
comment	beautiful form!!!
author	J.M.Reneau

POST Data	
Key	Value

Illustration 44: Sample Form - Data



Advanced Topics for Exploration

- Other HTML input types (datetime, hidden, ...)
- output
- progress
- meter

Vocabulary

- action
- checkbox
- Common Gateway Interface (CGI)
- date
- email
- fieldset
- form
- hidden
- input
- label
- method
- name
- number
- password
- radio
- range
- reset
- select
- submit
- tel
- text
- textarea
- type
- value



Exercises

Word Search

```
i e i m n d p e u t r a n g e C b l a a s l l m  
a e a e i e e p h t a u x I e o t n n l p s u g  
o n e t h u c e a m t o h p o m G a u e a a l t  
t m t h t c t i m s t c e e x m n a b l e b i t  
e w I o G r e t s i s a m e n o d t o t n p e r  
e c e d e v w e e e e w n o i n n t s o t r r l  
e n a i d d x x d C m f o x i t o o e d r t e  
l e i e t m t e s e e s r e G x m t i l e o s  
c s c d t r a a e u r n b e d a l s t r m e l G  
s h a d t m c f i e l d s e t t e b e a c e c n  
u e e f s t t t e l b s n f c e m l n d e e a t  
b e b c s n i d m t s i o m w a I a i s I C n  
m a l p k h o t h y h I i m l a i e m o d e i a  
i e h l b b n e m c t e e x u y l r e o n c t e  
t e l b d a o a a u s t t G p r o i m a h C  
e o C o p d x e l r p y f m I d d p o m i s r  
l C r a G l b e l b b m p p e n f o e m y d r n  
h t p w y I e d v a l u e a e t a a e e d d b  
n d t e x t a r e a c i e a v e d u a o i e g f  
d i a h t e n m n i e s a a o r c f e a n n e o  
b b t t p i e C r e s e t u h f i e a d p a e r  
i t G r e t t t r i e b y a p a e s t r u r h m  
m m n u m b e r p I l m d n v c n a s b t i r m  
y I u e t c e d n x t a b n e G d h i n t b o
```

CGI, Common Gateway Interface, action, checkbox, date, email, fieldset, form, hidden, input, label, method, name, number, password, radio, range, reset, select, submit, tel, text, textarea, type, value

Problems

eBook
Edition

