```
/home/jharvard/public html/students/ already existed; chmodding to 711
 2.
   TF: Peter Nore, Student: crystalhsieh7, Project: project0
 3.
4.
 5.
    /home/jharvard/public_html/students/crystalhsieh7 already existed; chmodding to 711
    /home/jharvard/public html/students/crystalhsieh7/banished temp files/ already existed; chmodding to 711
8. downloading project0 by crystalhsieh7executing ... cd /home/jharvard/public_html/students/crystalhsieh7; git clone
    git@repo.cs50.net:crystalhsieh7/project0.git; git checkout submit;
9.
    /banished_temp_files dir, chmodding to 711
10.
11. /banished_temp_files/firstpart.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/firstpart.php~
    /banished_temp_files/clearcart.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/clearcart.php~
    /banished_temp_files/checkout.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/checkout.php~
    /banished_temp_files/remove.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/remove.php~
15. /banished_temp_files/foodtype.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/foodtype.php~
    /banished temp files/cart.php~ banishing to /home/jharvard/public html/students/crystalhsieh7/banished temp files/cart.php~
     dir, chmodding to 711
17.
18. /banished temp files/quantitychange.php~ banishing to
    /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/quantitychange.php~
19. /banished_temp_files/index.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/index.php~
20. /banished_temp_files dir, chmodding to 711
21. /banished temp files/addtocart.php~ banishing to /home/jharvard/public html/students/crystalhsieh7/banished temp files/addtocart.php~
22. /banished_temp_files/receipt.php~ banishing to /home/jharvard/public_html/students/crystalhsieh7/banished_temp_files/receipt.php~
23. /home/jharvard/public_html/students dir, chmodding to 711
24. /project0 dir, chmodding to 711; found project directory
                                                                Scope (satisfying spec's requirements):
25. /project0/ending.php with permissions 0600
26. /project0 dir, chmodding to 711
27. /project0/quantitychange.php with permissions 0600
    /project0/clearcart.php with permissions 0600
29. /project0/README.txt~ public file, changing to 644
                                                                Correctness (bug-free-ness):
30. /project0/pictures dir, chmodding to 711
31. /project0/pictures/three aces header.png public file, changing
32. /project0/pictures dir, chmodding to 711
33. /project0/pictures/three_aces_store.jpg public file, changing to 644
34. /project0/remove.php with permissions 0600
35. /project0/#three_aces_style.css# public file, changing to 644 Design (elegance of implementation)
    /project0/addtocart.php with permissions 0600
    /project0/foodtype.php with permissions 0600
   /project0/menu.xml with permissions 0600
    /project0/000_teaching_fellow_script_summary.txt public file, changing to 644
                                                                Style (pretty-printed code):
     dir, chmodding to 711
    /project0/receipt.php with permissions 0600
42. /project0/three_aces_style.css~ public file, changing to 644
43. /project0/README.txt public file, changing to 644
    /project0 dir, chmodding to 711
```

45. /project0/cart.php with permissions 0600 46. /project0/index.php with permissions 0600

- 47. public file, changing to 644
- 48. /project0/blueprint dir, chmodding to 711
- 9. /project0/blueprint/plugins dir, chmodding to 711
- 50. /project0/blueprint/plugins/rtl dir, chmodding to 711
- 1. /project0/blueprint/plugins/rtl/readme.txt public file, changing to 644
- 52. /project0/blueprint/plugins/rtl dir, chmodding to 711
- 33. /project0/blueprint/plugins/rtl/screen.css public file, changing to 644
- 54. /project0/blueprint dir, chmodding to 711
- 55. /project0/blueprint/plugins/link-icons dir, chmodding to 711
- 56. /project0/blueprint/plugins/link-icons/readme.txt public file, changing to 644
- 57. /project0/blueprint/plugins/link-icons/icons dir, chmodding to 711
- 58. /project0/blueprint/plugins/link-icons/icons/visited.png public file, changing to 644
- 59. /project0/blueprint/plugins/link-icons/icons/email.png public file, changing to 644
- 60. /project0/blueprint/plugins/link-icons/icons/xls.png public file, changing to 644
- 61. /project0/blueprint/plugins/link-icons dir, chmodding to 711
- 62. /project0/blueprint/plugins/link-icons/icons/doc.png public file, changing to 644
- 63. /project0/blueprint/plugins/link-icons/icons/external.png public file, changing to 644
- 64. /project0/blueprint/plugins/link-icons/icons dir, chmodding to 711
- 65. /project0/blueprint/plugins/link-icons/icons/lock.png public file, changing to 644
- 66. /project0/blueprint/plugins/link-icons/icons/pdf.png public file, changing to 644
- 67. /project0/blueprint/plugins/link-icons/icons/feed.png public file, changing to 644
- 68. /project0/blueprint/plugins/link-icons/icons/im.png public file, changing to 644
- 69. /project0/blueprint/plugins/link-icons dir, chmodding to 711
- 70. /project0/blueprint/plugins/link-icons/screen.css public file, changing to 644
- 71. /project0/blueprint/plugins/buttons dir, chmodding to 711
- 72. /project0/blueprint/plugins/buttons/readme.txt public file, changing to 644
- 73. /project0/blueprint/plugins/buttons/icons dir, chmodding to 711
- 74. /project0/blueprint/plugins/buttons/icons/cross.png public file, changing to 644
- 75. /project0/blueprint/plugins/buttons dir, chmodding to 711
- 76. /project0/blueprint/plugins/buttons/icons dir, chmodding to 711
- 77. /project0/blueprint/plugins/buttons/icons/key.png public file, changing to 644
- 78. /project0/blueprint/plugins/buttons/icons/tick.png public file, changing to 644
- 79. /project0/blueprint/plugins/buttons dir, chmodding to 711
- 80. /project0/blueprint/plugins/buttons/screen.css public file, changing to 644
- 81. /project0/blueprint/plugins dir, chmodding to 711
- 82. /project0/blueprint/plugins/fancy-type dir, chmodding to 711
- 83. /project0/blueprint/plugins/fancy-type/readme.txt public file, changing to 644
- 84. /project0/blueprint/plugins/fancy-type dir, chmodding to 711
- 85. /project0/blueprint/plugins/fancy-type/screen.css public file, changing to 644
- 86. /project0/blueprint/src dir, chmodding to 711
- 87. /project0/blueprint/src/reset.css public file, changing to 644
- 88. /project0/blueprint/src/ie.css public file, changing to 644
- 89. /project0/blueprint/src/typography.css public file, changing to 644
- 90. /project0/blueprint dir, chmodding to 711
- 91. /project0/blueprint/src/forms.css public file, changing to 644
- 92. /project0/blueprint/src dir, chmodding to 711
- 93. /project0/blueprint/src/grid.css public file, changing to 644
- 94. /project0/blueprint/src/print.css public file, changing to 644

```
95. /project0/blueprint/src/grid.png public file, changing to 644
 96. /project0/blueprint/ie.css public file, changing to 644
 97. /project0 dir, chmodding to 711
 98. /project0/blueprint dir, chmodding to 711
 99. /project0/blueprint/print.css public file, changing to 644
100. /project0/blueprint/screen.css public file, changing to 644
101. /project0/firstpart.php with permissions 0600
102. /project0/checkout.php with permissions 0600
103. /project0/three_aces_style.css public file, changing to 644
104. dir, chmodding to 711
                                                                                                   correct url!
105. local path to dir: /home/jharvard/public_html/students/crystalhsieh7/project0
106. relative loc: /project0
107. url: http://192.168.56.50/~jharvard/students/crystalhsieh7/project0
108. http://192.168.56.50/~jharvard/students/crystalhsieh7/project0
109.
110. simulated local url: http://localhost/~jharvard/project0
```

111. summary: http://192.168.56.50/~jharvard/students/crystalhsieh7/project0/000_teaching_fellow_script_summary.txt;

The project setup is good. Next time, make a file in your project root called .gitignore, and put "*~" in it, which will keep it from tracking the changes to temp files.

Computer Science S-75: Building Dynamic Websites Harvard Summer School Summer 2011

Spend some time thinking about how best to represent this menu as XML, keeping these goals in mind: It must be easy for someone less technical than you to make changes to the menu. The XML should be straightforward to read and alter. You must somehow keep track of each item's category, name, price(s), size(s), and description, if any. You should avoid duplication of data. Just because Three Aces sells Tomato & Cheese pizzas in two sizes, that doesn't mean "Tomato & Cheese" needs to appear twice in your file! Your model should be extensible. If Three Aces eventually decides to sell medium pizzas, they shouldn't need to call you! Before deciding on a model, though, best to read on so that you know how your XML will be used. The overall design and aesthetics of this site are ultimately up to you, but we require that your site meet some requirements. **Feature Requirements** Your site must not display Three Aces's menu on one huge page but, rather, allow customers to browse the menu by category. It is fine to display multiple (but not all) categories per page. Spaghetti or Ziti and Home made Lasagna Ravioli or Manicotti, for instance, sound like they belong on the same page. You need not convert Three Aces's entire menu to XML, lest tedium take the fun out of design; three items per category suffice, so long as those triples make clear your overall design. However, we suspect you'll enjoy your site more if you input more than three items per category! Customers must be able to add items to a "shopping cart" whose contents persist until the customers check out or close their browsers. Customers must also be able to update quantities and remove items outright (without, e.q., having to change some item's quantity to 0). When customers follow some link to check out, they must be informed of their order's total cost and thanked for their order. Your site should perform rigorous error-checking. Under no circumstances should we be able to crash your site or induce unreasonable behavior. Letting us input negative quantities so that Three Aces owes us money is not, shall we say, reasonable. We will bang on your code and try to find faults; do not let us succeed. **Technical Requirements** You're welcome to develop your site on any computer using any IDE or text editor, but you must ensure that it works within the CS50 Appliance at a URL of http://localhost/~jharvard/project0/ when uploaded (as via SCP or SFTP) to /home/jharvard/public html/project0/. Your menu must be called menu.xml and be chmod'd 600.

All PHP files must also be chmod'd 600.

Computer Science S-75: Building Dynamic Websites Harvard Summer School Summer 2011

remember to validate more then the front page!

1	HTML 4.01 Transitional, HTML 5, XHTML 1.0 String, or XHTML 1.0 Transitional (or XHTML) should be valid, as per http://validator.w3.org/, unless some feature of your site requires otherwise (for the sake of some browser); explain in HTML (or XHTML) comments any intentional invalidities. Your HTML (or XHTML) should also be as pretty-printed as possible. Your CSS need not be valid.
4 + -	Your PHP must be extensively commented and be as pretty-printed as possible.
NA	You may use a WYSIWYG editor to generate HTML (or XHTML) and/or CSS that you would like to use in your site.
+ -	If you integrate third-party CSS or JavaScript libraries ($e.g.$, YUI) into your project, cite their origin with comments.
4 + "	If you incorporate or adapt snippets of PHP code from the Web into your project (e.g., examples from php.net), cite the code's origins with comments.
- -	If you incorporate images from the Web into your project, cite the images' with comments.
	Your website must appear and behave the same on at least two major browsers, namely:
4	☐ Chrome 12 or higher
4 +	☐ Firefox 4 or higher
	☐ Internet Explorer 8 or higher
	□ Opera 11 or higher□ Safari 5 or higher
	□ Salati 5 of fligher

So long as your site meets the foregoing requirements, you are welcome to interpret this specification as you see fit. Imagine, perhaps, an ideal site for Three Aces. Then go implement that. Or, at least, as much as you can!

And don't forget about help.cs75.net!

Exit Interview.



Once done with your site, put together a readme in a file called README that lives in the same folder as the rest of your project.

Treat this readme as your opportunity not only to explain but to justify your design decisions. Tell us why you modeled your XML as you did. Tell us why you chose, say, select menus over radio buttons for some feature. Tell us with which two (or more) browsers we should evaluate your site. And give us an overall sense of how your site works (*e.g.*, tell us which files do what). But still be succinct; keep this readme to just a few paragraphs in length.

- 1. Crystal Hsieh
- 2. CS-75
- 3. README FILE

¥ .

5. Hello, my name is Crystal Hsieh, and this is my \$PizzaML project.

6.

7. Please use Mozilla Firefox and Google Chrome to view my project.

8.

9. My XML file is categorized by food type, like Pizzas or Calzones, then by the actual food items, and then by the prices. Because the food categories are already separated in the original Three Aces Menu, I thought I would stay true to the original design and create my XML document in the same fashion. Because there were two different sizes for some categories and I didn't want to repeat names, I separated them by price. The prices could be different for each different item, but the sizes "small" or "large" could be consistent between items, so "size" became an attribute. I also decided to combine all the pastas in order to create a more concise and straightforward menu. Because in an XML document not all the same tags always need the same attributes, I did not add a "size" attribute to the items with only one size. I tried to make my XML document extensible for the pizza owner to add a new items, change prices, or add descriptions with my XML document.

10.

11. When creating my menu, I tried using tables to organize all the different items. I thought that the navigation bar would make it very easy to browse through the site to get to the desired food item.

12.

13. In order to fully utilize the XML document, the PHP page foodtype.php helped create all the different categories. By taking information from the XML file and using SimpleXML, foodtype.php takes the desired food category and print all the food items and prices for that category. After choosing a specific food item, a customer can click on the price and be redirected to the addtocart.php site, which recognizes what name and price was given, and parse it into a key for the cart.php document. When using the site, the customer does not see the addtocart.php, but she is redirected straight from addtocart.php to cart.php. Addtocart.php transfers the important information like the name and price and inputs it in the table of the cart.

14.

15. Once at the cart, the customer can change the quantity and refresh the page in order to see the changed subtotal of the item. In addition, they can click "Remove," which will remove the item from the cart (or cart array) without having the customer set the quantity to zero. The quantity cannot be set to a negative value, or it will just be reset to 0. When the quantity value is changed, the customer is redirected to quantitychange.php, which identifies which food name and size it is, and changes the quantity of that particular item. The "remove" button also redirects to a remove.php file, which basically "unsets" the unwanted file from the "cart" table.

16.

17. After ordering the customer will go to checkout.php and fill out their contact information for their delivery food. Because this form contains a lot of information and I didn't want all the personal information in the URL, the input fields are transferred through method=post.

18.

19. Because this project was based on mostly PHP and XML, if a field was not filled out, then the page would redirect the customer to the main.php page. If this project was for Javascript, it could have been used to create a full form validation and prevented submit because of empty fields. Using the attribute max_length, I tried to validate the form by allowing only a certain number of digits/characters for the phone number, the zip code, or the abbreviated state letters. It helps to know you are thinking about this!

20.

21. After confirming the order, the website delivers a receipt (receipt.php), reviewing what was just purchased as well as confirming the personal information plugged in. Also, if a person ordered a 2-way combo, then they were able to choose two different toppings for their pizza as they checked out. If they did it correctly and chose only two toppings, they would receive a response at the end of the receipt confirming what they had chosen. If they did not, they would not receive a confirmation message. I used checked boxes for the toppings because it prevents people from choosing the same topping twice and allows for multiple check boxes to be checked.

22.

23. In order to efficiently create the different php files, I was inspired by the sections in cs75.tv and I tried to make separate HTML files or the beginning and end of each php document, so that they could just be linked to rather than copy and paste. This is what the "firstpart.php"

and "ending.php" are for.

- 24.
- 25. I hope you enjoy my site!
- 26.
- 27. Crystal Hsieh

This does meet the requirements, this is just a little design finesse that would have been nice





Total: \$5.50



Welcome to the Three Aces Menu!

Choose from a variety of options and place your order online!





key takeaway: the user wants to see the result of their actions and at no time should they be able to feel like they just "did something wrong."

... also, I appreciate the dynamic principles you used when building the site ...



Order Confirmed!

Thank you joe schmo for ordering from Three Aces Pizza!

Order Information:

joe schmo
234 westside
omaha, ok 02341
(234)234-2345
woeifn@weofin.com

Nice! This goes "above and beyond" the assignment spec. When you do get around to making forms more robust, though,



Credit Card Number:

1234-1234-1234-1

	Item	Size	Quantit	ty	Unit Price	Item	Subtotal
	Tomato and Cheese		3		\$5.50	\$16.5	60
	Total: \$16.50		r	Your not ha	n't do thi user sho ave to "p / form!	ould	
	hone Number		ŗ	oreg_	ething lik _replace ",\$str) w	("/[^	
pu	ıt it in (XXX)XXX - XXX	XX for	''' r	remo	ve all no er chara	n-	

This is a programming class, not a design class, but I would be remiss if I did not point out that there isn't any indication that when you click here



Menu	Item(s): Grinders	SMALL	LARGE
Pizzas	Hamburger	<u>\$4.50</u>	<u>\$4.95</u>
Special Pizzas	Cheeseburger	<u>\$4.75</u>	<u>\$5.75</u>
Pastas	Meatball	<u>\$4.75</u>	<u>\$5.75</u>
Calzones	Cheese Steak	<u>\$5.10</u>	> \$6.00
Wraps	Onion Steak	<u>\$5.20</u>	<u>\$6.20</u>
Sides	Special Steak	<u>\$5.40</u>	<u>\$6.40</u>
Salads	Pepper and Egg	<u>\$4.50</u>	<u>\$4.95</u>
Grinders	Ham and Egg	<u>\$4.75</u>	<u>\$5.75</u>
Special Dinners	Steak and Egg	<u>\$5.60</u>	<u>\$6.50</u>
View Cart			
Clear Cart			

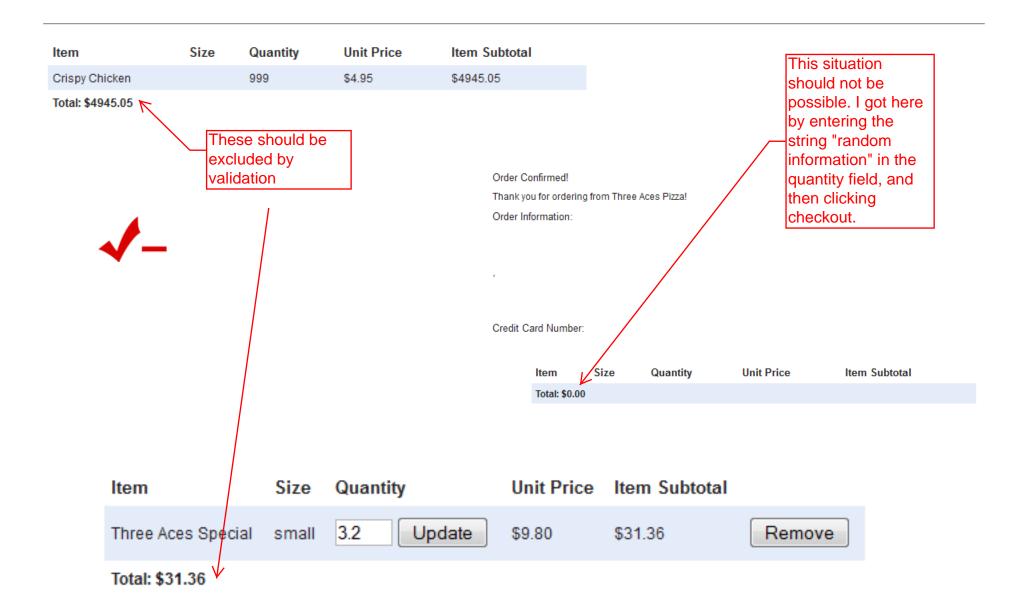
... it goes here, with no explanation of the significance of this page. It was not until I clicked on "view cart" and got the same thing that I realized this *was* the cart. What you need is an "affordance" [googleable buzz word]:

"In designing to accomodate visibility, each function and the method of operating it would be apparent ... by merely looking at it ... in analyzing interfaces, we should always ask how the user knows that an action is possible, and we should require that each visible feature provide a recognizable affordance." - Jef Raskin, The Humane Interface, p. 63



THE THREE ACES MENU

Menu		Item	Size	Quantity	Unit Price	Item Subtotal	
Pizzas		Onion Steak	small	1 Update	\$5.20	\$5.20	Remove
Special Pizzas		omon otoak	Oman	Ораско	\$ 0.20	ψ0.20	110111070
Pastas		Crispy Chicken		1 Update	\$4.95	\$4.95	Remove
Calzones		Mediterranean	large	1 Update	\$15.80	\$15.80	Remove
Wraps							
Sides		Grecian Supreme	small	1 Update	\$9.80	\$9.80	Remove
Salads		Cheese Steak	large	1 Update	\$6.00	\$6.00	Remove
Grinders		Total: \$41.75					
Special Dinners							
View Cart	w Cart Check-Out						
Clear Cart							



Check-Out!

```
1. <?
 2.
        //if there is a name, keep the value
 3.
        if(!isset($_GET["name"]))
 4.
 5.
            header("Location: index.php");
 7.
        $name = $_GET["name"];
 8.
9.
        //if there is a size, keep the value
        if(!isset($_GET["size"]))
10.
11.
12.
            header("Location: index.php");
13.
14.
        $size = $_GET["size"];
15.
16.
        //start a session to contain what the customer has ordered
17.
        session start();
18.
19.
        //create a key for the specific item so it can be identified in the array
20.
        $key = $name . "." . $size;
21.
22.
        //the quantity goes up by one!
        $ SESSION["cart"][$key] = $ SESSION["cart"] | $key] + 1;
23.
24.
25.
        //redirect the customer to their cart to see what they have ordered
26.
        header("Location: cart.php");
27. ?>
```

It seems like you might be missing something subtle about PHP here. It is not necessary to send the user an HTTP header to tell their browser to request a different PHP page from your own server. Instead, you should have just used require("cart.php");, which would have then inserted the result of executing cart.php immediately below this. The \$_SESSION variable would have been the same ...

```
1. <?
 2.
        $xml = new SimpleXMLElement(file_get_contents("menu.xml"));
 3.
        total = 0;
 4.
        session_start();
 5. ?>
 6. <? require_once("firstpart.php"); ?> <
7.
            8.
            <? //If there is a session started, then show the contents of the cart
               if(isset($_SESSION["cart"])){
9.
10.
                   $cart = $_SESSION["cart"];
11.
                   <!--create the heading for the table -->
           ?>
                   ItemSizeQuantityUnit PriceItem Subtotal
12.
13.
            <?
14.
                   $keys = array_keys($cart);
15.
                   //if there is a session, but an item was removed, then return an empty cart!
16.
                   if(count($keys) == 0){ ?>
17.
                       Your cart is now empty!
18.
                   <? header("Location: clearcart.php");</pre>
19.
20.
21.
                   foreach($keys as $key)
22.
23.
                       //take each item that was ordered and read it so it can be printed back onto the table
24.
                       $array = explode(".", $key);
                                                               PHPvocabulary++
25.
                       $name = $array[0];
26.
                       $size = $array[1];
27.
                       $quantity = $_SESSION["cart"][$key];
    your php is
28.
    correctly
29.
                       //if there is a quantity of 0, then the item should be erased!
    lindented and
30.
                       if($quantity == 0){
31.
                              unset($_SESSION['cart'][$key]);
    pretty printed
32.
    here; thanks!
33.
34.
                       //take the price and the possible size from the XML file
35.
                       $price = $xml->xpath("//food[@name='$name']/price[@size='$size']");
36.
                       if($size == null){
37.
                           $price = $xml->xpath("//food[@name='$name']/price");
38.
39.
                       //use the price from the XML and figure out the subtotal
40.
                       $uPrice = (float) $price[0];
41.
                       $itemSubtotal = $uPrice * $quantity;
42.
43.
                       $total += $itemSubtotal;
44.
45.
                       <!-- Print the name and the size of each item -->
46.
47.
                           <; print($name); ?>
48.
                           <? print($size); ?>
```

require_once
actually slows down
the server, because
it has to keep track
of which files have
been executed and
which files have
not. Try just
require().

```
49.
                          >
                              <!--if the customer wants to change the quantity, they can update it-->
50.
51.
                              <form method="post" action="quantitychange.php?name=<?print(htmlspecialchars($name));?>&size=<?echo $size;?>">
52.
                              <input type="text" value="<? print($quantity); ?>" size="3" maxlength="3" name="quantity">
53.
                              <input type="submit" value="Update" name="refresh">
54.
                              </form>
55.
                          56.
                          <!--print the unit price and the subtotal with a float up to two decimal points-->
57.
                          $<? printf("%0.2f", $uPrice); ?>
58.
                          $<? printf("%0.2f", $itemSubtotal); ?>
59.
60.
                          <!-- if someone wants to remove an item, they will be redirected to the remove.php page -->
61.
                              <form method="post" action="remove.php?name=<?print(htmlspecialchars($name));?>&size=<?echo $size;?>">
62.
                                  <input type="submit" value="Remove" name="<? echo $name ?>">
63.
                              </form>
64.
                          65.
                      66.
           <?
67.
68.
           ?>
69.
                   70.
                   <!-- the total will be printed with a span over 5 columns, with a float value up to two decimal points -->
                   <b>Total: $<? printf("%0.2f", $total); ?></b>
71.
72.
                   73.
                   <!-- when they are done ordering, they can finally click the button to check out! -->
74.
                   <a href="checkout.php"><input type="submit" value="Check-Out!">
75.
               <?
76.
77.
               else
78.
79.
           ?>
80.
                   <!-- when there is no items in the cart, it defaults to "Your cart is empty!" -->
81.
                   Your cart is empty!
                   82.
83.
           <? } ?>
84.
85.
86. <? require_once("ending.php"); ?>
                                                          Modular code!
```

```
1. <?
 2.
       $xml = new SimpleXMLElement(file_get_contents("menu.xml"));
3.
       total = 0;
 4.
       $combo = 0;
 5.
       session_start();
 6.
       //access the cart session to know what was ordered
7.
       $cart = $_SESSION["cart"];
8. ?>
9. <? require_once("firstpart.php"); ?>
10.
11.
       You have currently ordered:
       12.
13.
       14.
           ItemSizeQuantityUnit PriceItem Subtotal
15.
           <?
16.
                  $keys = array_keys($cart);
17.
                  foreach($keys as $key)
18.
19.
                      //take the array of items and separates the name and size from the key
20.
                      $items = explode(".", $key);
21.
                      $name = $items[0];
                                                                                        A little bit brittle:
22.
                      $size = $items[1];
                                                                                       what if the owner
23.
                      //if there is a 2-way combo order, record it
24.
                      if($name == '2-way Combo'){
                                                                                        changes the name?
25.
                          $combo++;
                                                                                        At the very least,
26.
27.
                      //same as the cart table, this code shows what was ordered
                                                                                        this might warrant a
                      $quantity = $_SESSION["cart"][$key];
28.
                                                                                        comment in
29.
                      $price = $xml->xpath("//food[@name='$name']/price[@size='$size']");
                                                                                        menu.xml
30.
31.
                      if($size == null){
32.
                          $price = $xml->xpath("//food[@name='$name']/price");
33.
                      $uPrice = (float) $price[0];
34.
35.
                      $itemSubtotal = $uPrice * $quantity;
36.
37.
                      $total += $itemSubtotal;
38.
                      ?>
39.
                      40.
41.
                          <; print($name); ?>
                          <; print($size); ?>
42.
43.
                          <? print($quantity); ?>
44.
                          $<? printf("%0.2f", $uPrice); ?>
                          $<? printf("%0.2f", $itemSubtotal); ?>
45.
                      46.
47.
                  <?
48.
```

```
49.
                   50.
       51.
52.
       <?
53.
       //if there was a combo order, a checkbox list is created for the customer to choose two toppings
54.
       if($combo > 0){ ?>
55.
       <form method="post" action="receipt.php">
56.
57.
       for(\$i = 0; \$i < \$combo; \$i++) \{ ?>
58.
       You have selected a 2-way Combo! <br/> Please select your two desired toppings!
59.
               //this has a loop to show all the different toppings that can be chosen
60.
61.
               $foodlist = $xml->xpath("foodtype[@name='Pizzas']/food"); ?>
                      <? foreach($foodlist as $topping){</pre>
62.
                          $topname = $topping['name'];
63.
64.
                          if($topname == '2-way Combo'){
65.
                              break;
66.
67.
                      ?>
68.
                      <!--checkboxes list is created with the foreach loop -->
                          <input type="checkbox" name="topping[]" value = "<?print($topname)?>">
69.
70.
                          <label for="<? print($topname) ?>"><? echo $topname ?></label><br/>>
71.
72.
                      <? } ?>
73.
                      <br/>
74.
       <?
75.
       }else{ ?>
76.
77.
       <form method="post" action="receipt.php">
       <?}?>
78.
79.
       <!--a form is created for the customer to give their delivery information -->
       Fill out your contact information and you will complete your order!
80.
81.
       <br/>
       First Name
82.
83.
        <input type="text" size="20" maxlength="20" name="fname">
84.
       <br/>
85.
       Last Name
86.
        <input type="text" size="20" maxlength="20" name="lname">
       <br/>
87.
       Phone Number
88.
89.
       put it in (XXX)XXX-XXXX form
       <input type="text" size="20" maxlength="13" name="phone">
90.
91.
92.
       Address:
       <input type="text" size="20" maxlength="20" name="address">
93.
       <br/>
94.
95.
       City:
96.
        <input type="text" size="20" maxlength="20" name="city">
```

```
97.
         <br/>
98.
         >
99.
         State
100.
         <input type="text" size="3" maxlength="2" name="state">
101.
102.
         <input type="text" size="7" maxlength="5" name="zipcode">
103.
         104.
         Email Address
105.
         <input type="text" size="30" maxlength="30" name="email">
106.
         Credit Card Number
107.
108.
         <!--although this is not a very secure way to keep the credit card number,
109.
            it is a simple way to implement the form without a database or way to
            hide the number-->
110.
111.
         <input type="text" size="20" maxlength="16" name="credit">
112.
         <br/>
113.
         <!--submit the form and complete purchase-->
114.
         <input type="submit" value="Confirm Order" name="confirm">
115.
         </form>
116.
117. <? require_once("ending.php"); ?>
```

```
1. <?
2. session_start();
3.    //a session must be started before it can destroy a session
4. unset($_SESSION["cart"]);
5.    //access cart session and destroy it
6. session_destroy();
7.    //now the cart is cleared, return customer back to home page
8. header("Location: index.php");
9. ?>
```

while this approach works, you might consider putting this in a function next time:

```
function clearCart() {
    session_start();
    //a session must be started before it can destroy a session
    unset($_SESSION["cart"]);
    //access cart session and destroy it
    session_destroy();
    //now the cart is cleared, return customer back to home page
    header("Location: index.php");
}
```

- 1. <!--closing tags to end the html documents -->
- 2. </div>
- 3. </div>
- 4. </body>
- 5. </html>

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
 3. <html xmlns="http://www.w3.org/1999/xhtml">
 4.
        <head>
 5.
            <title>The Three Aces Menu</title>
            <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
 6.
 7.
            <!-- Used CSS Framework from Blueprint -->
 8.
            <link rel="stylesheet" href="blueprint/screen.css" type="text/css" media="screen" />
            <link rel="stylesheet" href="blueprint/print.css" type="text/css" media="print" />
 9.
10.
            <!-- Added my own stylesheet to the Framework -->
11.
            <link href="three_aces_style.css" type="text/css" rel="stylesheet"/>
12.
        </head>
13.
        <!-- This is the first part of every page in the site -->
14.
15.
            <div class="container">
16.
                <div id="header" class="span-23">
17.
                <h1>The Three Aces Menu</h1>
18.
                <!-- the header here, which is implemented with CSS, was made by me in Paint -->
19.
                </div>
20.
                <div class="span-4">
21.
                ul id="navigation">
22.
                <!-- Navigation Bar is created from the categories given from the XML documentation -->
23.
                   <a href="index.php" shape="rect">Menu</a>
24.
                   <?
25.
                   $xml = new SimpleXMLElement(file_get_contents("menu.xml"));
26.
                   $foodtypes = $xml->xpath("foodtype");
27.
                   foreach($foodtypes as $foodtype) { ?>
28.
                       <a href="foodtype.php?foodtype=<? print($foodtype['name']); ?>"><? print($foodtype["name"]); ?></a>
29.
                   <? } ?>
                   <a href="cart.php" shape="rect">View Cart</a>
30.
31.
                   <a href="clearcart.php" shape="rect">Clear Cart</a>
32.
                33.
                </div>
                <div class="span-16">
34.
35.
                <!--main info-->
```



```
1. <?
       $xml = new SimpleXMLElement(file_get_contents("menu.xml"));
2.
3.
       $foodchosen = $_GET["foodtype"];
4.
       $foodlist = $xml->xpath("foodtype[@name='$foodchosen']/food");
5.
       //In order to create this document, I tried to better understand the php concepts
6.
       //by watching the CS-75 sections from cs75.tv, so some of my code is inspired from
7.
       //the section. This link is for all the food categories - the foodchosen will help
       //determine which page it is!
8.
9. ?>
10. <? require_once("firstpart.php"); ?>
11.
12.
          13.
          <!-- Since some items do not have specific sizes, I had to differentiate what kind of columns to use -->
14.
          <? if($foodchosen != 'Pastas' && $foodchosen != 'Calzones' && $foodchosen != 'Wraps' && $foodchosen != 'Special Dinners'){?>
15.
              Item(s): <? print($foodchosen); ?>
16.
              SMALL
                                                                                  This information should be loaded
17.
              LARGE
                                                                                  dynamically or should be a part of your
18.
          <? }else{ ?>
                                                                                  data structure. If the owner changes
19.
              Item(s): <? print($foodchosen); ?>
20.
              PRICE
                                                                                  the names, it will break the headers ...
21.
          <? } ?>
22.
          <!--here all the prices were inputted from the XML document -->
23.
           <? foreach($foodlist as $food) {</pre>
              $fname = $food['name'];?>
24.
25.
              26.
                 <; print($fname);
27.
                        //if there is a description, then post it under the name
28.
                        if($food['desc'] != null){?>
                        <br/>
29.
                        30.
31.
                        <? print $food['desc']; ?>
32.
                        33.
                     <? }?>
34.
                 35.
                 <!--adding all the prices to each food item-->
36.
                 <? foreach($food->price as $price) { ?>
37.
                    ?></a>
38.
                 <? } ?>
39.
              40.
41.
          <? } ?>
42.
          43.
44. <? require_once("ending.php"); ?>
```

```
1. <!-- This is the Home Page for the site! -->
 2. <? require_once("firstpart.php"); ?>
 3.
            <div id="welcome">
 4.
            <h1>Welcome to the Three Aces Menu </h1>
 5.
            <!--Welcome Menu -->
 6.
            <q>>
 7.
            Choose from a variety of options and place your order online!
 8.
            <!--When searching Three Aces Pizza on Googla, I found this picture at
9.
10.
            http://www.wickedlocal.com/cambridge/news/business/x426332345/Harvard-eats-up-Three-Aces-Pizza?photo=0#axzz1Sd25Sncl
11.
12.
            <img id="storefront" src="pictures/three_aces_store.jpg" alt="Three Aces Pizza Place"/><br/>c"/>
            <!--I found their old address from a website, so I put it on here-->
13.
            14.
15.
            Three Aces
16.
            17.
            >
18.
            <br/>
19.
            1613 Massachusetts Ave <br/>
20.
            Cambridge, MA 02139 <br/>
                                                                                  This is at the top of most pages, so you could make it
21.
            Btwn Mellen & Everett St <br/>
22.
            more modular by having an index.php that does
23.
                                                                                  something like this:
            617 491-2884 <br/>
24.
25.
            617 491-2889 <br/>
                                                                                  //logic about $page to load
26.
            27.
            </div>
                                                                                  ///
28.
                                                                                  require("firstpart.php");
29. <? require_once("ending.php"); ?> <
                                                                                  require($page);
                                                                                  require("ending.php");
```

```
1. <?xml version="1.0" ?>
 2.
    <menu>
3.
        <foodtype name="Pizzas">
 4.
             <food name="Tomato and Cheese">
 5.
                 <price type="small">5.50</price>
 6.
                 <price type="large">9.75</price>
 7.
             </food>
8.
             <food name="Onions">
9.
                 <price size="small">6.85</price>
10.
                 <price size="large">10.85</price>
11.
             </food>
             <food name="Peppers">
12.
13.
                 <price size="small">6.85</price>
14.
                 <price size="large">10.85</price>
15.
             </food>
16.
             <food name="2-way Combo">
                 <price size="small">7.95</price>
17.
18.
                 <price size="large">11.80</price>
19.
             </food>
20.
             <food name="Extra Cheese">
                 <price size="small">1.25</price>
21.
22.
                 <price size="large">1.85</price>
23.
             </food>
24.
        </foodtype>
25.
        <foodtype name="Special Pizzas">
26.
             <food name="Three Aces Special">
27.
                 <price size="small">9.80</price>
                 <price size="large">15.80</price>
28.
29.
             </food>
             <food name="Mediterranean" desc="Sliced Tomatoes, Olives, Spinach, Fresh Garlic, Mozzarella and Feta Cheese">
30.
31.
                 <price size="small">9.80</price>
32.
                 <price size="large">15.80</price>
33.
             </food>
             <food name="Grecian Supreme" desc="Grilled Chicken, Feta Cheese, Tomatoes, Kalamata Olives">
34.
35.
                 <price size="small">9.80</price>
36.
                 <price size="large">15.80</price>
37.
             </food>
        </foodtype>
38.
39.
        <foodtype name="Pastas">
             <food name="Spaghetti With Sauce">
40.
41.
                 <price>5.40</price>
             </food>
42.
43.
             <food name="Spaghetti With Sausage">
44.
                 <price>6.45</price>
45.
             </food>
             <food name="Spaghetti With Veal">
46.
47.
                 <price>6.45</price>
48.
             </food>
```

```
49.
             <food name="Ziti With Sauce">
50.
                 <price>5.40</price>
51.
             </food>
52.
             <food name="Ziti With Sausage">
53.
                 <price>6.45</price>
54.
             </food>
55.
             <food name="Ziti With Meatball">
56.
                 <price>6.45</price>
57.
             </food>
58.
             <food name="A La Three Aces" desc="Sausage, Mushrooms, Bacon and Ham, Topped with Sauce and Mozzarella Cheese">
59.
                 <price>7.25</price>
             </food>
60.
61.
             <food name="Lasagna Ravioli With Sauce">
62.
                 <price>6.25</price>
63.
             </food>
64.
             <food name="Lasagna Ravioli With Chicken Cutlet">
65.
                 <price>7.25</price>
66.
             </food>
67.
             <food name="Lasagna Ravioli With Mushrooms">
68.
                 <price>7.25</price>
69.
             </food>
70.
             <food name="Manicotti With Meatball">
71.
                 <price>7.25</price>
72.
             </food>
73.
             <food name="Manicotti With Veal">
74.
                 <price>7.25</price>
75.
             </food>
             <food name="Manicotti With Veggie">
76.
77.
                 <price>7.25</price>
             </food>
78.
79.
        </foodtype>
80.
        <foodtype name="Calzones">
81.
             <food name="Vegetarian">
82.
                 <price>7.35</price>
            </food>
83.
84.
             <food name="Sausage">
85.
                 <price>7.35</price>
86.
             </food>
             <food name="Chicken Cutlet">
87.
88.
                 <price>7.35</price>
89.
            </food>
             <food name="Meatball">
90.
91.
                 <price>7.35</price>
92.
             </food>
93.
        </foodtype>
94.
        <foodtype name="Wraps">
95.
             <food name="Turkey Club Wrap" desc="Turkey, Cheese, Lettuce, Tomato, Onion, Mayo">
96.
                 <price>4.95</price>
```

```
97.
              </food>
 98.
              <food name="Chicken Cobb" desc="Grilled Chicken, Bacon, Cheese, Lettuce, Tomatop, Onion, Honey Mustard">
99.
                  <price>4.95</price>
100.
              </food>
101.
              <food name="Crispy Chicken" desc="Chicken Finger, Lettuce, Tomato, Onion, Honey Mustard">
102.
                  <price>4.95</price>
103.
             </food>
104.
         </foodtype>
105.
         <foodtype name="Sides">
106.
              <food name="Onion Rings">
107.
                  <price size="small">2.60</price>
108.
                  <price size="large">2.95</price>
109.
              </food>
              <food name="French Fries">
110.
111.
                  <price size="small">2.25</price>
112.
                  <price size="large">2.65</price>
113.
              </food>
              <food name="Spicy Fries">
114.
115.
                  <price size="small">2.60</price>
116.
                  <price size="large">2.95</price>
117.
              </food>
118.
         </foodtype>
         <foodtype name="Salads">
119.
120.
              <food name="Garden Salad">
121.
                  <price size="small">3.50</price>
122.
                  <price size="large">4.50</price>
123.
              </food>
124.
              <food name="Greek Salad">
125.
                  <price size="small">4.50</price>
126.
                  <price size="large">5.50</price>
127.
              </food>
128.
              <food name="Kabob Salad" desc="with grilled Chicken and feta cheese">
129.
                  <price size="small">5.45</price>
130.
                  <price size="large">6.45</price>
             </food>
131.
132.
         </foodtype>
133.
         <foodtype name="Grinders">
134.
              <food name="Hamburger">
135.
                  <price size="small">4.50</price>
136.
                  <price size="large">4.95</price>
137.
             </food>
138.
              <food name="Cheeseburger">
139.
                  <price size="small">4.75</price>
140.
                  <price size="large">5.75</price>
141.
             </food>
142.
              <food name="Meatball">
143.
                  <price size="small">4.75</price>
144.
                  <price size="large">5.75</price>
```

```
145.
             </food>
146.
             <food name="Cheese Steak">
147.
                  <price size="small">5.10</price>
148.
                  <price size="large">6.00</price>
149.
             </food>
150.
              <food name="Onion Steak">
151.
                  <price size="small">5.20</price>
152.
                  <price size="large">6.20</price>
             </food>
153.
154.
              <food name="Special Steak">
155.
                  <price size="small">5.40</price>
156.
                  <price size="large">6.40</price>
157.
             </food>
158.
              <food name="Pepper and Egg">
159.
                  <price size="small">4.50</price>
160.
                  <price size="large">4.95</price>
161.
             </food>
162.
              <food name="Ham and Egg">
163.
                  <price size="small">4.75</price>
164.
                  <price size="large">5.75</price>
165.
             </food>
166.
             <food name="Steak and Egg">
167.
                  <price size="small">5.60</price>
168.
                  <price size="large">6.50</price>
169.
             </food>
170.
         </foodtype>
171.
         <foodtype name="Special Dinners">
172.
              <food name="Chicken Wing Dinner">
173.
                  <price>7.25</price>
174.
             </food>
175.
              <food name="Gyro Plate">
176.
                  <price>7.25</price>
177.
178.
              <food name="Steak Tips Dinner" desc="Served with side salad and french fries">
179.
                  <price>8.25</price>
180.
             </food>
181.
              <food name="Fish and Chips Dinner" desc="Served with side salad, french fries, and tartar sauce">
182.
                  <price>7.35</price>
183.
             </food>
184.
         </foodtype>
185. </menu>
```

```
1. <?
2.
        //get the name
3.
        if(!isset($_GET["name"]))
 4.
 5.
            header("Location: index.php");
 6.
7.
        $name = $_GET["name"];
8.
        //get the size
9.
        if(!isset($_GET["size"]))
10.
11.
            header("Location: index.php");
12.
13.
        $size = $_GET["size"];
14.
        //get the quantity, as long as one is inputted
15.
        if(!isset($_POST["quantity"]))
16.
17.
            header("Location: index.php");
18.
19.
20.
        $quantity = $_POST["quantity"];
21.
22.
         //if the quantity is less than zero, it will automatically become 0;
23.
         if($quantity < 0){</pre>
24.
            $quantity = 0;
25.
26.
27.
        $key = $name . "." . $size;
28.
29.
        session_start();
30.
31.
32.
33.
        //changing the field quantity of this item to the new quantity inputted
34.
        $_SESSION["cart"][$key] = $quantity;
35.
36.
        header("Location: cart.php");
37.
38. ?>
```

```
1. <?
  2.
                    session_start();
  3.
                    $cart = $_SESSION["cart"];
  4.
                    $total = 0;
  5. ?>
  6. <? require_once("firstpart.php"); ?>
  7.
                    //after filling out the form, the variables will store all their inputted fields
  8.
  9.
                    $fname = $_POST["fname"];
10.
                    $lname = $_POST["lname"];
                    $phone = $_POST["phone"];
11.
                    $address = $_POST["address"];
12.
13.
                    $city = $_POST["city"];
14.
                    $state = $_POST["state"];
15.
                    $zipcode = $_POST["zipcode"];
16.
                    $email = $_POST["email"];
17.
                    $credit = $_POST["credit"];
18.
                    //normally one would not put the credit card number so easily as a post, but we do not have a database to hash this or any quick ways to
          secure this information
19.
                    if(isset($_POST["topping"])){
20.
                             $topping = $_POST["topping"];
21.
22.
                    //if any of the fields are not filled, they will be redirected to the home page
                    if(isset($_POST["fname"]) && isset($_POST["lname"])&&isset($_POST["phone"])&&isset($_POST["address"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POST["city"])&&isset($_POS
          $_POST["state"])&&isset($_POST["zipcode"])&&isset($_POST["email"])&&isset($_POST["credit"]))
24.
25.
                    ?>
26.
                    Order Confirmed!
27.
                    echo "Thank you ".$fname." ".$lname." for ordering from Three Aces Pizza!"
28.
29.
30.
                    Order Information:
31.
                    <? echo $fname." ".$lname ?>
32.
33.
                    <br/>
34.
                    <? echo $address ?>
35.
                    <br/>
36.
                    <? echo $city.", ".$state." ".$zipcode ?>
37.
                    <br/>
                    <? echo $phone ?>
38.
39.
                    <br/>
                    <? echo $email ?>
 40.
41.
                    <br/>
42.
                    >
43.
                    Credit Card Number:
                    44.
 45.
                    <? echo $credit ?>
 46.
                    <br/>
```

```
47.
       <!--here is the table to show what the customer had ordered-->
48.
49.
           ItemSizeQuantityUnit PriceItem Subtotal
50.
           <?
51.
                  $keys = array_keys($cart);
52.
                  foreach($keys as $key)
53.
54.
                      $items = explode(".", $key);
                      $name = $items[0];
55.
56.
                      $size = $items[1];
57.
                      $quantity = $_SESSION["cart"][$key];
58.
59.
60.
                      $price = $xml->xpath("//food[@name='$name']/price[@size='$size']");
61.
                      //in case there is no sizes, there is another way to get the prices
62.
                      if($size == null){
                          $price = $xml->xpath("//food[@name='$name']/price");
63.
64.
65.
                      //the rest of this code is similar to the cart code - both displaying the same information
66.
                      //but it confirms the customer's order
67.
                      $uPrice = (float) $price[0];
68.
                      $itemSubtotal = $uPrice * $quantity;
69.
70.
                      $total += $itemSubtotal;
71.
                      ?>
72.
                      73.
74.
                          <? print($name);?>
75.
                          <? print($size); ?>
76.
                          <? print($quantity); ?>
77.
                          $<? printf("%0.2f", $uPrice); ?>
78.
                          $<? printf("%0.2f", $itemSubtotal); ?>
79.
                      <?
80.
81.
82.
83.
                  <b>Total: $<? printf("%0.2f", $total); ?></b>
84.
       85.
       <?
       //if there are two toppings that were chosen, display those choices
86.
87.
       if(isset($_POST["topping"]) && count($topping) == 2){
           $first = $topping[0];
88.
89.
           $second = $topping[1];
90.
       ?>
91.
           >
92.
           Also, for your 2-way combo, you ordered these toppings:
93.
           <? echo $first." and ".$second ?>
94.
```

```
<? }?>
95.
96.
         >
 97.
         <?
98.
         //after the receipt is given, the session is destroyed because the purchase was made
99.
         unset($_SESSION["cart"]);
100.
         session_destroy();
101.
         }else{
102.
         //if a customer didn't fill it out correctly, they are redirected to the main page
103.
         header("Location: index.php");
104.
105.
         ?>
106. <? require_once("ending.php"); ?>
```

```
1. <?
 2.
        session_start();
3.
 4.
        if(!isset($_GET["name"]))
 5.
 6.
            header("Location: index.php");
 7.
 8.
        $name = $_GET["name"];
9.
10.
        if(!isset($_GET["size"]))
11.
12.
            header("Location: index.php");
13.
14.
        $size = $_GET["size"];
15.
16.
        $key = $name . "." . $size;
17.
18.
        unset($_SESSION["cart"][$key]);
19.
20.
21.
22.
        header("Location: cart.php");
23. ?>
24.
25.
```

```
 body{

 2.
         background-color: #151B54;
 3.
 4. #header{
 5.
         padding-top: 140px;
 6.
         overflow: hidden;
 7.
        background-image: url(pictures/three_aces_header.png);
 8.
        background-repeat: no-repeat;
 9.
        height: 0px;
        margin: 5px 0 0 0;
10.
11. }
12.
13. th {
         font-size: 110%;
14.
15.
16.
17. #desc{
18.
         font-style: oblique;
19.
         font-size: 90%;
20.
21.
22. a:link{
         font-size: 110%;
23.
24.
25.
26. #navigation{
                                                                   You should really
27.
        width: 12em;
28.
        border-right: 1px solid;
                                                                   cite your source
         text-decoration: none;
29.
                                                                   here
30.
        color: black;
31.
        margin: 10px 0 0 0;
32.
        list-style: none;
        margin: 0;
33.
34.
        padding: 0;
35. }
36.
37. #navigation li {
        margin: 0;
38.
        border-top: 1px none;
39.
40.
41.
    #navigation li a {
43.
         display: block;
44.
        background-color:#E8E8E8;
        border-top: 1px solid;
45.
         font-size: 130%;
46.
47.
        border-bottom: 1px solid;
         text-decoration: none;
48.
```

```
49.
        padding: 2px 2px 2px 0.5em;
50.
        color:
                    #41383C;
51. }
52. #navigation .cart a{
53.
        background-color: #ADDFFF;
54. }
55.
56. }
57. .span-16{
58.
        background-color: #FFFFFF;
        color: #2B60DE;
59.
60.
        margin: 20px 0px 0px 15px;
61.
        padding: 25px 40px 50px;
62.
63. h1 {
64.
        color: #2B60DE;
65.
66. p{
67.
        font-size: 110%;
        margin: 0px;
68.
69.
        padding: 6px 0px;
70. }
71. #help {
72.
        font-style: italic;
73.
        color: gray;
        font-size: 90%;
74.
75. }
76. #storefront{
        float: right;
77.
78.
        clear: left;
79.
        height: 180px;
        margin: 20px 0px 0px 0px;
80.
81. }
82. .span-23{
83.
        background-color: #6D7B8D;
84. }
85. .container{
        background-color: #FFFFFF;
86.
        padding: 20px 0px 35px 35px;
87.
88.
89. table, #welcome{
        padding: 10px 10px 10px 50px;
90.
91. }
92.
93. #threeaces{
94.
        font-weight: bold;
95.
        font-size: 190%;
        padding: 0px;
96.
```

```
97. }
98. .empty{
99. padding: 15px;
100. font-size: 150%;
101. }
```



Jump To: Notes and Potential Issues Validation Output

Errors found while checking this document as XHTML 1.0 Strict!

Result:	12 Errors, 14 warning(s)					
Source:	<pre><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> </pre>					
Encoding:	utf-8	(detect automatically)				
Doctype:	XHTML 1.0 Strict	(detect automatically)				
Root Element:	html					
Root Namespace:	http://www.w3.org/1999/xhtml					



The W3C validators are hosted on server technology donated by HP, and supported by community donations. **Donate** and help us build better tools for a better web.

2126

Options			
Show Source	Show Outline	List Messages Sequentially Group Error Messages by Type	
☐ Validate error pages	■ Verbose Output	Clean up Markup with HTML-Tidy	
Help on the options is available.			Revalidate

Notes and Potential Issues

The following notes and warnings highlight missing or conflicting information which caused the validator to perform some guesswork prior to validation, or other things affecting the output below. If the guess or fallback is incorrect, it could make validation results entirely incoherent. It is *highly recommended* to check these potential issues, and, if necessary, fix them and re-validate the document.

Using Direct Input mode: UTF-8 character encoding assumed

Unlike the "by URI" and "by File Upload" modes, the "Direct Input" mode of the validator provides validated content in the form of characters pasted or typed in the validator's form field. This will automatically make the data UTF-8, and therefore the validator does not need to determine the character encoding of your document, and will ignore any charset information specified.

If you notice a discrepancy in detected character encoding between the "Direct Input" mode and other validator modes, this is likely to be the reason. It is neither a bug in the validator, nor in your document.

↑ TOP

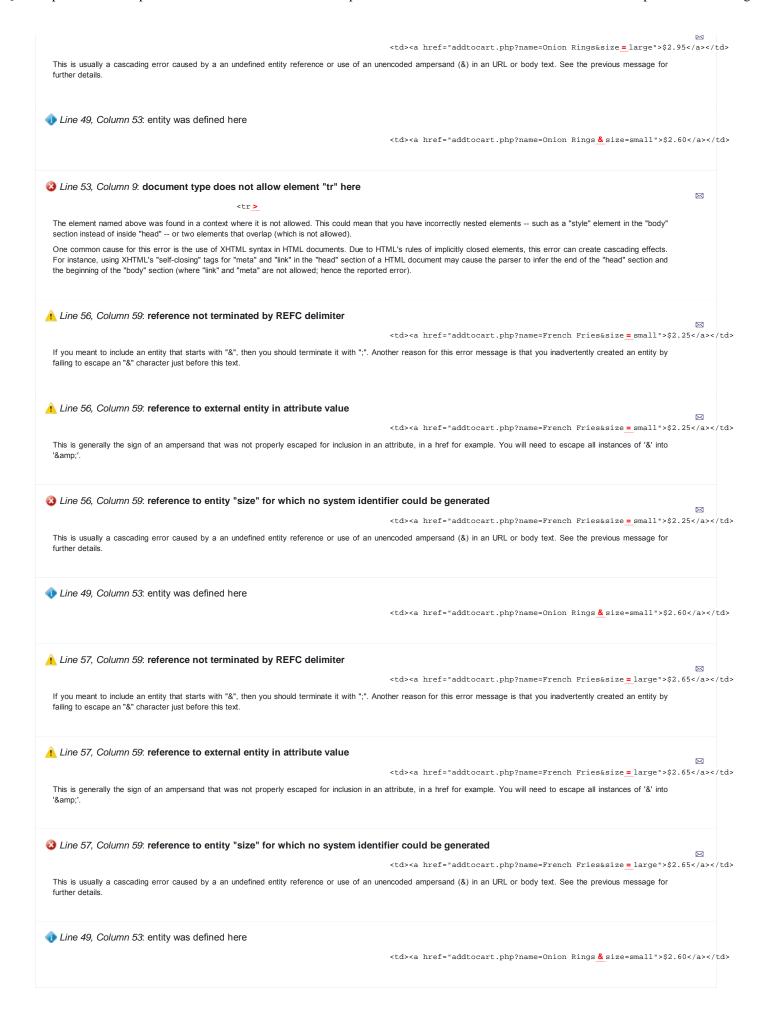
 \bowtie

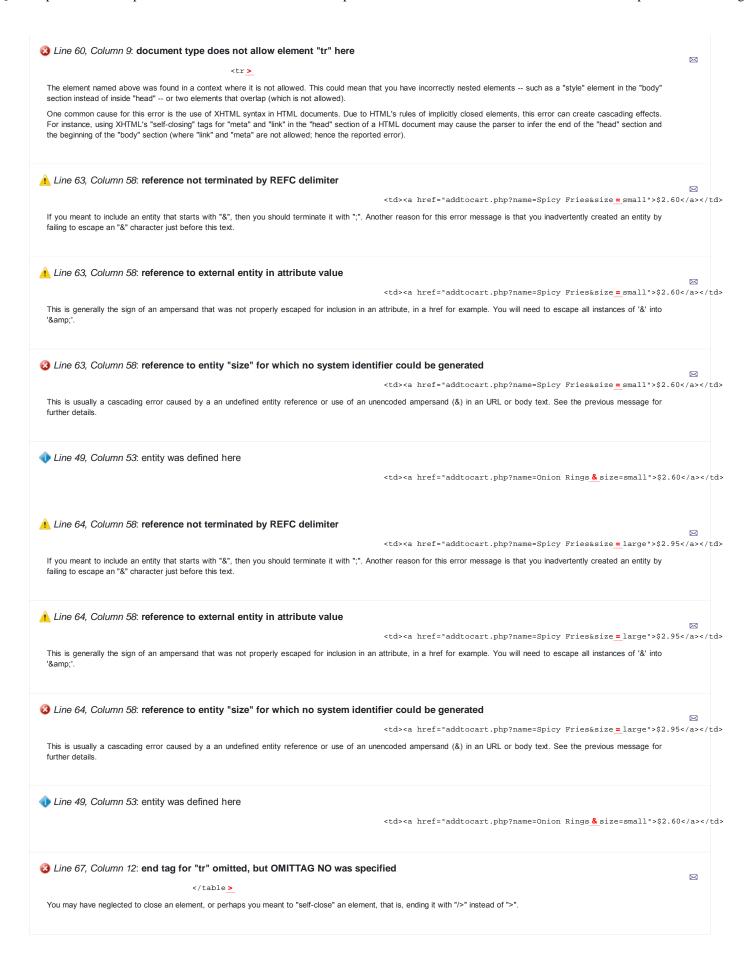
Validation Output: 12 Errors

Line 42, Column 21: document type does not allow element "th" here; assuming missing "tr" start-tag

Item(s): Sides

3 Line 46, Column 9: document type does not allow element "tr" here \bowtie The element named above was found in a context where it is not allowed. This could mean that you have incorrectly nested elements -- such as a "style" element in the "body" section instead of inside "head" -- or two elements that overlap (which is not allowed). One common cause for this error is the use of XHTML syntax in HTML documents. Due to HTML's rules of implicitly closed elements, this error can create cascading effects. For instance, using XHTML's "self-closing" tags for "meta" and "link" in the "head" section of a HTML document may cause the parser to infer the end of the "head" section and the beginning of the "body" section (where "link" and "meta" are not allowed; hence the reported error). Line 49, Column 54: cannot generate system identifier for general entity "size" \$2.60 An entity reference was found in the document, but there is no reference by that name defined. Often this is caused by misspelling the reference name, unencoded ampersands, or by leaving off the trailing semicolon (:). The most common cause of this error is unencoded ampersands in URLs as described by the WDG in "Ampersands in URLs" Entity references start with an ampersand (&) and end with a semicolon (;). If you want to use a literal ampersand in your document you must encode it as "&" (even inside URLs!). Be careful to end entity references with a semicolon or your entity reference may get interpreted in connection with the following text. Also keep in mind that named entity references are case-sensitive; &Aelig; and æ are different characters. If this error appears in some markup generated by PHP's session handling code, this article has explanations and solutions to your problem. Note that in most documents, errors related to entity references will trigger up to 5 separate messages from the Validator. Usually these will all disappear when the original problem is fixed. 3 Line 49, Column 54: general entity "size" not defined and no default entity \$2.60 This is usually a cascading error caused by a an undefined entity reference or use of an unencoded ampersand (&) in an URL or body text. See the previous message for A Line 49, Column 58: reference not terminated by REFC delimiter \bowtie \$2.60 If you meant to include an entity that starts with "&", then you should terminate it with ";". Another reason for this error message is that you inadvertently created an entity by failing to escape an "&" character just before this text. A Line 49, Column 58: reference to external entity in attribute value \$2.60 This is generally the sign of an ampersand that was not properly escaped for inclusion in an attribute, in a href for example. You will need to escape all instances of '&' into '&' 🐼 Line 49, Column 58: reference to entity "size" for which no system identifier could be generated \$2.60 This is usually a cascading error caused by a an undefined entity reference or use of an unencoded ampersand (&) in an URL or body text. See the previous message for further details Line 49, Column 53: entity was defined here \$2.60 Line 50, Column 58: reference not terminated by REFC delimiter \$2.95 If you meant to include an entity that starts with "&", then you should terminate it with ",". Another reason for this error message is that you inadvertently created an entity by failing to escape an "&" character just before this text. 1 Line 50, Column 58: reference to external entity in attribute value \$2.95 This is generally the sign of an ampersand that was not properly escaped for inclusion in an attribute, in a href for example. You will need to escape all instances of '&' into '&' Line 50, Column 58: reference to entity "size" for which no system identifier could be generated







Home About... News Docs Help & FAQ Feedback Contribute

W3C® open source

This service runs the W3C Markup Validator, <u>v1.2+hg</u>.

COPYRIGHT © 1994-2011 W3C® (MIT, ERCIM, KEIO), ALL RIGHTS RESERVED. W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

I VALIDATOR