ICA03 – Debugging – Silly Song Generator

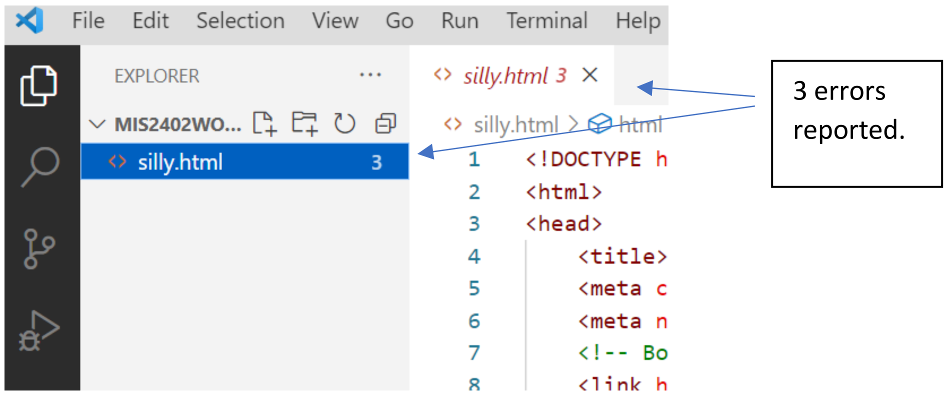
In this activity, students will work through the process of troubleshooting / debugging code. This gives students a "hands on" debugging experience and it also highlights some tips / tricks that can be best learned by doing!

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 nouns, a plural noun, a place, and an adjective. The code then uses these words to construct a Silly Song using concatenation.

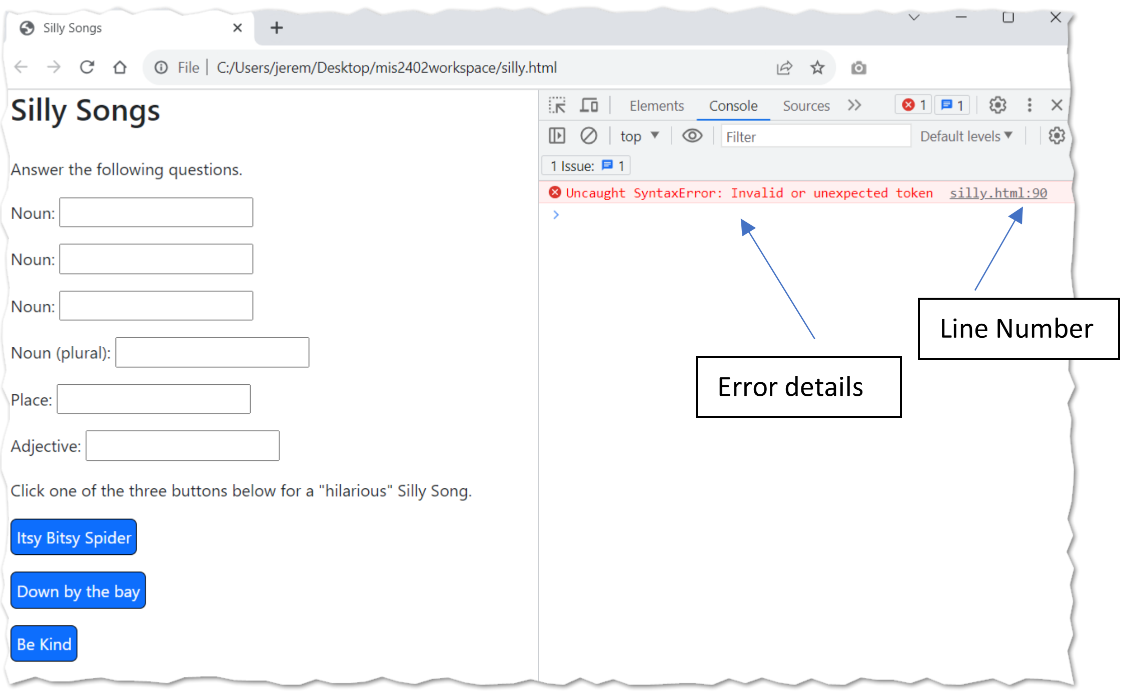
Instructions

1. Retrieve ica03_sillysongs.zip provided by your instructor.
2. Unzip the file so that you have ica03_sillysongs folder in your MIS2402 workspace folder.
3. Preview the silly.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 silly.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!



```
<!DOCTYPE h
<html>
<head>
    <title>
    <meta c
    <meta n
    <!-- Bo
    <link h
```

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.



Silly Songs

Answer the following questions.

Noun:

Noun:

Noun:

Noun (plural):

Place:

Adjective:

Click one of the three buttons below for a "hilarious" Silly Song.

Itsy Bitsy Spider

Down by the bay

Be Kind

```
Uncaught SyntaxError: Invalid or unexpected token   silly.html:90
```

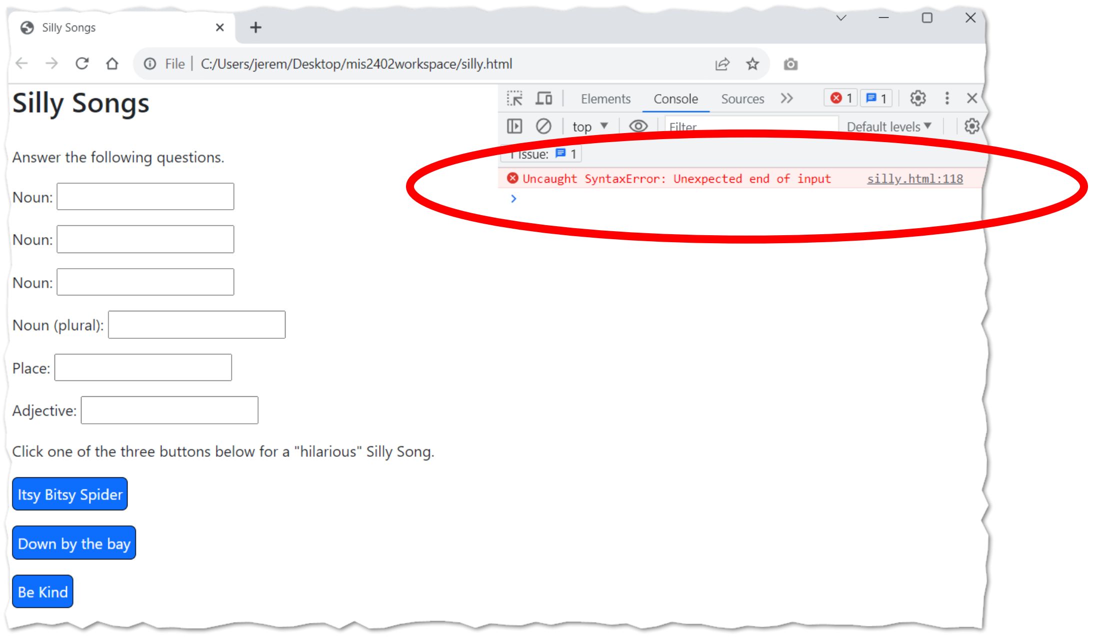
7. Looking there we see that there is “unexpected token” on **line 90**.

8. Take a moment and google “what is unexpected token in javascript”? Read what comes up.

   Discuss – what does this “unexpected token” error mean?

9. Go to **line 90** in VS Code. Fix the problem.

10. Try your code again in Chrome. You now have a new error!!



Silly Songs

Answer the following questions.

Noun:

Noun:

Noun:

Noun (plural):

Place:

Adjective:

Click one of the three buttons below for a "hilarious" Silly Song.

Itsy Bitsy Spider

Down by the bay

Be Kind

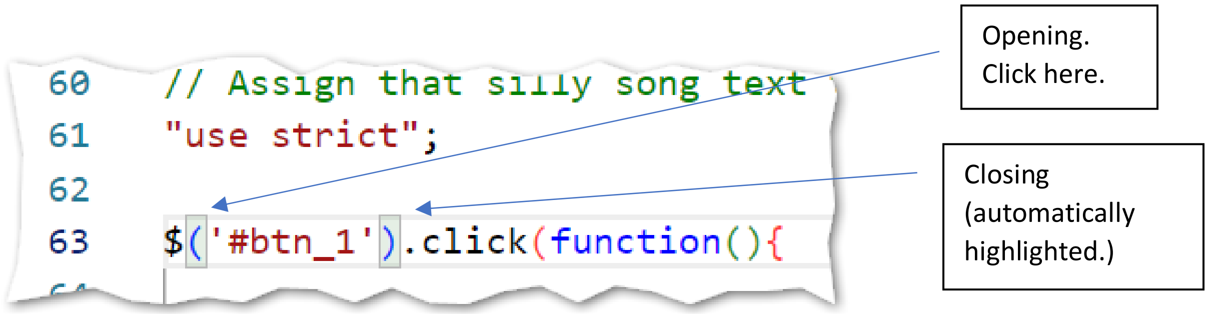
```
Uncaught SyntaxError: Unexpected end of input   silly.html:118
```

11. This error says “Unexpected end of input” on **line 118**. 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.

12. 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.



```
60    // Assign that silly song text
61    "use strict";
62
63    $('#btn_1').click(function(){
```

13. 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.

There’s ANOTHER thing that can help you find tricky errors. Comments!

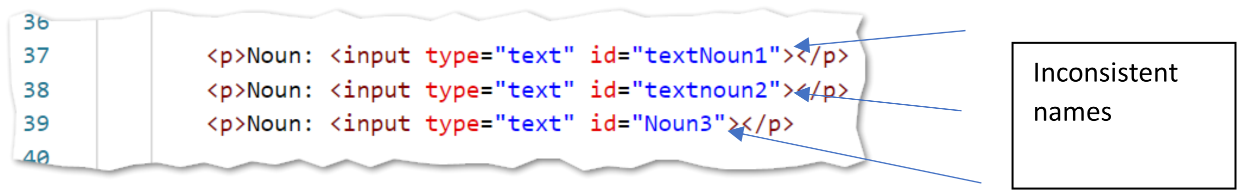
Add in single line comment on **lines 62, 79, and 98** to describe what the click event code is doing.

For example: on line 62 you could add:

```
// Process the Itsy Bitsy Spider Song for Button 1
```

14. 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. Let’s start by looking at the HTML portion of your code.



```
36
37          <p>Noun: <input type="text" id="textNoun1"></p>
38          <p>Noun: <input type="text" id="textnoun2"></p>
39          <p>Noun: <input type="text" id="Noun3"></p>
40
```

16. In the image above, you can see that the HTML itself has inconsistencies in it.

    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 38** and **39** so that they match the naming convention suggested by **line 37**.

17. Two errors remain! One is a logic error. Weirdly enough, it seems that all three buttons are doing the same thing. Investigate why all three buttons might be triggering the same block of code. 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 textNoun1 pattern. You can scan through the code to see this for yourself!

18. The remaining error is a syntax error. Check the console log and fix it.

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