Forms

Why Forms?

The best known form on the web is probably the search box that sits right in the middle of Google's homepage.

Form Controls

ADDING TEXT:

Text input (single-line)
Used for a single line of text such as email addresses and names.

Ivy

Password input
Like a single line text box but it masks the characters entered.

•••••••

Text area (multi-line)
For longer areas of text, such as messages and comments.

Enter your comments...

MAKING CHOICES:

Radio buttons
For use when a user must select one of a number of options.

◉ Rock ○ Pop ○ Jazz

Checkboxes
When a user can select and unselect one or more options.

☑ iTunes ☐ Last.fm ☐ Spotify

Drop-down boxes
When a user must pick one of a number of options from a list.

iPod ▾

SUBMITTING FORMS:

Submit buttons
To submit data from your form to another web page.

Subscribe

Image buttons
Similar to submit buttons but they allow you to use an image.

SUBSCRIBE

UPLOADING FILES:

File upload
Allows users to upload files (e.g. images) to a website.

Browse...

Upload

How Forms Work

Form Structure

`<form>`

Form controls live inside a `<form>` element. This element should always carry the `action` attribute and will usually have a `method` and `id` attribute too.

action

Every `<form>` element requires an `action` attribute. Its value is the URL for the page on the server that will receive the information in the form when it is submitted.

chapter-07/form-structure.html **HTML**

```
<form action="http://www.example.com/subscribe.php"
  method="get">
  <p>This is where the form controls will appear.
    </p>
</form>
```

RESULT

This is where the form controls will appear.

method

Forms can be sent using one of two methods: `get` or `post`.

With the `get` method, the values from the form are added to the end of the URL specified in the `action` attribute. The `get` method is ideal for:

- short forms (such as search boxes)

- when you are just retrieving data from the web server (not sending information that should be added to or deleted from a database)

With the `post` method the values are sent in what are known as HTTP headers. As a rule of thumb you should use the `post` method if your form:

- allows users to upload a file

- is very long

- contains sensitive data (e.g. passwords)

- adds information to, or deletes information from, a database

```
<form action="http://www.example.com/login.php">
  <p>Username:
    <input type="text" name="username" size="15"
      maxlength="30" />
  </p>
</form>
```

RESULT

Username: [   ]

```
<form action="http://www.example.com/login.php">
  <p>Username:
    <input type="text" name="username" size="15"
      maxlength="30" />
  </p>
  <p>Password:
    <input type="password" name="password" size="15"
      maxlength="30" />
  </p>
</form>
```

RESULT

Username: Ivy

Password: •••••••

```html
<form>
    <label for="firstname">First name: </label>
    <input type="text" name="firstname"  required>
    <br>
    <label for="lastname">Last name: </label>
    <input type="text" name="lastname"  required>
    <br>
    <label for="email">email: </label>
    <input type="email" name="email"  required>
    <br>
    <label for="password">password: </label>
    <input type="password" name="password"  required>
    <br>
    <input type="submit" value="Login!">
</form>
```

`<input>`

`type="radio"`

Radio buttons allow users to pick just one of a number of options.

`name`

The name attribute is sent to the server with the value of the option the user selects. When a question provides users with options for answers in the form of radio buttons, the value of the name attribute should be the same for all of the radio buttons used to answer that question.

`value`

The `value` attribute indicates the value that is sent to the server for the selected option. The value of each of the buttons in a group should be different (so that the server knows which option the user has selected).

`checked`

The checked attribute can be used to indicate which value (if any) should be selected when the page loads. The value of this attribute is `checked`. Only one radio button in a group should use this attribute.

chapter-07/radio-button.html **HTML**

```
<form action="http://www.example.com/profile.php">
  <p>Please select your favorite genre:
  <br />
  <input type="radio" name="genre" value="rock"
    checked="checked" /> Rock
  <input type="radio" name="genre" value="pop" />
    Pop
  <input type="radio" name="genre" value="jazz" />
    Jazz
  </p>
</form>
```

RESULT

Please select your favorite genre:
○ Rock ○ Pop ● Jazz

Please note: Once a radio button has been selected it cannot be deselected. The user can only select a different option. If you are only allowing the user one option and want them to be able to deselect it (for example if they are indicating they agree to terms and conditions), you should use a checkbox instead.

Exercise

Create an HTML file named login_form.html.

Inside the HTML file, include the necessary code to create a login form with:

- A label and input field for the username.
- A label and password input field for the password.
- A submit button to log in.
- Ensure both the username and password fields are required for submission.
- Use the appropriate HTML structure and attributes for form elements.

```
<form action="http://www.example.com/profile.php">
  <p>Please select your favorite music service(s):
    <br />
    <input type="checkbox" name="service"
      value="itunes" checked="checked" /> iTunes
    <input type="checkbox" name="service"
      value="lastfm" /> Last.fm
    <input type="checkbox" name="service"
      value="spotify" /> Spotify
  </p>
</form>
```

RESULT

Please select your favorite music service(s):
☑ iTunes ☐ Last.fm ☐ Spotify

```
<form action="http://www.example.com/profile.php">
  <p>What device do you listen to music on?</p>
  <select name="devices">
    <option value="ipod">iPod</option>
    <option value="radio">Radio</option>
    <option value="computer">Computer</option>
  </select>
</form>
```

What device do you listen to music on?

✓ iPod
Radio
Computer

Multiple Select Box

```html
<form action="http://www.example.com/profile.php">
  <p>Do you play any of the following instruments?
    (You can select more than one option by holding
    down control on a PC or command key on a Mac
    while selecting different options.)</p>
  <select name="instruments" size="3"
    multiple="multiple">
    <option value="guitar" selected="selected">
      Guitar</option>
    <option value="drums">Drums</option>
    <option value="keyboard"
      selected="selected">Keyboard</option>
    <option value="bass">Bass</option>
  </select>
</form>
```

RESULT

Do you play any of the following instruments? (You can select more than one option by holding down control on a PC or command key on a Mac while selecting different options.)

```
Guitar
Drums
Keyboard
Bass
```

Exercise

1. Create an HTML file named **basic_form.html**.
2. Inside the HTML file, include the necessary code to create a form with:
   - Input fields for the user's first name, last name, email address, and age.
   - A dropdown menu for selecting the user's country.
   - Radio buttons for selecting the user's gender.
   - Checkboxes for selecting user interests (e.g., sports, music, reading).
   - A submit button to submit the form.
3. Use appropriate labels for each input field and group related fields together.

File Input Box

```
<form action="http://www.example.com/upload.php"
  method="post">
  <p>Upload your song in MP3 format:</p>
  <input type="file" name="user-song" /><br />
  <input type="submit" value="Upload" />
</form>
```

RESULT

Browse...

Upload

Submit Button

```html
<form action="http://www.example.com/subscribe.php">
  <p>Subscribe to our email list:</p>
  <input type="text" name="email" />
  <input type="submit" name="subscribe"
    value="Subscribe" />
</form>
```

RESULT

Subscribe to our email list:

[   ] Subscribe

Image button

```html
<form action="http://www.example.org/subscribe.php">
  <p>Subscribe to our email list:</p>
  <input type="text" name="email" />
  <input type="image" src="images/subscribe.jpg"
    width="100" height="20" />
</form>
```

Subscribe to our email list:

SUBSCRIBE

```
<form action="http://www.example.com/add.php">
  <button><img src="images/add.gif" alt="add"
    width="10" height="10" /> Add</button>
  <input type="hidden" name="bookmark"
    value="lyrics" />
</form>
```

RESULT

+ Add

<button>

The <button> element was introduced to allow users more control over how their buttons appear, and to allow other elements to appear inside the button.

This means that you can combine text and images between the opening <button> tag and closing </button> tag.

<input>

type="hidden"

This example also shows a hidden form control. These form controls are not shown on the page (although you can see them if you use the *View Source* option in the browser). They allow web page authors to add values to forms that users cannot see. For example, a web page author might use a hidden field to indicate which page the user was on when they submitted a form.

```
<label>Age: <input type="text" name="age" /></label>
<br/ >
Gender:
<input id="female" type="radio" name="gender"
  value="f">
<label for="female">Female</label>
<input id="male" type="radio" name="gender"
  value="m">
<label for="male">Male</label>
```

RESULT

Age: [   ]
Gender: ○ Female ○ Male

Grouping Form Elements

chapter-07/grouping-form-elements.html

```
<fieldset>
  <legend>Contact details</legend>
  <label>Email:<br />
  <input type="text" name="email" /></label><br />
  <label>Mobile:<br />
  <input type="text" name="mobile" /></label><br />
  <label>Telephone:<br />
  <input type="text" name="telephone" /></label>
</fieldset>
```

RESULT

Contact details

Email:

Mobile:

Telephone:

\<fieldset\>

You can group related form controls together inside the `<fieldset>` element. This is particularly helpful for longer forms.

Most browsers will show the `fieldset` with a line around the edge to show how they are related. The appearance of these lines can be adjusted using CSS.

\<legend\>

The `<legend>` element can come directly after the opening `<fieldset>` tag and contains a caption which helps identify the purpose of that group of form controls.

```
<form action="http://www.example.com/login/"
    method="post">
  <label for="username">Username:</label>
  <input type="text" name="username"
    required="required" /></title><br />
  <label for="password">Password:</label>
  <input type="password" name="password"
    required="required" />
  <input type="submit" value="Submit" />
</form>
```

RESULT

Username: [Ivy]
Password: [ ] (Submit)

Please fill out this field.

```
<form action="http://www.example.com/bookings/"
    method="post">
  <label for="username">Departure date:</label>
  <input type="date" name="depart" />
  <input type="submit" value="Submit" />
</form>
```

Departure date: 2011-06-27 ⟠ (Submit)

```
<form action="http://www.example.org/subscribe.php">
  <p>Please enter your email address:</p>
  <input type="email" name="email" />
  <input type="submit" value="Submit" />
</form>
```

Please enter your email address:

ivy| Submit

Please enter an email
address.

```
<form action="http://www.example.org/profile.php">
  <p>Please enter your website address:</p>
  <input type="url" name="website" />
  <input type="submit" value="Submit" />
</form>
```

Please enter your website address:

ivy   Submit

Please enter a URL.

```
<form action="http://www.example.org/search.php">
  <p>Search:</p>
  <input type="search" name="search" />
  <input type="submit" value="Search" />
</form>
```

Search:

| Thelonius ⊗ | Search |

```
<form action="http://www.example.org/search.php">
  <p>Search:</p>
  <input type="search" name="search"
    placeholder="Enter keyword" />
  <input type="submit" value="Search" />
</form>
```

Search:

| Enter keyword | Search |

Your Details:

Name: [_____]

Email: [_____]

Your Review:

How did you hear about us? [Google ▼]

Would you visit again?
○ Yes ○ No ○ Maybe

Comments:

[_____]

☑ Sign me up for email updates

[Submit review]