

## ICA03 – Debugging – Coupon Generator

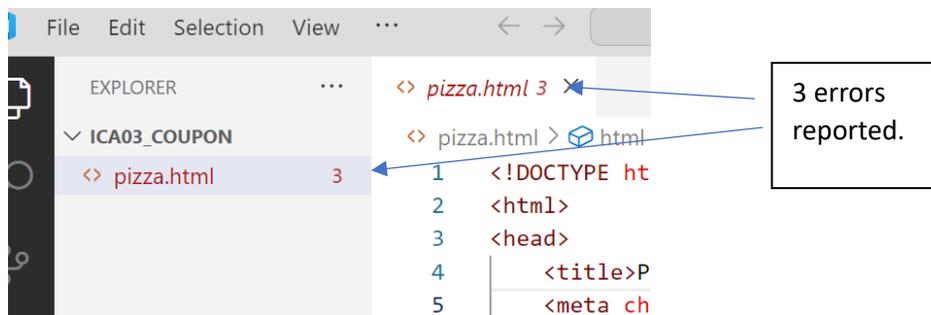
In this activity, students will work through the process of troubleshooting / debugging code. This gives students a “hands on” debugging experience and it also highlights some tips / tricks that can be best learned by doing!

As you progress through this activity, you will receive less and less instruction regarding what to fix, and where the problem is. This is intentional!

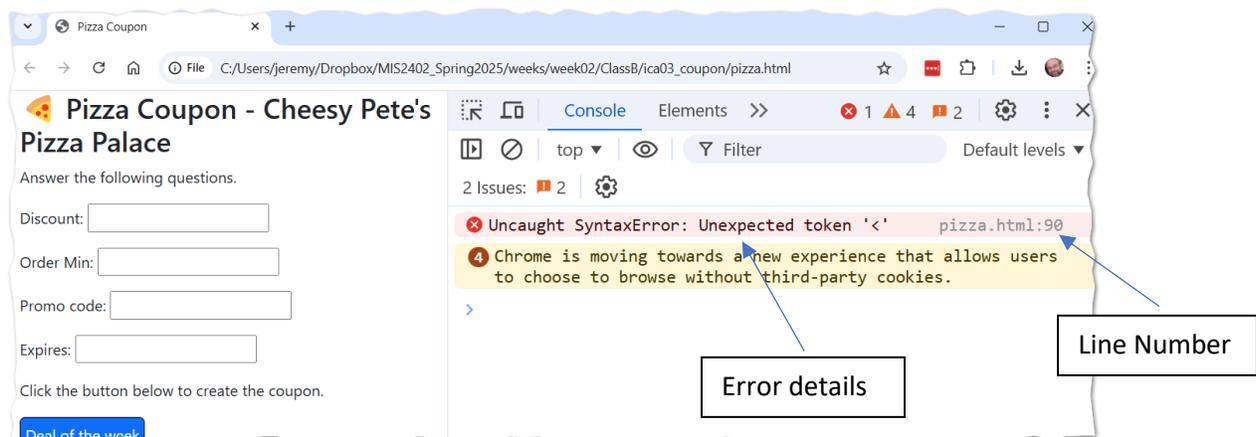
The purpose of our code is to prompt the user for some input, and then use that input to generate coupons for an imaginary restaurant, Cheesy Pete’s Pizza Palace. (We will see more of Cheesy Pete’s later this semester.)

### Instructions

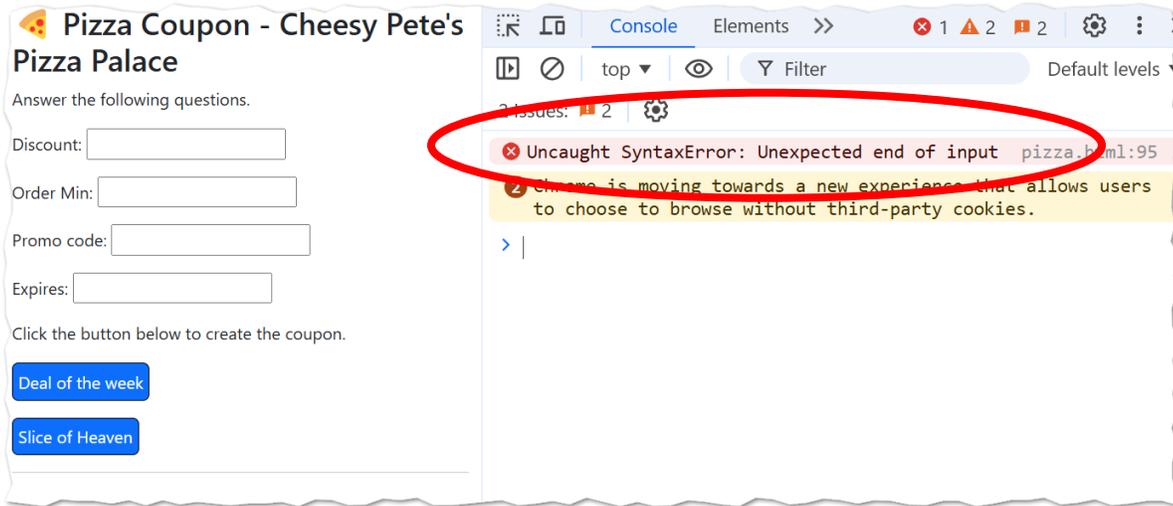
1. Retrieve ica03\_coupon.zip provided by your instructor.
2. Unzip the file so that you have ica03\_coupon folder in your MIS2402 workspace folder.
3. Preview the pizza.html file in Chrome. Oh no! Nothing shows up! Check the console log. View the page source in Chrome. Still nothing.
4. If you haven’t spotted the error yet, now would be a good time to check the HTML using FireFox. Open pizza.html in FireFox, view the page source, the problem should now be obvious. Fix this problem in VS Code and save your work. Close FireFox when you are done with this step.
5. Looking at VS Code, you can quickly see that there are at least 3 syntax errors in this file. That is, there are three syntax errors that VS Code can detect!



6. We could start looking in the code now. Or we could try opening our work in Chrome again. Either approach is good. But... for the sake of this demonstration, let’s open this page up in Chrome. View the console log.



- Looking there we see that there is “unexpected token” on **line 90**.
- Take a moment and google “what is unexpected token in javascript”? Read what comes up.  
Discuss – what does this “unexpected token” error mean?
- Go to **line 90** in VS Code. Fix the problem.
- Try your code again in Chrome. You now have a new error!!

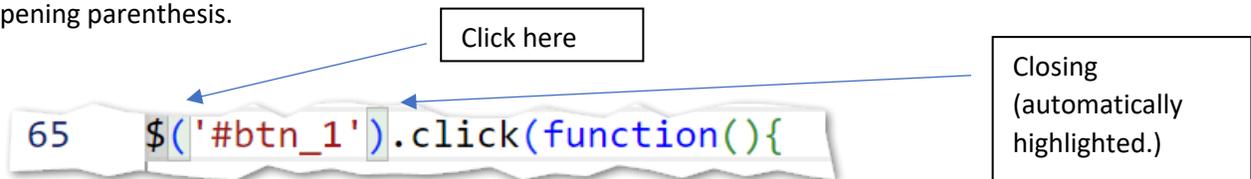


- This error says “Unexpected end of input” on **line 95**. Use Google search to try to figure out what that means.  
Discuss – what is an “Unexpected end of input” error?

This kind of error is quite hard to troubleshoot. It’s hard because your correction needs to take place much earlier in the code. VS Code can let you know that something unexpected happened, but it doesn’t “know” or “understand” what you wanted to do! VS Code can only look for patterns in your code, and alert you when it detects a pattern that does not conform to the syntax of the languages.

And ... just like failing to close a <title> tag correctly has consequences, so does failing to close a pair of quotes "", a pair of parentheses (), a pair of curly brackets {} or some combination of those things.

- Go back to the beginning of your <script> section. Click on the opening parenthesis on line 63. When you do that, VS Code will automatically highlight the closing parenthesis that **appears to be** the correct partner to the opening parenthesis.



- Systematically go through the opening parenthesis in your file, clicking each one, until you find the opening / closing combination that is incorrect. Fix the bug.
- You’ll notice that there are now no more syntax errors being reported by VS Code. Are we done?!? Let’s test.

15. You will notice that we have some bugs left to fix. You might see output like this:



```
83     let promo = $("#promo").val();
84     let expires = $("#textexpires").val();
```

Look at the code on lines 83 and 84 and see if you can spot the problem.

16. In the last step, you can see that the identifiers in HTML and JavaScript were inconsistent.

Whether you are working in HTML, JavaScript, or any other language it really pays to name things consistently and sensibly.

In JavaScript (and most other languages) case matters. So, here you should fix **lines 83** and **84** so that they match the HTML.

17. Errors remain! It seems like one or two of the buttons aren't working quite right. Make your corrections.

*Do I have to use those names?* There's nothing magical about our name choices here. We could have given this input tags ids of thing1, thing2, and thing3.

However, later in the code, we have used ids with the textName pattern. You can scan through the code to see this for yourself!

18. Turn in your work by uploading **pizza.html** to the ICA 3 assignment on Canvas.

### How will this be graded?

- You get 80 points for making all the fixes in steps 1 through 16
- You get 10 points for fixing the buttons
- You get 10 points for fixing the problem related to textDisplayed1