

## Focus on Web Design

The design of a form, such as the justification of the labels, the use of background colors, and even the order of the form elements can either increase or decrease the usability of a form. Visit some of the following resources to explore form design:

- Designing Efficient Web Forms: <https://www.smashingmagazine.com/2017/06/designing-efficient-web-forms/>
- Form Design Best Practices: <https://blog.hubspot.com/marketing/form-design>
- Best Practices for Mobile Form Design: <https://www.smashingmagazine.com/2018/08/best-practices-for-mobile-form-design/>

Feel free to search on your own and locate additional resources. Create a web page that lists the URLs of at least two useful resources along with a brief description of the information you found most interesting or valuable. Design a form on the web page that applies what you've just learned in your exploration of form design. Place your name in an e-mail link on the web page.

## Website Case Study Adding a Form

Each of the following case studies continues throughout most of the textbook. This chapter adds a page containing a form that invokes server-side processing to the websites.

### JavaJam Coffee Bar

See [Chapter 2](#) for an introduction to the JavaJam Coffee Bar case study. [Figure 2.32](#) shows a site map for the JavaJam site. Use the [Chapter 8](#) JavaJam website as a starting point for this case study.

You will create the new Jobs page that contains a form. You have three tasks in this case study:

1. Create a new folder for this JavaJam case study.
2. Modify the style sheet (javajam.css) to configure style rules for the new form.
3. Create the new Jobs page shown in [Figure 9.34](#).

---

**Figure 9.34** JavaJam Jobs page

---



## Hands-On Practice Case Study

**TASK 1: THE WEBSITE FOLDER.** Create a folder called javajam9. Copy all of the files from your Chapter 8 ☐ javajam8 folder into the javajam9 folder. Copy the herojobs.jpg and coffeecup.jpg files from the chapter9/starters folder in the student files.

**Task 2: Configure the CSS.** Modify the external style sheet (javajam.css). Review Figure 9.34 ☐ and the grid layout sketch in Figure 9.35 ☐. Notice how the text labels for the form controls are on the left side of the content area. Notice the empty space between each form control. When displaying on a narrow viewport, the display will be more pleasing if there is only one column, as shown in Figure 9.36 ☐. Open javajam.css in a text editor. Configure the CSS as follows:

1. Format the hero image. Code a selector above the media queries for an id named **herojobs** with 300px height. Set the background image to coffeecup.jpg. The background image will not repeat. Also configure background-size: 100% 100%.

2. Configure the single column display for narrow viewports using flexbox. Add the following CSS above the media queries to configure the form element selector as a flex container with one column, 1em left padding, and 80% width. Also set a .5em bottom margin on the input and textarea element selectors.

```
form { display: flex;
      flex-direction: column;
      padding-left: 1em; width: 80%; }
input, textarea { margin-bottom: .5em; }
```

3. Configure the two-column display with grid layout and format a larger hero image. Add CSS to the first media query to accomplish this.
  - a. Configure a form element selector. Set 40% width, grid display with 1em grid gap, and two columns (6em width and 1 fr width).
  - b. Configure an attribute selector for the submit button. Use the `grid-column` property to place this in the second column. Set width to 9em.
  - c. Configure a selector for an id named `herojobs`. Set the background image to `herojobs.jpg`.

---

**Figure 9.35** The grid layout sketch of the form

---

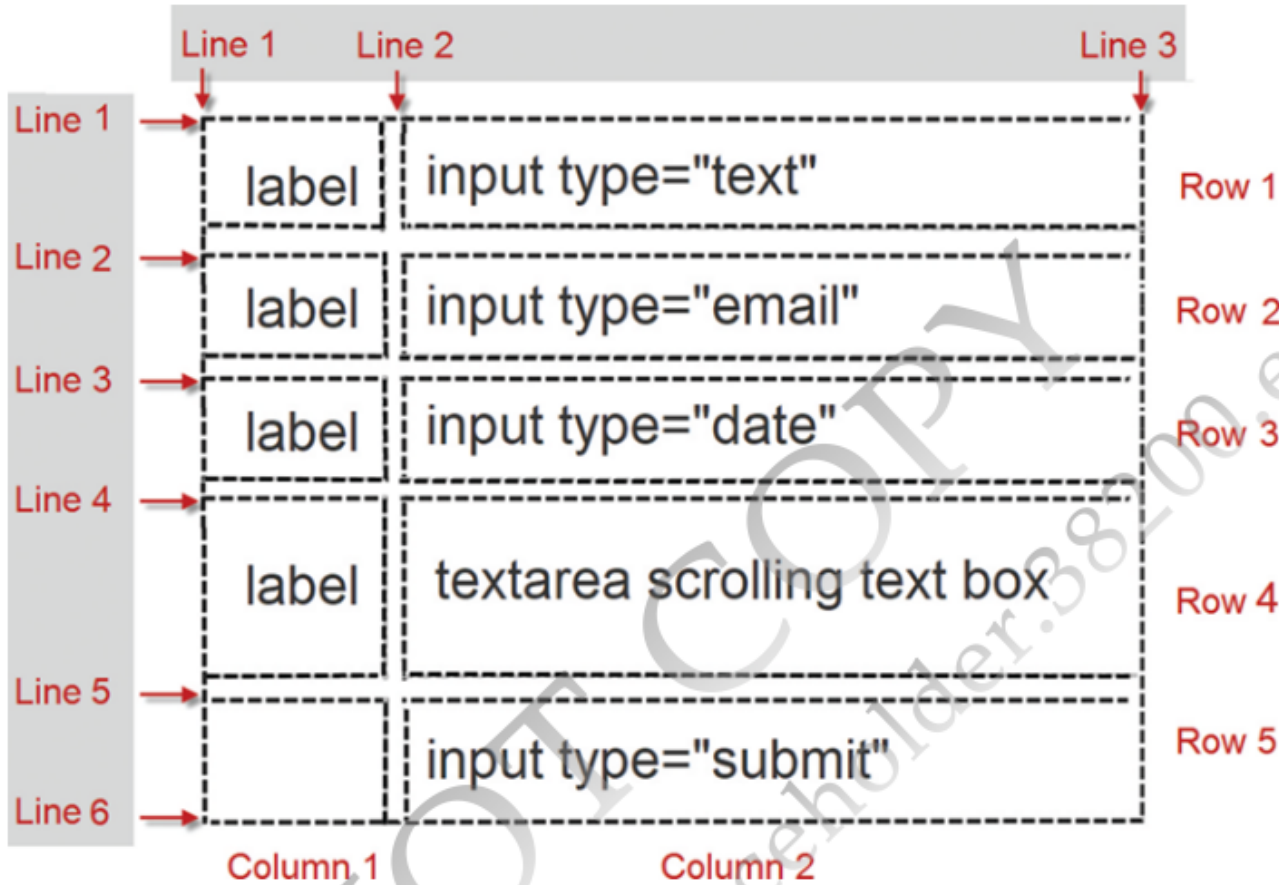




Figure 9.36 Single column form in a narrow viewport





# JavaJam Coffee Bar

- Home
- Menu
- Music
- Jobs



## Jobs at JavaJam

Want to work at JavaJam? Fill out the form below to start your application. All informaiton is required.

Information is required.

Name:

E-mail:

Start Date:

Experience:

Copyright © 2020 JavaJam Coffee Bar  
[yourfirstname@yourlastname.com](mailto:yourfirstname@yourlastname.com)

Save the javajam.css file.

**TASK 3: CREATE THE JOBS PAGE.** Use the Menu page as the starting point for the Jobs page. Launch a text editor and open menu.html. Save the file as jobs.html. Modify your jobs.html file to look similar to the Jobs page (shown in Figure 9.34) as follows:

1. Change the page title to an appropriate phrase.
2. The Jobs page will contain an h2, a paragraph, and a form in the main element. Assign the first div to an id named `herojobs`.
3. Edit the text within the h2 element to say "Jobs at JavaJam". Replace the text in the paragraph with the following: "Want to work at JavaJam? Fill out the form below to start your application. All information is required."
4. Delete the remaining menu-related content from the page: the table and the div assigned to the `flow` id.
5. Prepare to code the HTML for the form area. Begin with a form element that uses the post method and the action attribute to invoke server-side processing. Unless directed otherwise by your instructor, configure the action attribute to send the form data to <https://webdevbasics.net/scripts/javajam8.php>.

6. Configure the form control for the Name information. Create a label element that contains the text "Name:". Create a text box named myName. Configure the browser to require entry in the text box. Use the `for` attribute to associate the label element with the form control.
7. Configure the form control for the E-mail information. Create a label element that contains the text "E-mail:". Create e-mail address input form control named myEmail. Configure the browser to require entry in the text box. Use the `for` attribute to associate the label element with the form control.
8. Configure the form control for the date available to start work. Create a label element that contains the text "Start Date:" Create a calendar date-picker form control named myStart. Configure the browser to require the user to enter or select this information. Use the `for` attribute to associate the label element with the form control.
9. Configure the Experience area on the form. Create a label element that contains the text "Experience:". Create a textarea element named myExperience with `rows` set to 2 and `cols` set to 20. Configure the browser to require entry in the textarea. Use the `for` attribute to associate the label element with the form control.
10. Configure the submit button. Code an input element with `type="submit"` and `value="Apply Now"`. Assign the input element to an id named mySubmit.
11. Code an ending tag on a blank line after the submit button.

Save your file and test your web page in a browser. It should look similar to the page shown in [Figure 9.34](#). If you resize the browser viewport to be narrower, the display should be similar to [Figure 9.36](#). If you are connected to the Internet, enter all the information and submit the form. This will send your form information to the server-side script configured in the form tag. A confirmation page that lists the form information and their corresponding names will be displayed.

Next, submit the form with missing information or only a partial e-mail address. Depending on the browser's level of HTML5 support, the browser may perform form validation and display an error message. [Figure 9.37](#) shows the Jobs page rendered in the Firefox browser with an incorrectly formatted e-mail address.

---

**Figure 9.37 The Jobs page shows an error message**

---



The screenshot shows a web browser window with the title "JavaJam Coffee Bar Jobs". The website has a dark sidebar with a coffee cup icon and links for Home, Menu, Music, and Jobs. The main content area features a large image of a latte with a heart design. Below the image is a yellow section titled "Jobs at JavaJam" with a form for job applications. The form includes fields for Name, E-mail, Start Date, and Experience, each with a text input box. A red box highlights the E-mail field, which contains the text "sp". A message "Please enter an email address." is displayed below the E-mail field. The Experience field contains the text "One year barista in college cafeteria". An "Apply Now" button is at the bottom of the form. A copyright notice at the bottom of the page reads "Copyright © 2020 JavaJam Coffee Bar" and "yourintname@yourintname.com".

JavaJam Coffee Bar Jobs

JavaJam Coffee Bar

Home  
Menu  
Music  
Jobs

**Jobs at JavaJam**

Want to work at JavaJam? Fill out the form below to start your application. All information is required.

Name:

E-mail:

Start Date:

Experience:

Copyright © 2020 JavaJam Coffee Bar  
yourintname@yourintname.com

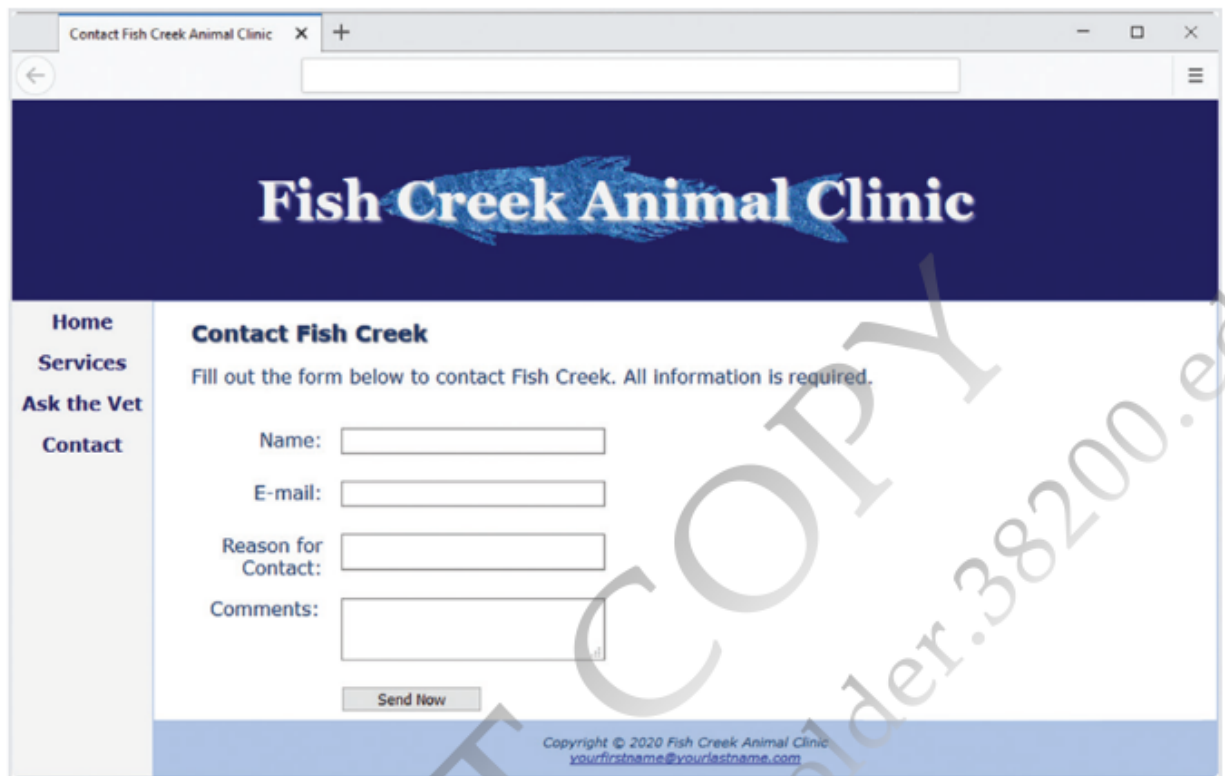
## Fish Creek Animal Clinic

See [Chapter 2](#) for an introduction to the Fish Creek Animal Clinic case study. [Figure 2.36](#) shows a site map for Fish Creek. Use the [Chapter 8](#) Fish Creek website as a starting point for this case study. You will create the new Contact page that contains a form. You have three tasks in this case study:

1. Create a new folder for this Fish Creek case study.
2. Modify the fishcreek.css style sheet to configure style rules for the new form.
3. Create the new Contact page shown in [Figure 9.38](#).

**Figure 9.38** Fish Creek Contact page





The screenshot shows a web browser window titled "Contact Fish Creek Animal Clinic". The page has a dark blue header with the text "Fish Creek Animal Clinic" in white. Below the header is a navigation menu with links: "Home", "Services", "Ask the Vet", and "Contact". The "Contact" link is highlighted. The main content area is titled "Contact Fish Creek" and contains a form with the following fields: "Name:", "E-mail:", "Reason for Contact:", and "Comments:". Each field has a corresponding input box. Below the form is a "Send Now" button. The footer of the page contains the text "Copyright © 2020 Fish Creek Animal Clinic" and "yourfirstname@yourlastname.com".

## Hands-On Practice Case Study

**TASK 1: THE WEBSITE FOLDER.** Create a folder called fishcreek9. Copy all of the files from your **Chapter 8** fishcreek8 folder into the fishcreek9 folder.

**TASK 2: CONFIGURE THE CSS.** Modify the external style sheet (fishcreek.css). Open fishcreek.css in a text editor. Review **Figure 9.38** and the grid layout sketch in **Figure 9.39**. Notice how the text labels for the form controls are on the left side of the content area but contain right-aligned text. Notice the empty vertical space between each form control. When displaying on a narrow viewport, the display will be more pleasing if there is only one column, as shown in **Figure 9.40**. Configure CSS as follows:

1. Configure the single column display for narrow viewports using flexbox. Add the following CSS above the media queries to configure the form element selector as a flex container with one column, 1em left padding, and 80% width. Also set a .5em bottom margin on the input and textarea element selectors.

```
form { display: flex;
      flex-direction: column;
      padding-left: 1em; width: 80%; }
input, textarea { margin-bottom: .5em; }
```

- 2. Configure the two-column display with grid layout. Add CSS to the first media query.
  - a. Configure a form element selector. Set 40% width, grid display with 1em grid gap, and two columns (6em width and 1 fr width).
  - b. Configure an attribute selector for the submit button. Place this in the second column using the `grid-column` property. Set width to 9em.
  - c. Configure a label element selector with right alignment.

Figure 9.39 The grid layout sketch of the form

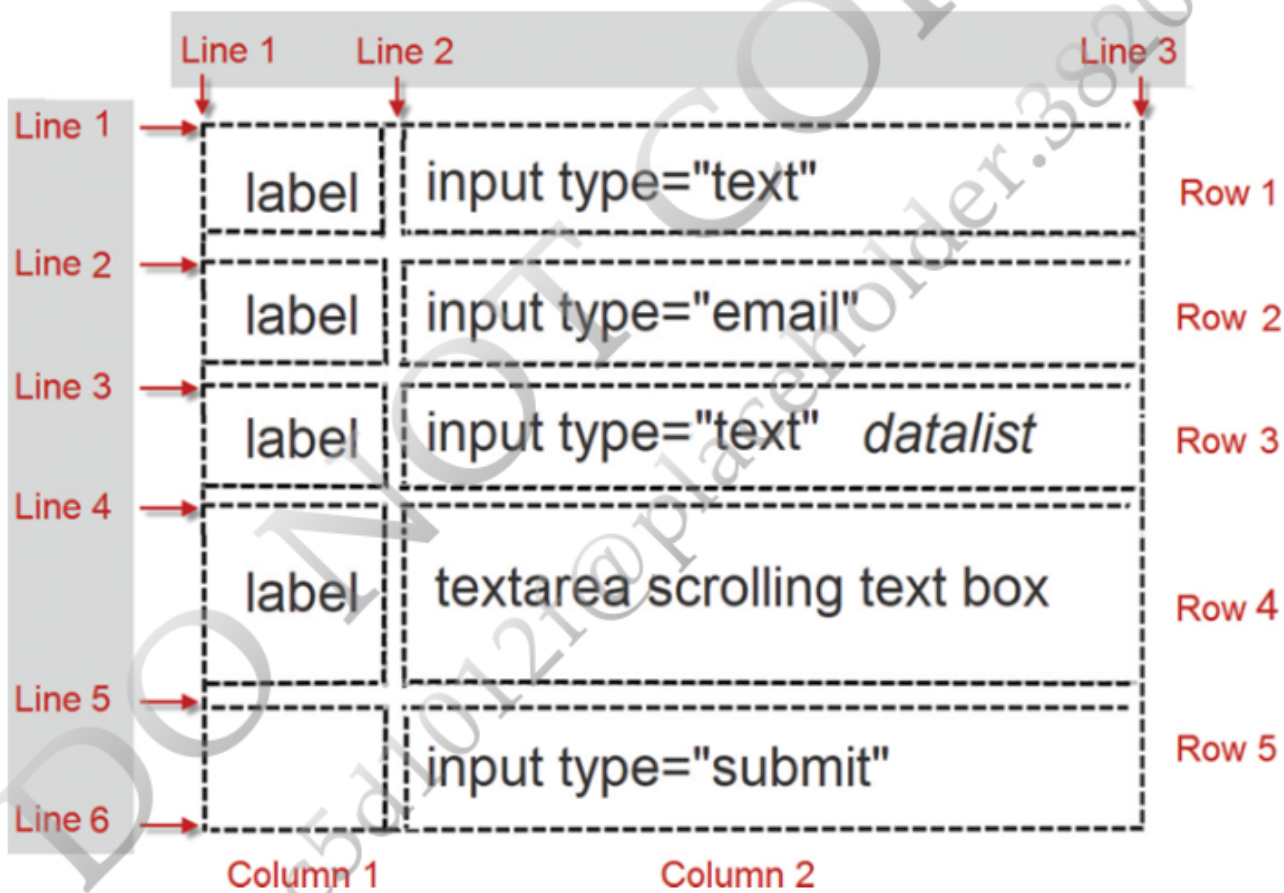


Figure 9.40 Single-column form in a narrow viewport

# Fish Creek Animal Clinic

Home

Services

Ask the Vet

Contact

## Contact Fish Creek

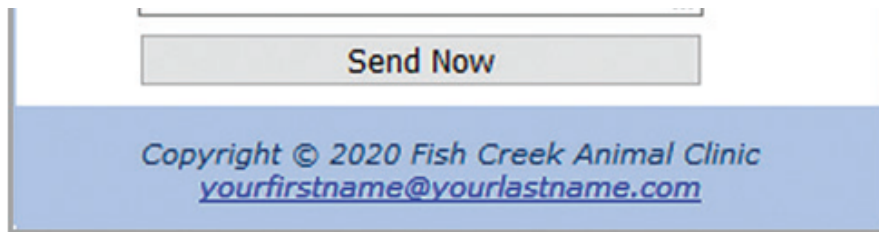
Fill out the form below to contact Fish Creek. All information is required.

Name:

E-mail:

Reason for Contact:

Comments:



Save the fishcreek.css file.

**TASK 3: CREATE THE CONTACT PAGE.** Use the Ask the Vet page as the starting point for the Contact page. Launch a text editor and open askvet.html. Save the file as contact.html. Modify your contact.html file to look similar to the Contact page (shown in [Figure 9.36](#)) as follows:

1. Change the page title to an appropriate phrase.
2. Delete the description list, img element, and div assigned to the id `flow`.
3. Replace the text within the h2 element with following: "Contact Fish Creek".
4. Replace the text in the paragraph with the following:  
Fill out the form below to contact Fish Creek. All information is required.
5. Prepare to code the HTML for the form area. Begin with a form element that uses the post method and the action attribute to invoke server-side processing. Unless directed otherwise by your instructor, configure the action attribute to send the form data to <https://webdevbasics.net/scripts/fishcreek.php>.
6. Configure the form control for the Name information. Create a label element that contains the text "Name:". Create a text box named myName. Use the `for` attribute to associate the label element with the form control. Configure the browser to require entry in the text box.
7. Configure the form control for the E-mail information. Create a label element that contains the text "E-mail:". Create e-mail address input form control named myEmail. Use the `for` attribute to associate the label element with the form control. Configure the browser to require entry in the text box.