CSC230 Assignment – Week 2

1. Complete all the Exercises (Exercise 9-1, 9-2, and 9-3) in "The Great Form Control Roundup" Section.

EXERCISE 9-1. Starting the pizza order form

Here's the scenario. You are the web designer in charge of creating an online pizza ordering form for Black Goose Bistro. The owner has handed you a sketch (FIGURE 9-9) of the form's content. There are sticky notes from the programmer with information about the script and variable names you need to use.

Your challenge is to turn the sketch into a functional form. I've given you a head start by creating a bare-bones document with text content and minimal markup and styles. This document, *pizza.html*, is available online at *learningwebdesign.com/5e/materials*. The finished form is also provided.

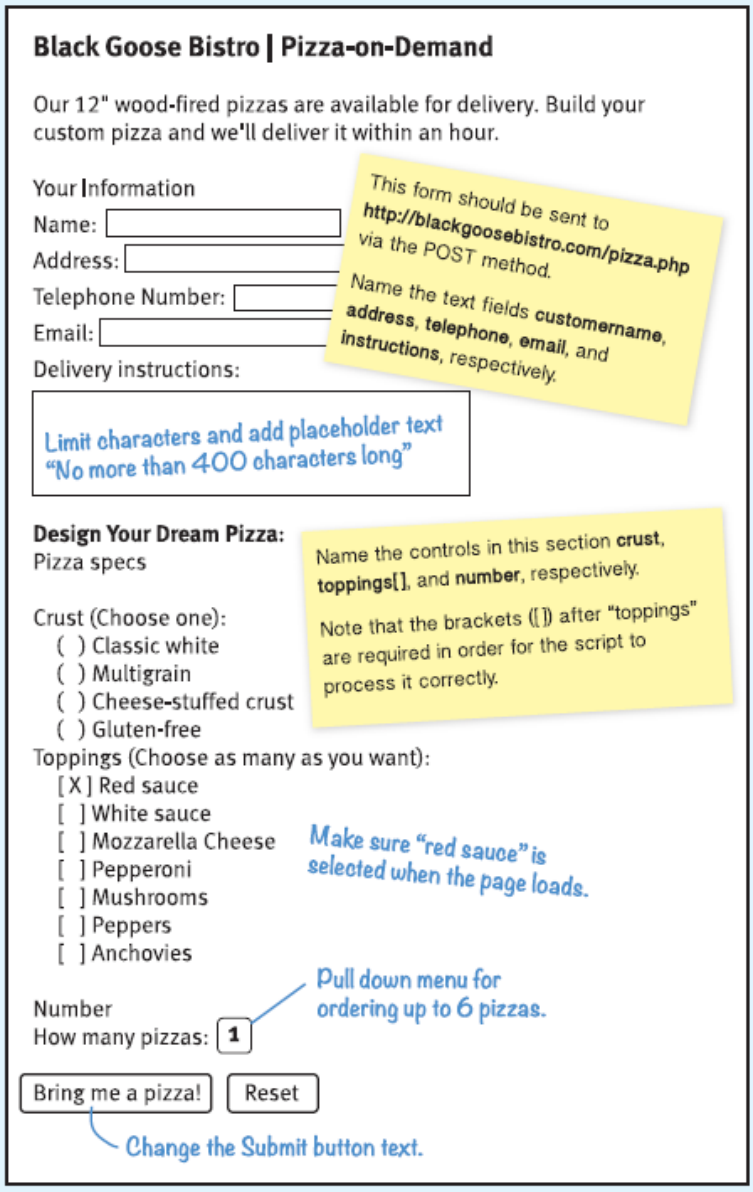

FIGURE 9-9. A sketch of the Black Goose Bistro pizza ordering form.

EXERCISE 9-1. Continued

1. Open the file *pizza.html* in a text editor.
2. The first thing we'll do is put everything after the intro paragraph into a **form** element. The programmer has left a note specifying the **action** and the **method** to use for this form. The resulting **form** element should look like this (keep it on one line):

```
<form action="http://www.blackgoosebistro.com/
pizza.php" method="POST">
…
</form>
```

3. In this exercise, we'll work on the "Your Information" section of the form. Start with the first four short text-entry form controls that are marked up appropriately as an unordered list. Here's the first one; you insert the other three:

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

   HINTS: Choose the most appropriate input type for each entry field. Be sure to name the input elements as specified in the programmer's note.

4. After "Delivery instructions:" add a line break and a multiline text area. Because we aren't writing a style sheet for this form, use markup to make it four rows long and 60 characters wide (in the real world, CSS is preferable because it gives you more fine-tuned control):

```
<li>Delivery instructions:<br>
<textarea name="instructions" rows="4" cols="60"
maxlength="400" placeholder="No more than 400
characters long"></textarea></li>
```

5. We'll skip the rest of the form for now until we get a few more controls under our belt, but we can add the submit and reset buttons at the end, just before the **</form>** tag. Note that they've asked us to change the text on the submit button.

```
<p><input type="submit" value="Bring me a
pizza!"><input type="reset"></p>
```

6. Now, save the document and open it in a browser. The parts that are finished should generally match FIGURE 9-2. If they don't, then you have some more work to do.

Once the document looks right, take it for a spin by entering some information and submitting the form. You should get a response like the one shown in FIGURE 9-10. Yes, *pizza.php* actually works, but sorry, no pizzas will be delivered.

FIGURE 9-10. You should see a response page like this if your form is working. The pizza description fields will be added in later exercises, so they will return "empty" for now.

EXERCISE 9-2. Adding radio buttons and checkboxes

The next section of the Black Goose Bistro pizza ordering form uses radio buttons and checkboxes for selecting pizza options. Open the *pizza.html* document and follow these steps:

1. In the "Design Your Dream Pizza" section, there are lists of Crust and Toppings options. The Crust options should be radio buttons because pizzas have only one crust. Insert a radio button before each option. Follow this example for the remaining crust options:

```
<li><input type="radio" name="crust" value="white"> Classic white</li>
```

2. Mark up the Toppings options as you did the Crust options, but this time, the **type** should be **checkbox**. Be sure the variable name for each is **toppings[]**, and that the "Red sauce" option is preselected (**checked**), as noted on the sketch.
3. Save the document and check your work by opening it in a browser to make sure it looks right; then submit the form to make sure it's functioning properly.

EXERCISE 9-3.
Adding a menu

The only other control that needs to be added to the order form is a pull-down menu for selecting the number of pizzas to have delivered.

1. Insert a **select** menu element with the option to order between 1 and 6 pizzas:

```
<p>How many pizzas:
<select name="pizzas"
size="1">
    <option>1</option>
<-- more options here -->
  </select>
</p>
```

2. Save the document and check it in a browser. You can submit the form, too, to be sure that it's working. You should get the "Thank You" response page listing all of the information you entered in the form.

Congratulations! You've built your first working web form. In EXERCISE 9-4, we'll add markup that makes it more accessible to assistive devices.

Blackstone Bistro | Pizza-on-Demand

Our 12" wood-fired pizzas are avaiable for delivery. Build your custom pizza and we'll deliver it within an hour.

Your Information

Name:
Address:
Telephone Number:
Email:
Delivery instructions:
No more than 400 characters long

Design your dream pizza:

Pizza specs

Crust *(Choose one)*:

- ○ Classic white
- ○ Multigrain
- ○ Cheese-stuffed crust
- ○ Gluten-free

Toppings *(Choose as many as you want)*

- ☑ Red sauce
- ☐ White sauce
- ☐ Mozzarella Cheese
- ☐ Pepperoni
- ☐ Mushrooms
- ☐ Peppers
- ☐ Anchovies

Number

How many pizzas: 1

Bring me a pizza! Reset

2. Complete Exercise 11-2 on page 244.

EXERCISE 11-2. Your first style sheet

Open *cooking.html* in a text editor. In the **head** of the document you will find that I have set up a **style** element for you to type the rules into. The **style** element is used to embed a style sheet in an HTML document. To begin, we'll simply add the small style sheet that we just looked at in this section. Type the following rules into the document, just as you see them here:

```
<style>
h1 {
  color: green;
}
p {
  font-size: large;
  font-family: sans-serif;
}
</style>
```

Save the file, and take a look at it in the browser. You should notice some changes (if your browser already uses a sans-serif font, you may see only a size change). If not, go back and check that you included both the opening and closing curly bracket and semicolons. It's easy to accidentally omit these characters, causing the style sheet not to work.

Now we'll edit the style sheet to see how easy it is to write rules and see the effects of the changes. Here are a few things to try.

IMPORTANT: Remember that you need to save the document after each change in order for the changes to be visible when you reload it in the browser.

- Make the **h1** element "gray" and take a look at it in the browser. Then make it "blue". Finally, make it "orange". (We'll run through the complete list of available color names in **Chapter 13, Colors and Backgrounds**.)
- Add a new rule that makes the **h2** elements orange as well.
- Add a 100-pixel left margin to paragraph (**p**) elements by using this declaration:

```
margin-left: 100px;
```

Remember that you can add this new declaration to the existing rule for **p** elements.

- Add a 100-pixel left margin to the **h2** headings as well.
- Add an orange, 1-pixel border to the bottom of the **h1** element by using this declaration:

```
border-bottom: 1px solid orange;
```

- Move the image to the right margin, and allow text to flow around it with the **float** property. The shorthand **margin** property shown in this rule adds zero pixels of space on the top and bottom of the image and 12 pixels of space on the left and right of the image (the values are mirrored in a manner explained in **Chapter 14, Thinking Inside the Box**):

```
img {
  float: right;
  margin: 0 12px;
}
```

When you are done, the document should look something like the one shown in FIGURE 11-4.

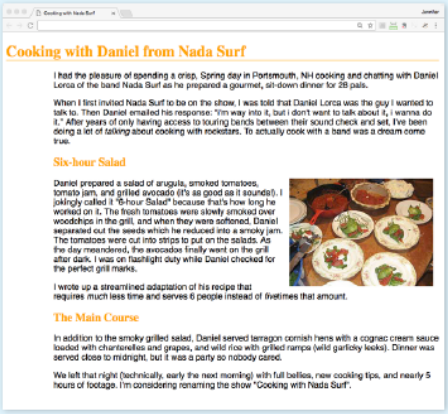

FIGURE 11-4. The article after we add a small style sheet. Not beautiful—just different.

3. Write style rules to show the following color.

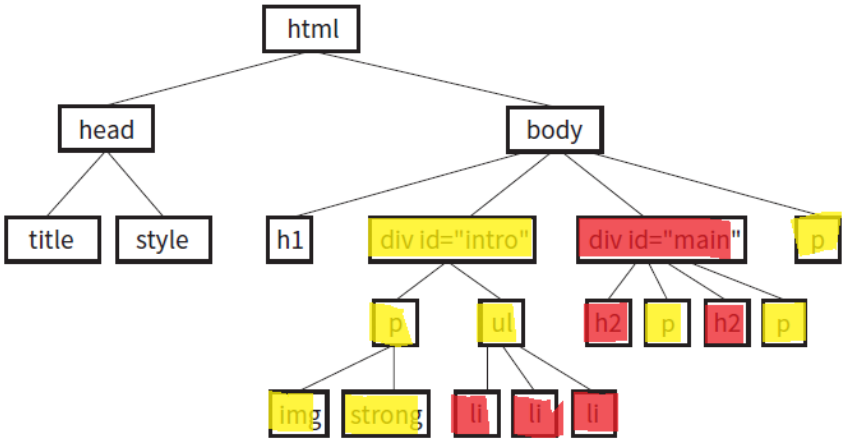

FIGURE 11-13. The document structure of a sample document.