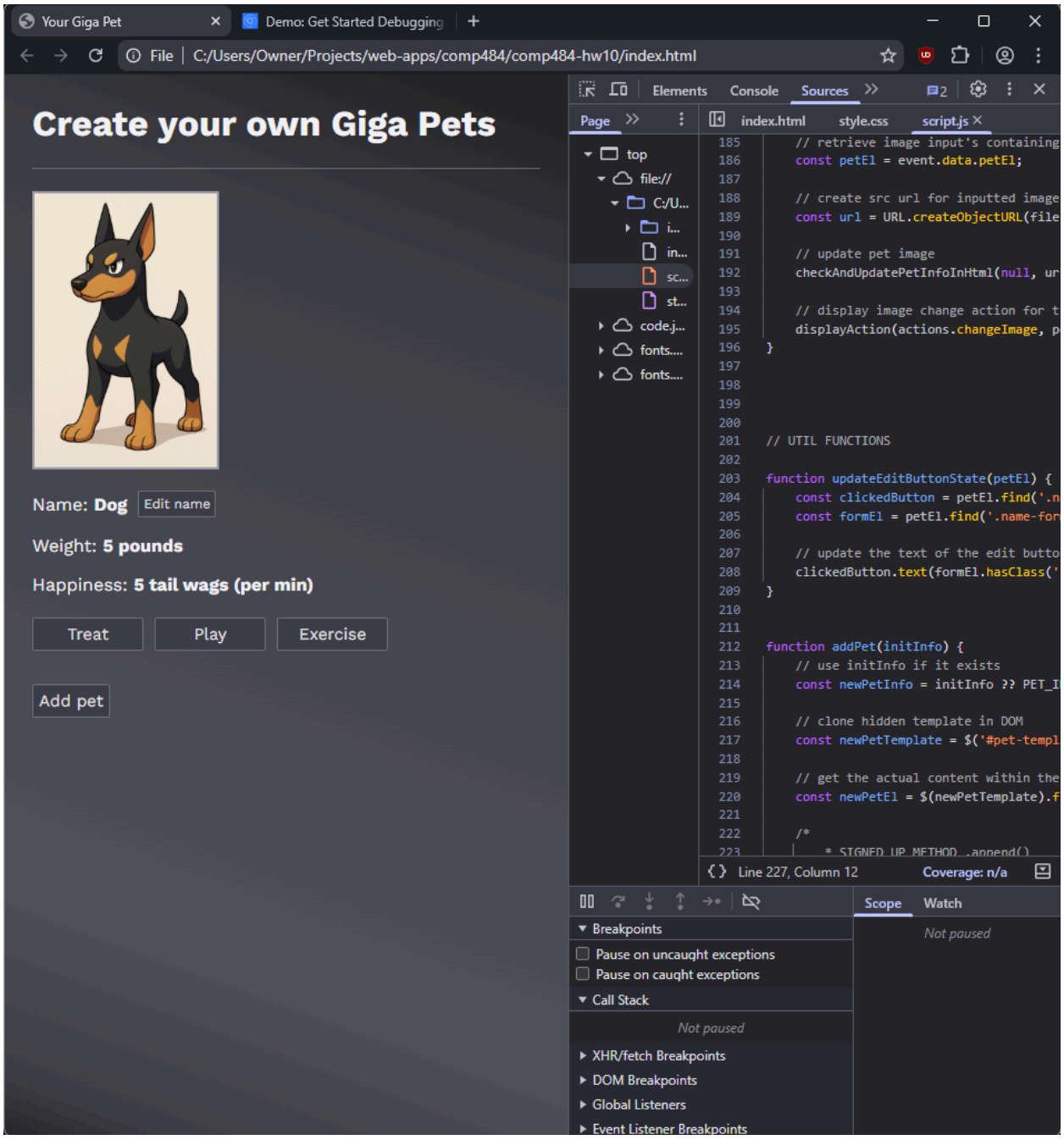
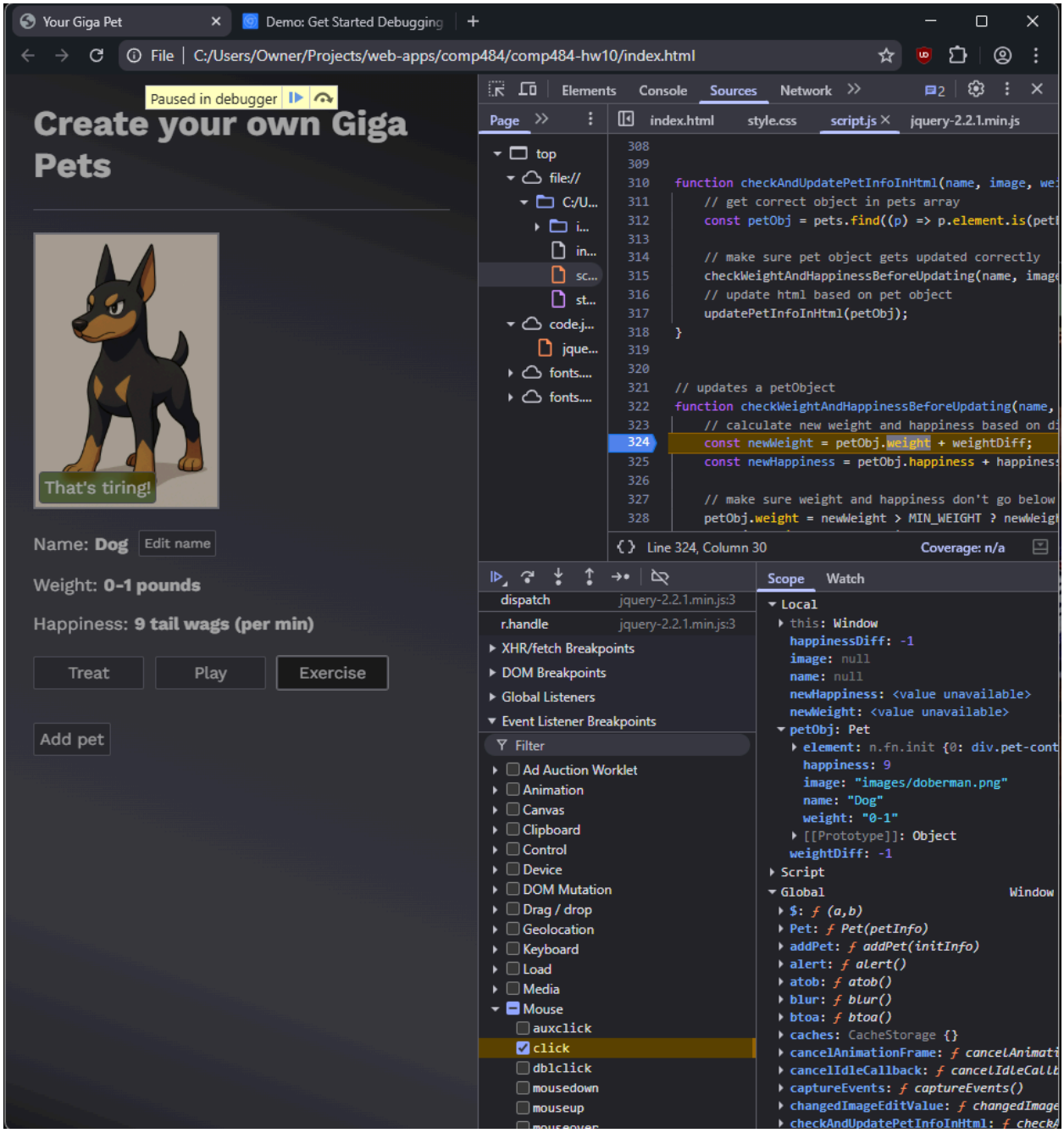
Thomas Scott
3 December 2025

Homework 10 - Chrome Dev Tools



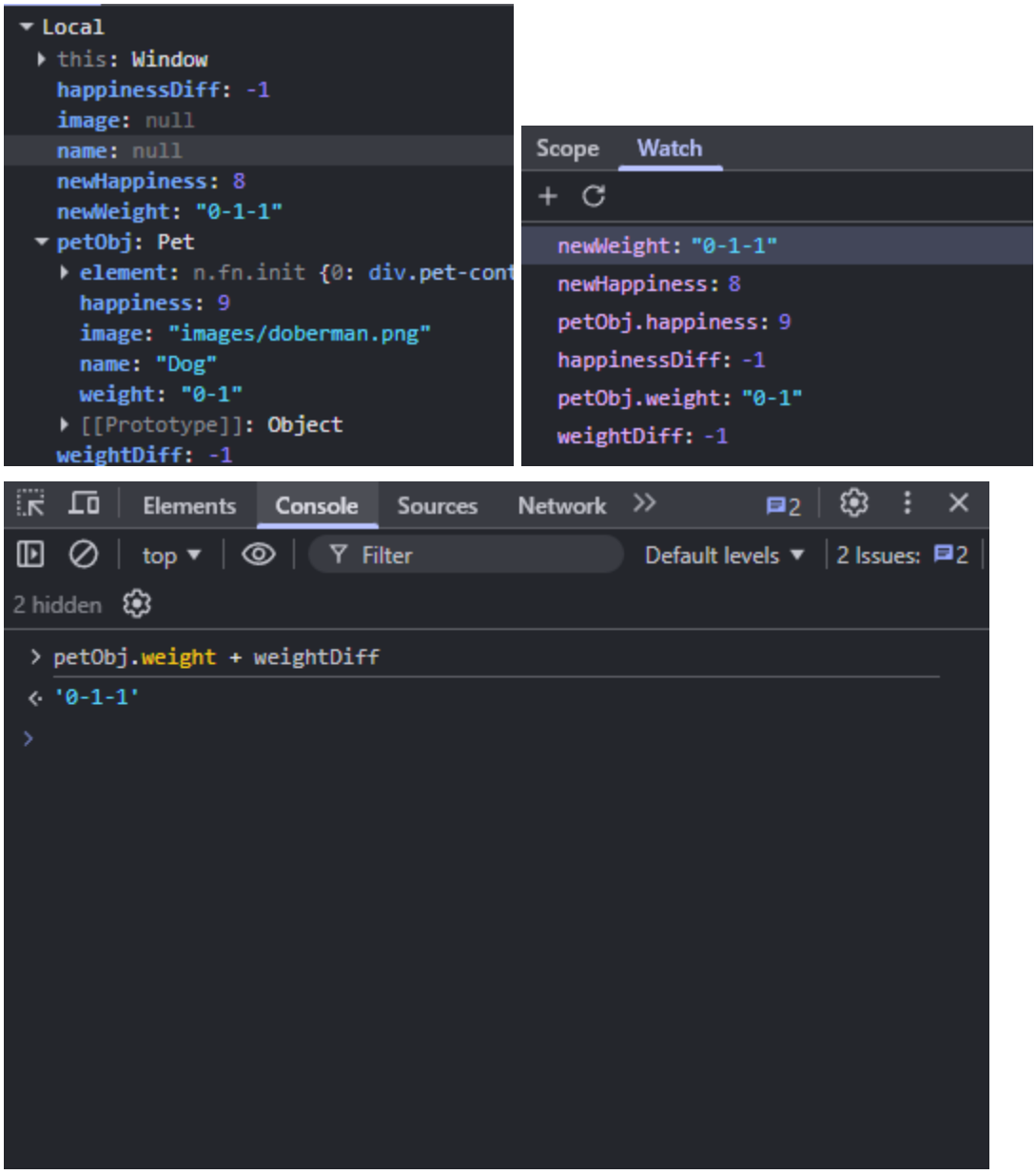
Step 1-2, Debug JavaScript

I opened the dev tools on my project 2 site and navigated to the Sources tab.



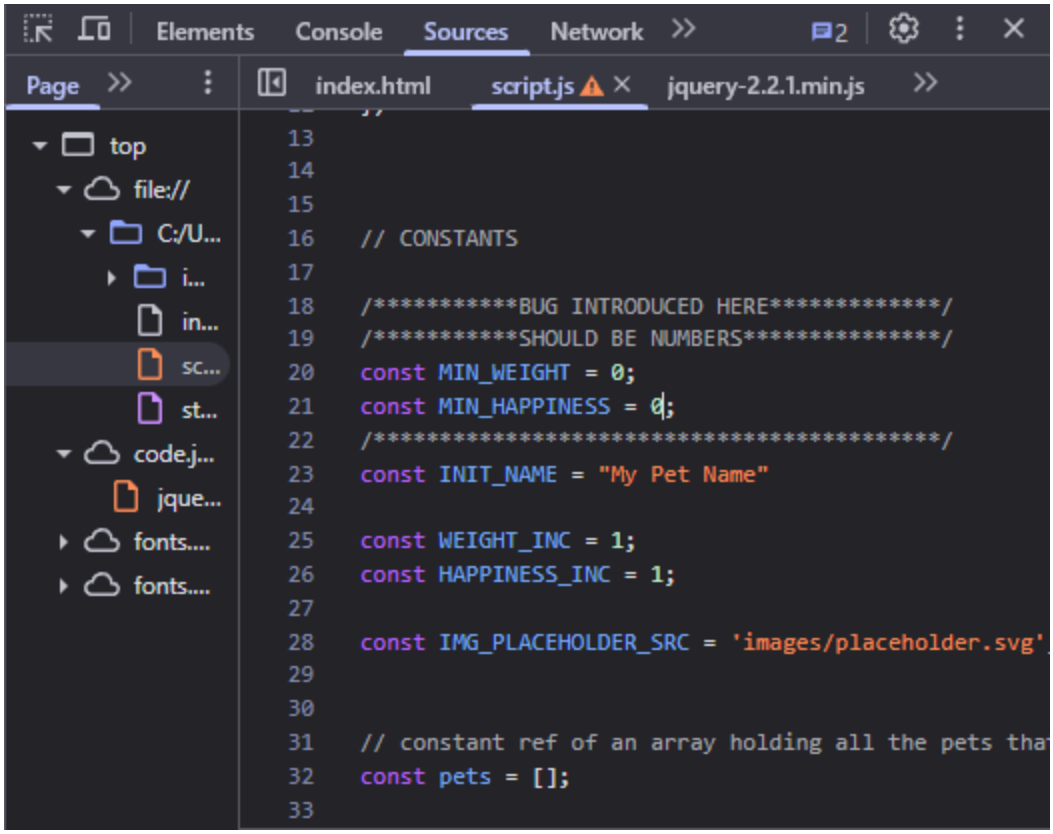
Step 3-5, Debug JavaScript

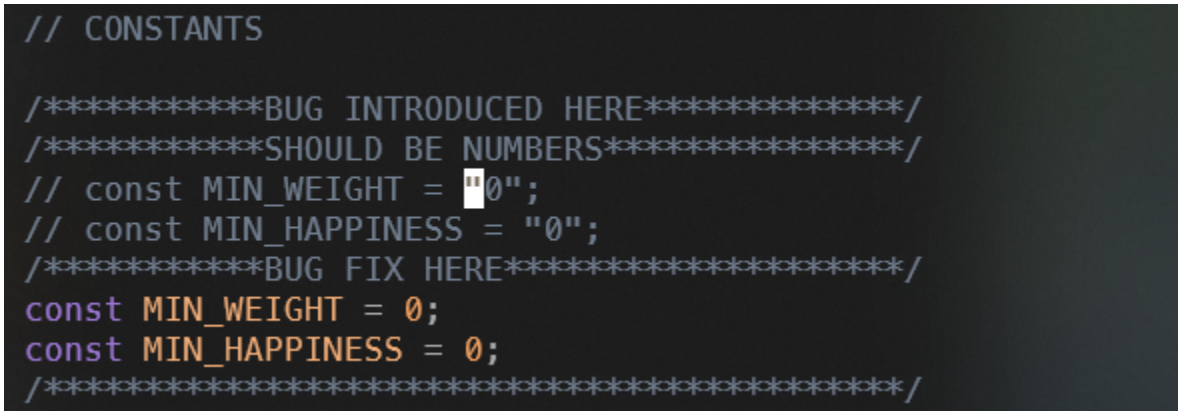
In order to debug a bug within the weight calculation, I set a breakpoint on click events as the bug occurred when clicking the exercise button. I then set a line breakpoint on the line that calculated the new weight value, and stepped through the code to see what was happening.



Step 6, Debug JavaScript

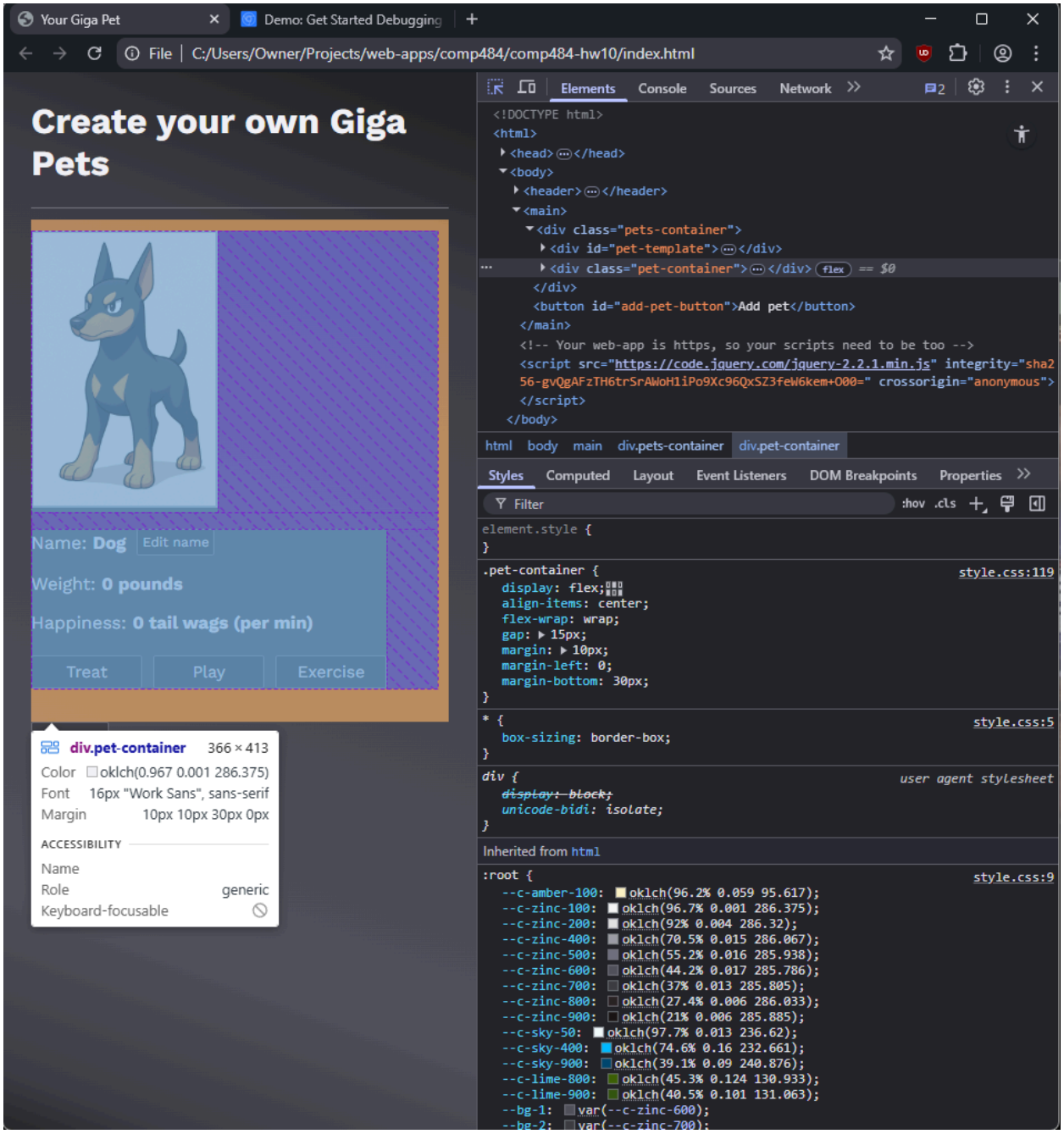
In finding the function that seemed to be problematic, I checked the values of variables in a few ways. The first method I used was to just look at the variables in the local scope at the breakpoint. The second method was setting watch variables to see how specific variables change over time. With these two methods, you can see that the value of petObj.weight is a string when it should be a number. With this in mind, I used the third method of tracking variables, which is to use them in the console. This method allowed me to confirm that the result of the addition was a string concatenation rather than number addition.





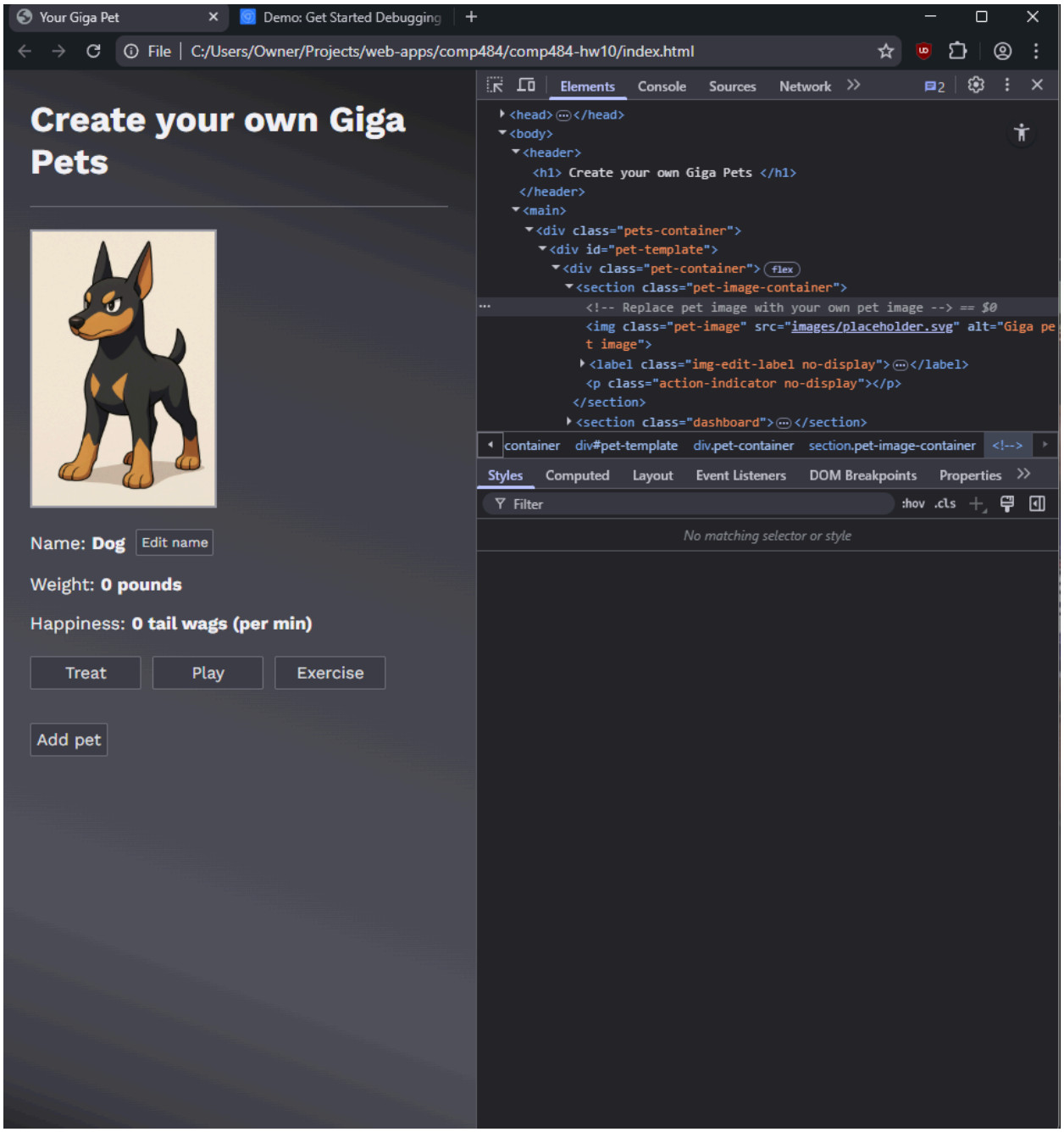
Step 7, Debug JavaScript

The dev tools allow you to directly change the code that has been sent to the browser. Understanding that the bug existed because weights were being processed as strings, I changed the initial value of weight to be a number rather than a string. However, since the pet object on the screen had already been loaded, this fix within the dev tools didn't work as the change would need to occur before the pets were loaded. With this in mind, I changed the code directly in my own source code and refreshed the page to fix the bug.



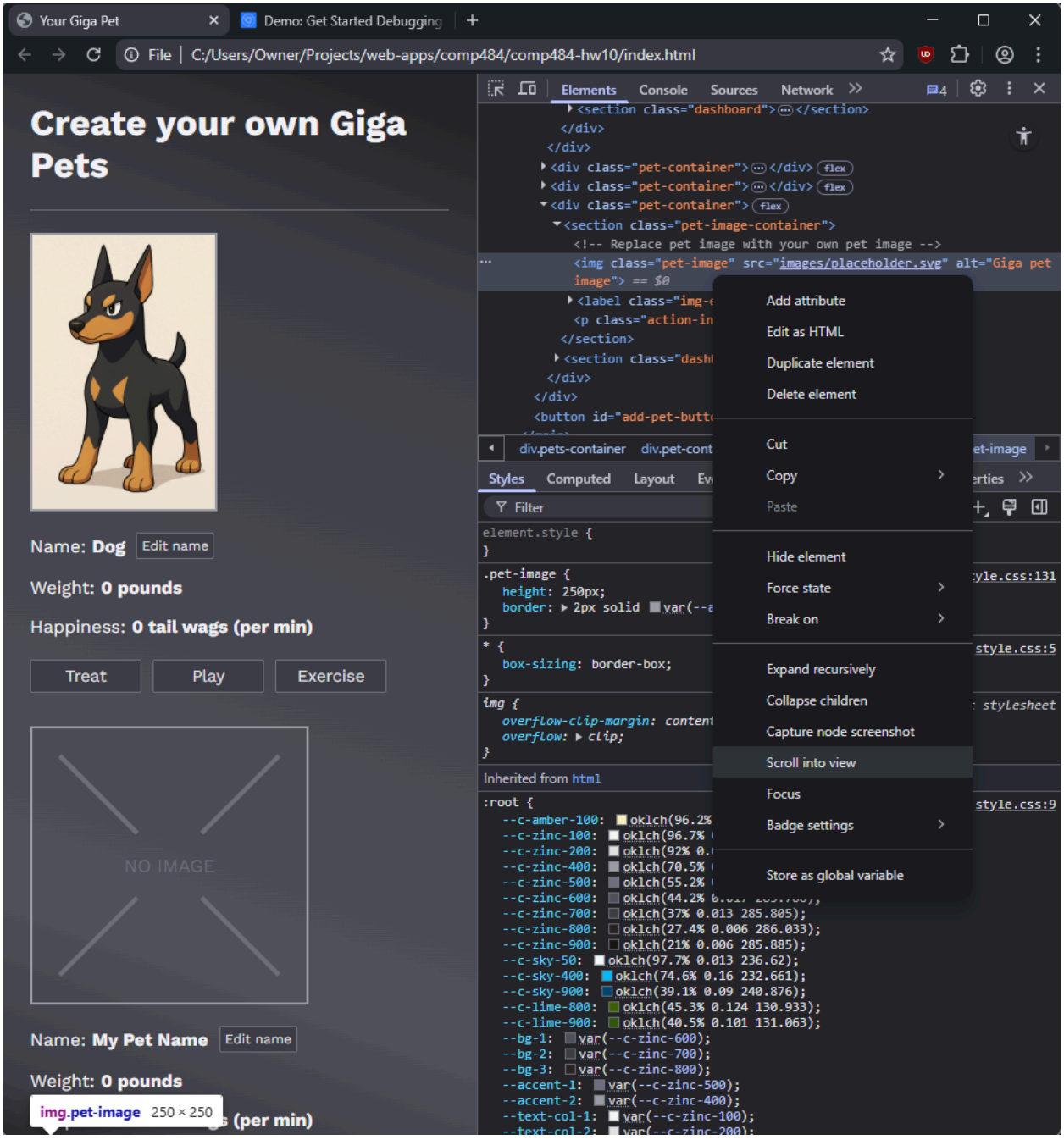
Select element, View DOM nodes

Here you can see me selecting a pet container using the arrow symbol that is on the top left of the dev tools.



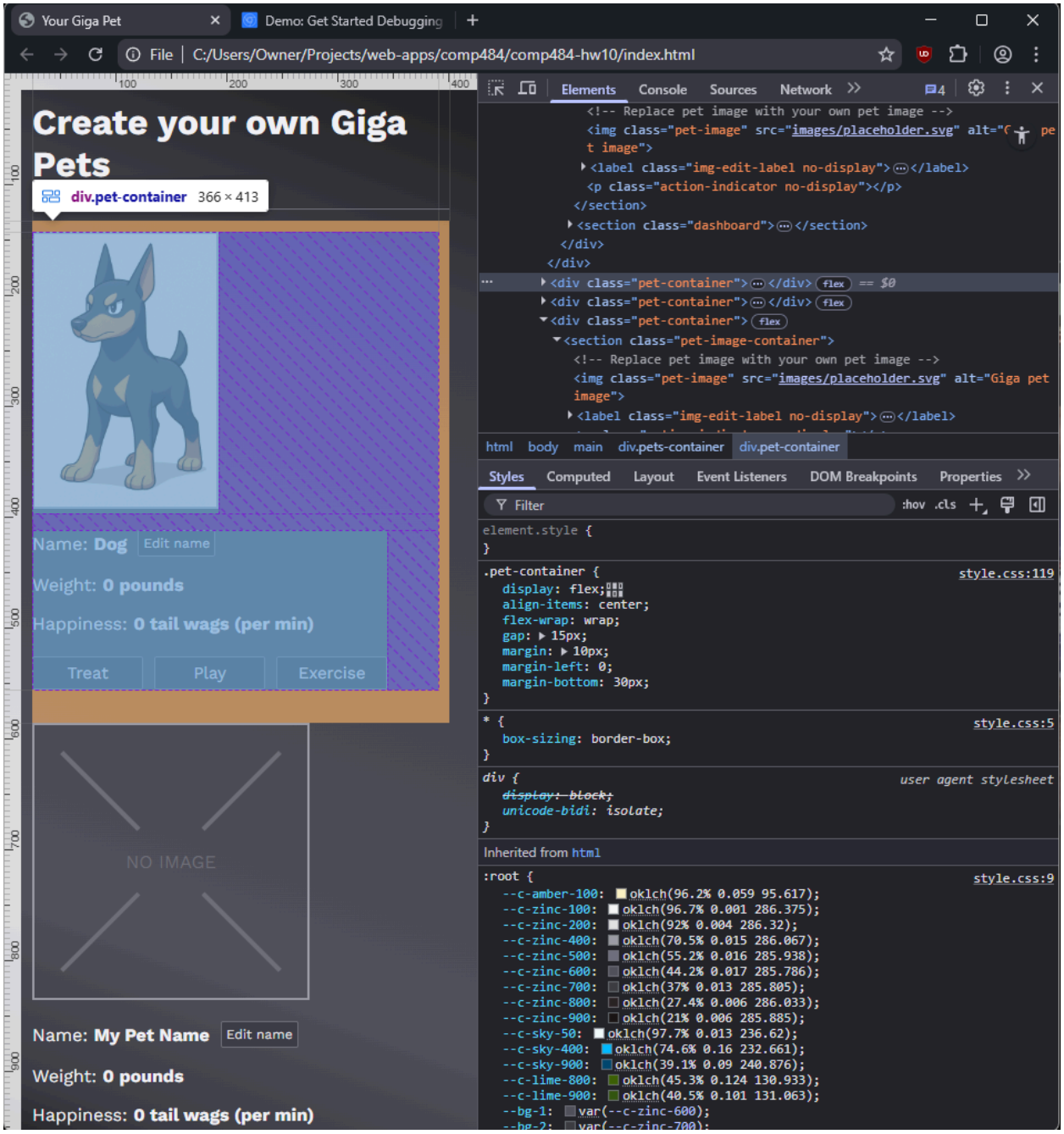
Arrow keys collapse, View DOM nodes

Here you can see the result of me using the right arrow key to open the pet container along with elements inside of it.



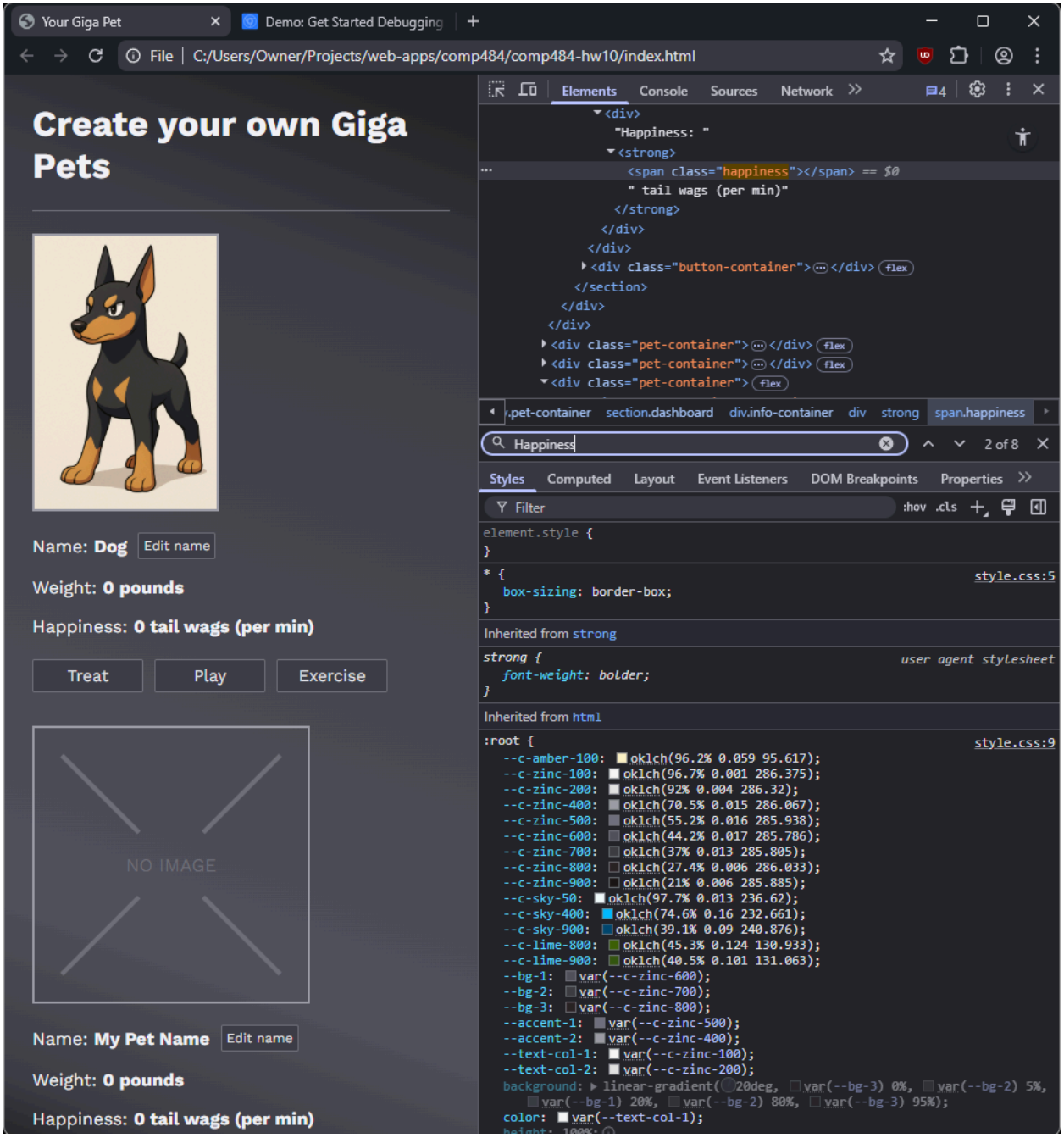
Scroll into view, View DOM nodes

On the bottom left, you can see that I have selected a pet-image that is out of view. In order to quickly get it into view, I selected the “Scroll into view” option when right clicking the element in the DOM.



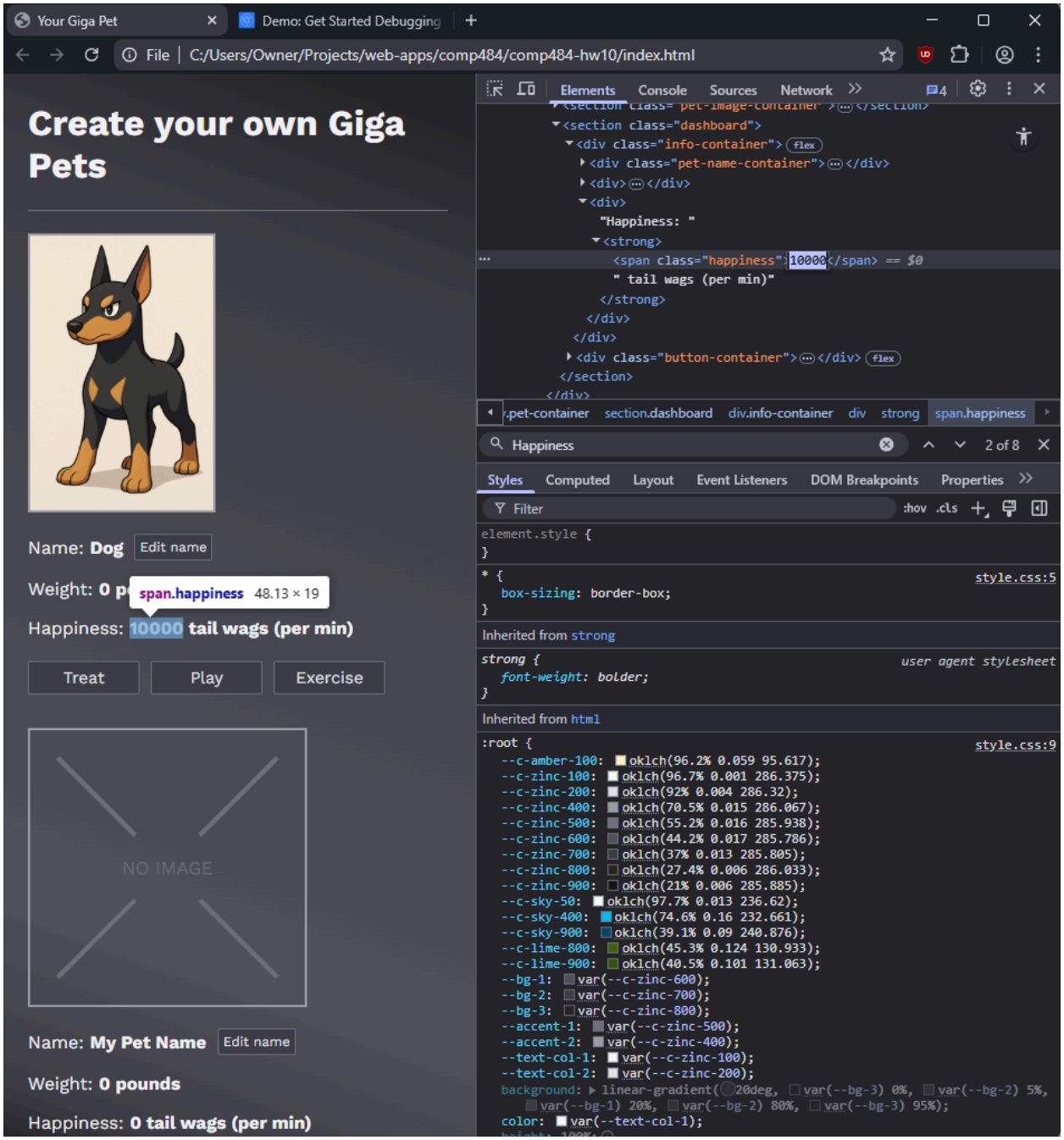
Show ruler, View DOM nodes

Using the command menu, I enabled the option to show rulers when hovering over DOM elements.



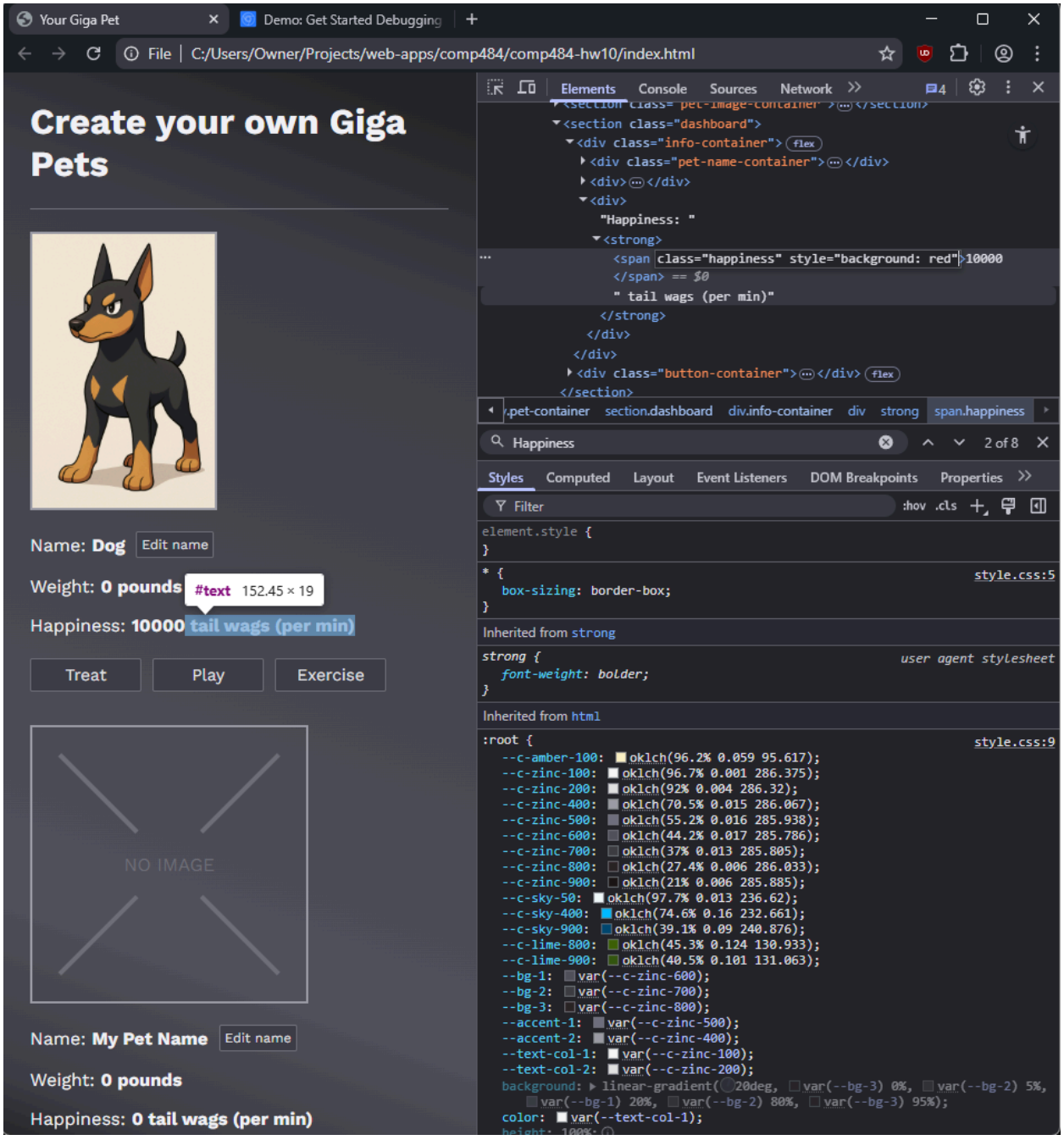
Search nodes, View DOM nodes

Here I use the Ctrl+F shortcut to search for keywords within the DOM. I searched for "Happiness," and one of the results that I was given was a span with a class of "happiness."



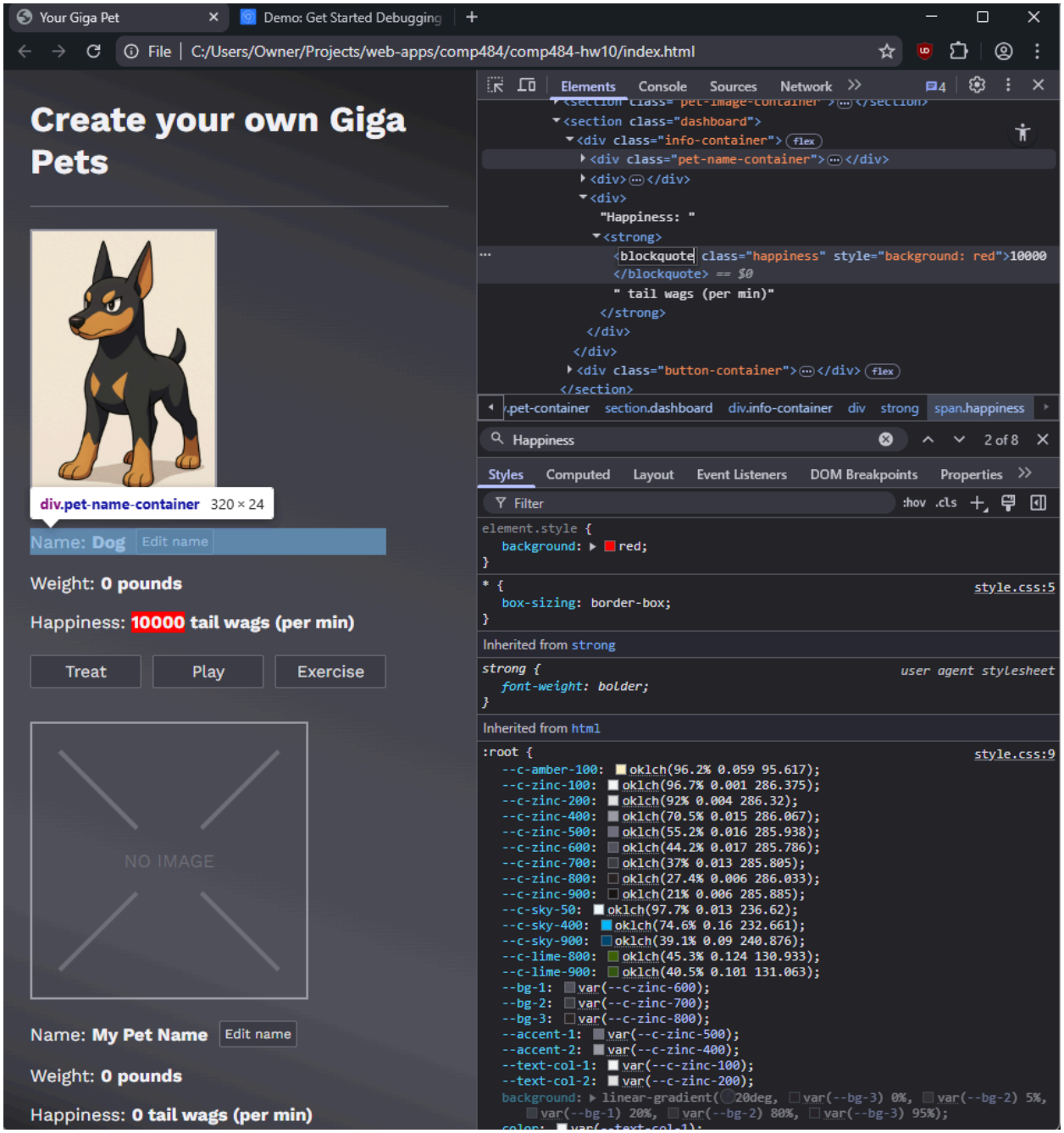
Edit content, Edit the DOM

Here I edited the text content within the span that I had selected. I changed the text to “10000”.



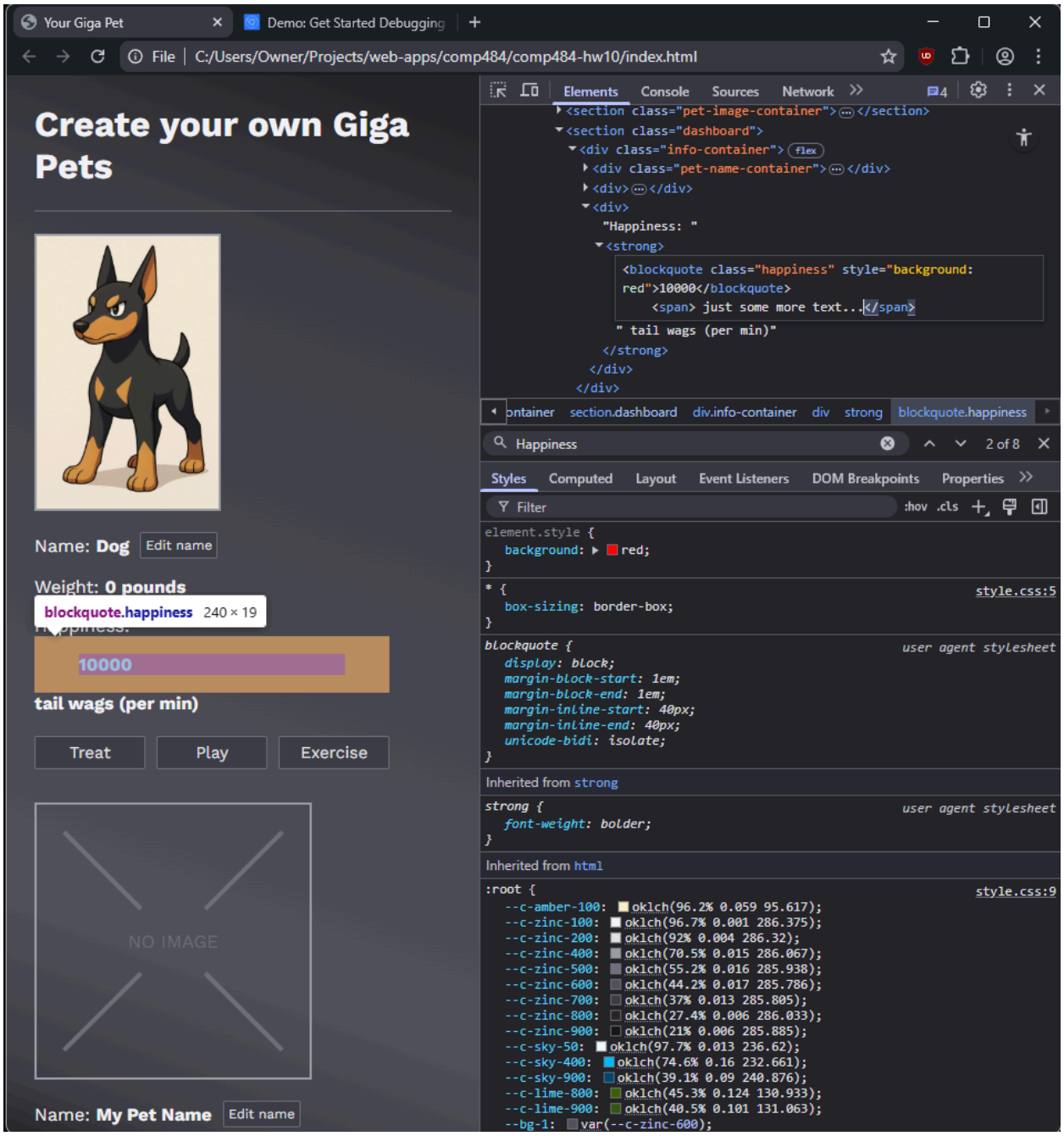
Edit attributes, Edit the DOM

Here you can see me adding a style attribute to the span element that is selected. The result of this was making the background of the element red.



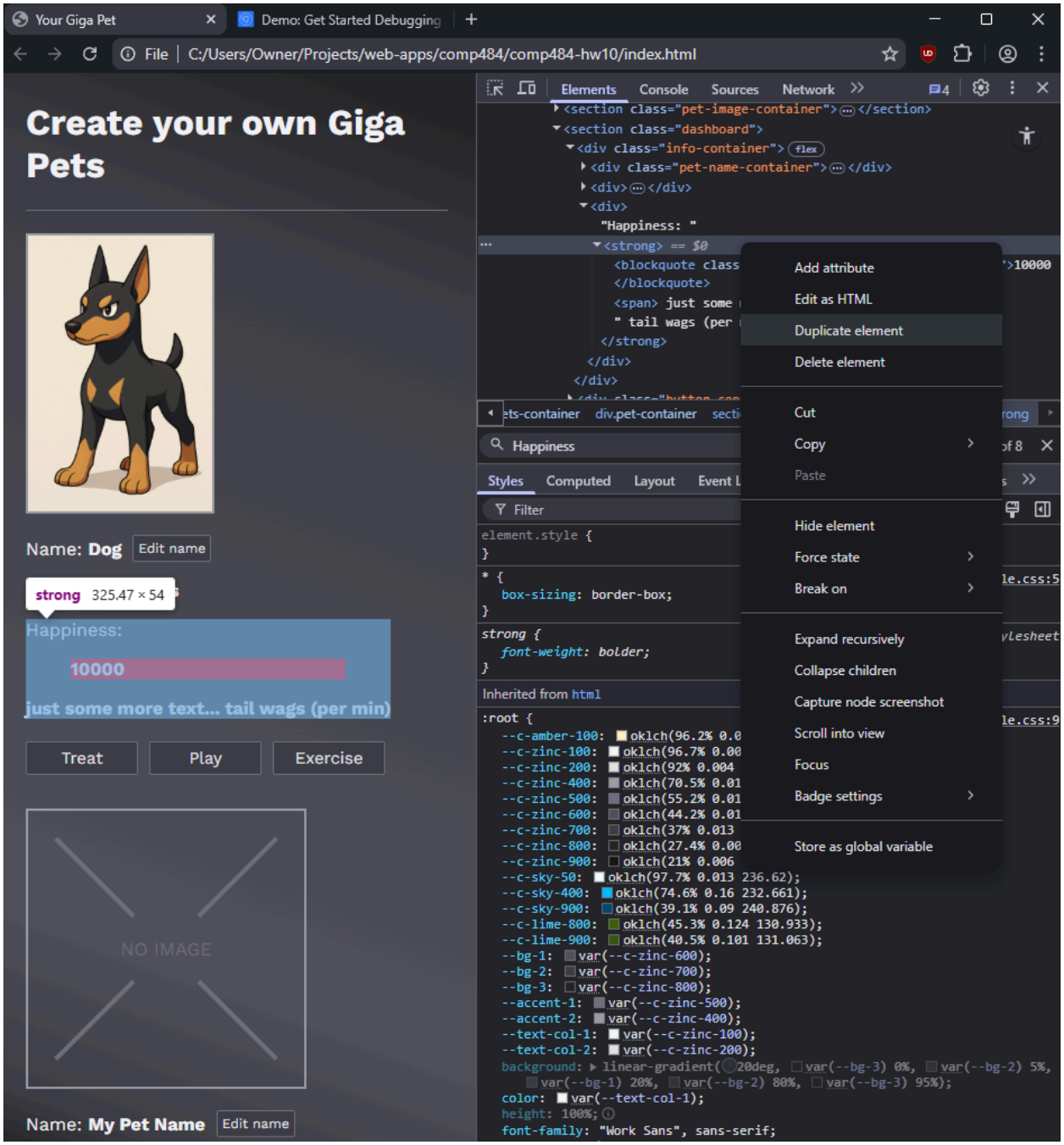
Edit node type, Edit the DOM

Here I change the type of the red node that contains the text “10000” from a span to a blockquote. As seen in the next screenshot, this will change some of the default styling of the element.



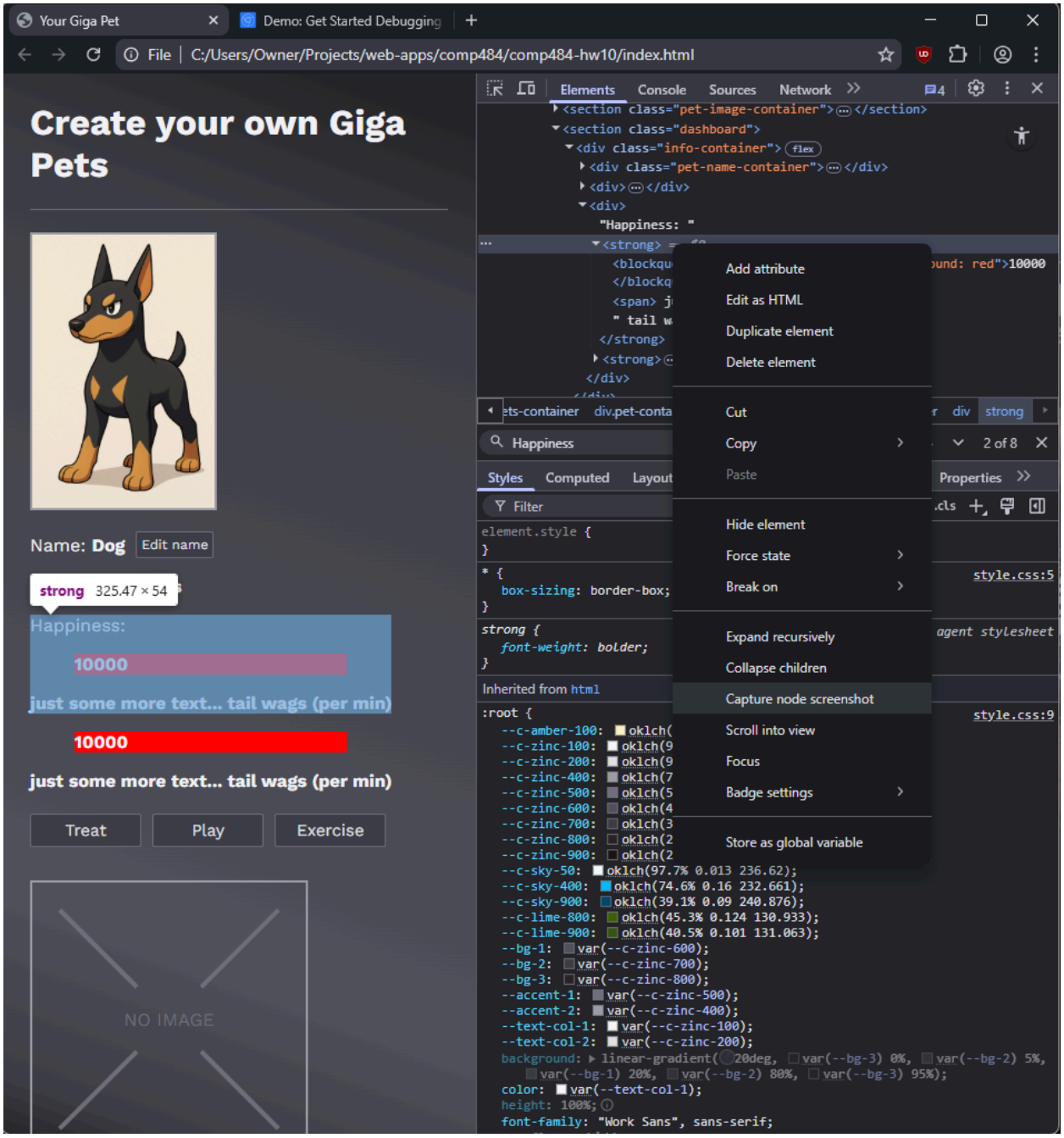
Edit as HTML, Edit the DOM

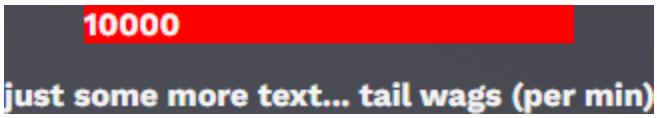
After right clicking on the blockquote, I was able to select the option to edit it as HTML. What this did was open up a mini editor in which I was able to freely write HTML. As you can see, I added another span element after the blockquote using this feature.



Duplicate a node, Edit the DOM

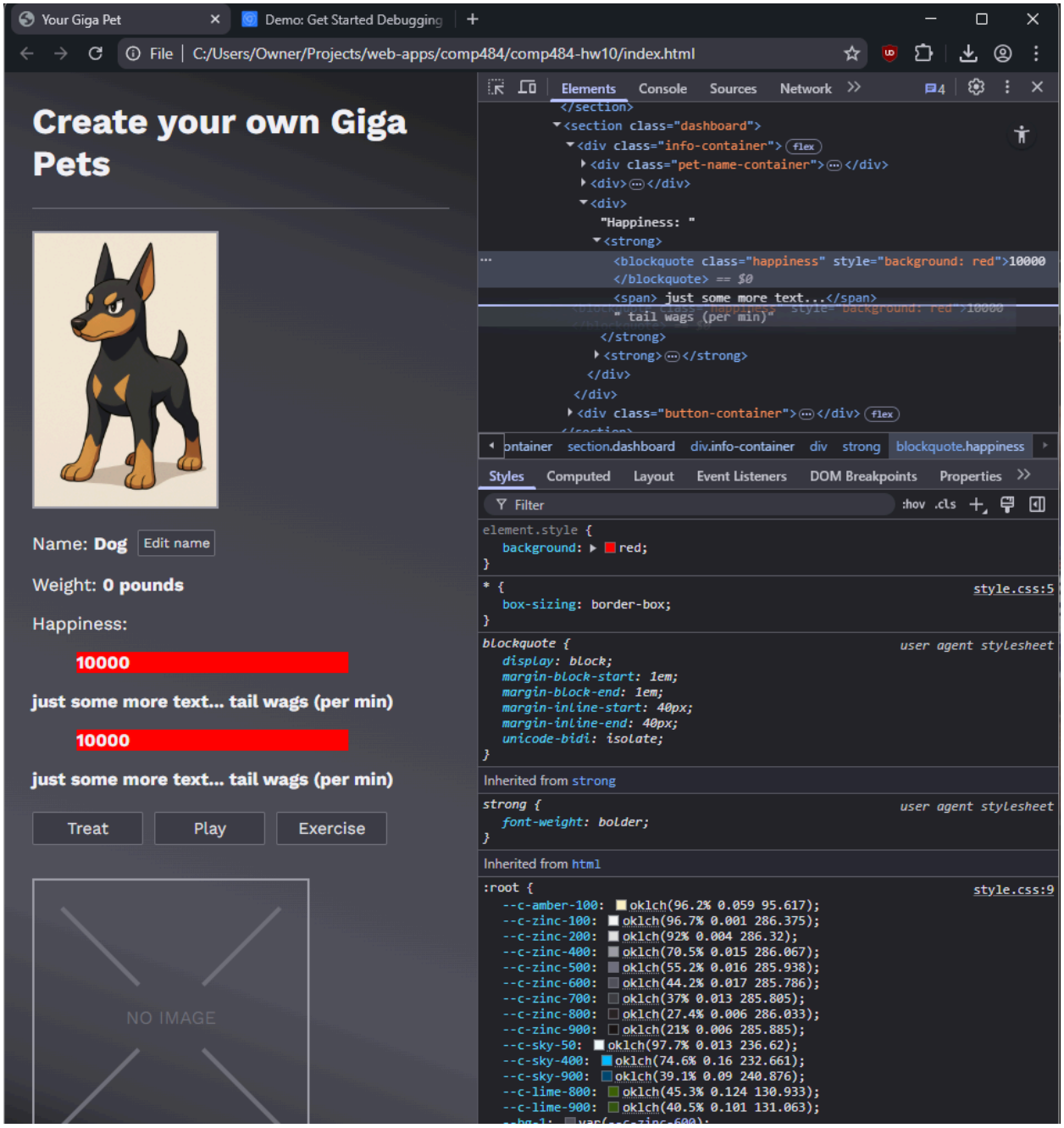
After right clicking the strong element, I was able to select the option to duplicate it. The dev tools duplicated the node and added the new node after the current one selected.





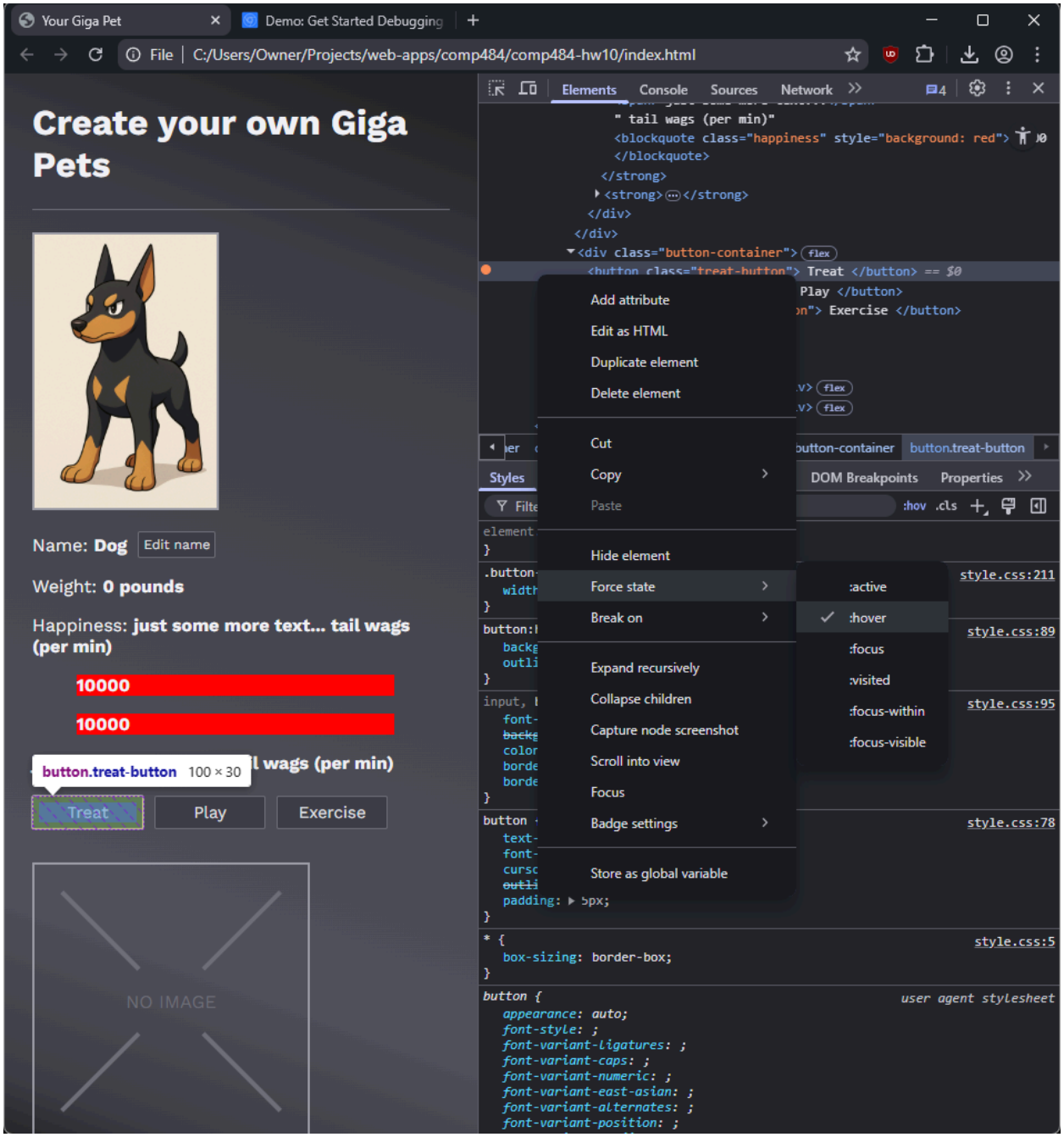
Capture a node screenshot, Edit the DOM

After right clicking the strong element, I was able to select the option to capture a screenshot of it. The result of this screenshot is seen above.



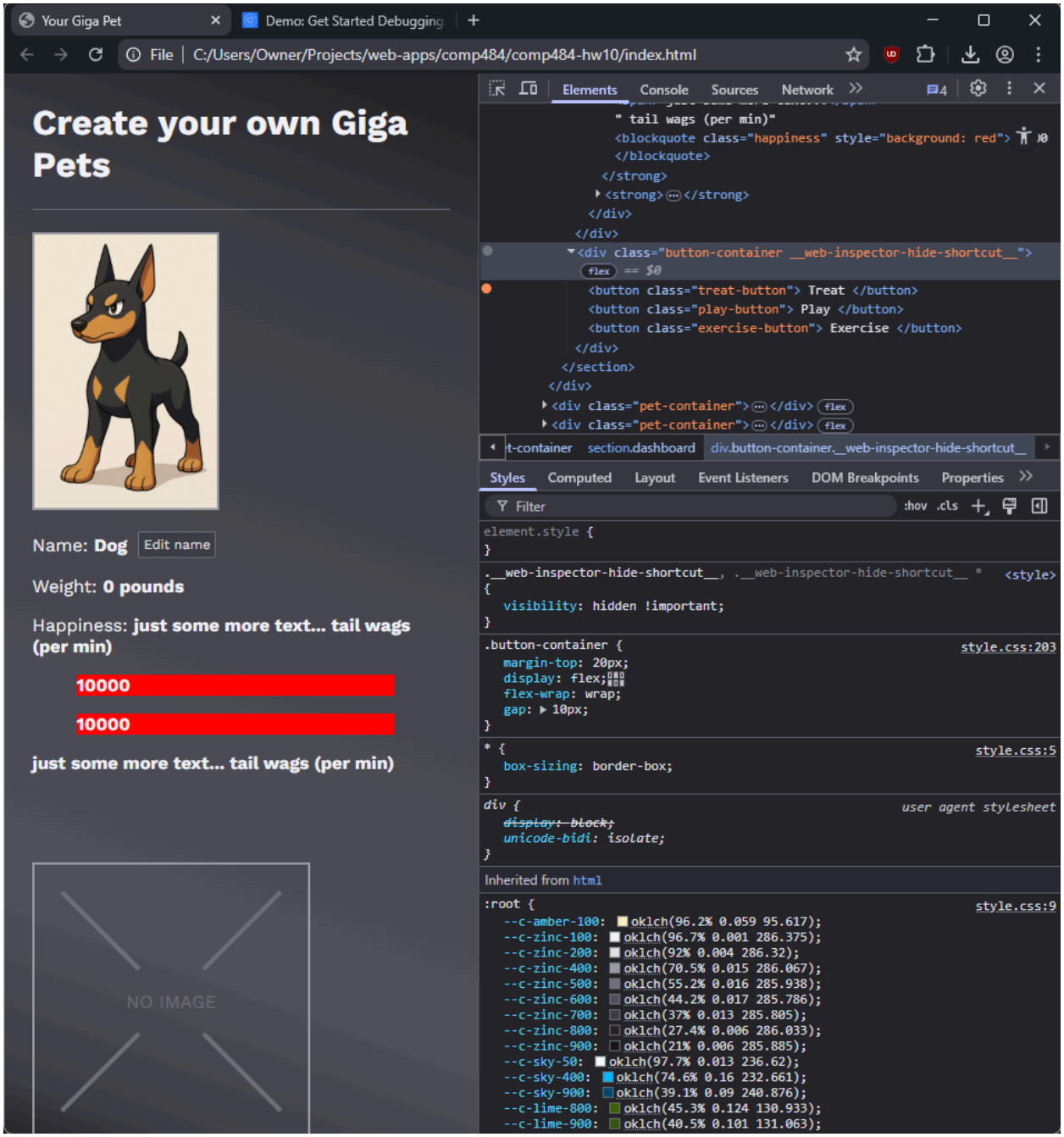
Reorder DOM nodes, Edit the DOM

Here you can see me in the middle of reordering DOM nodes. I selected the text that says “tail wags (per min)” and dragged it under the span that says “just some more text…”.



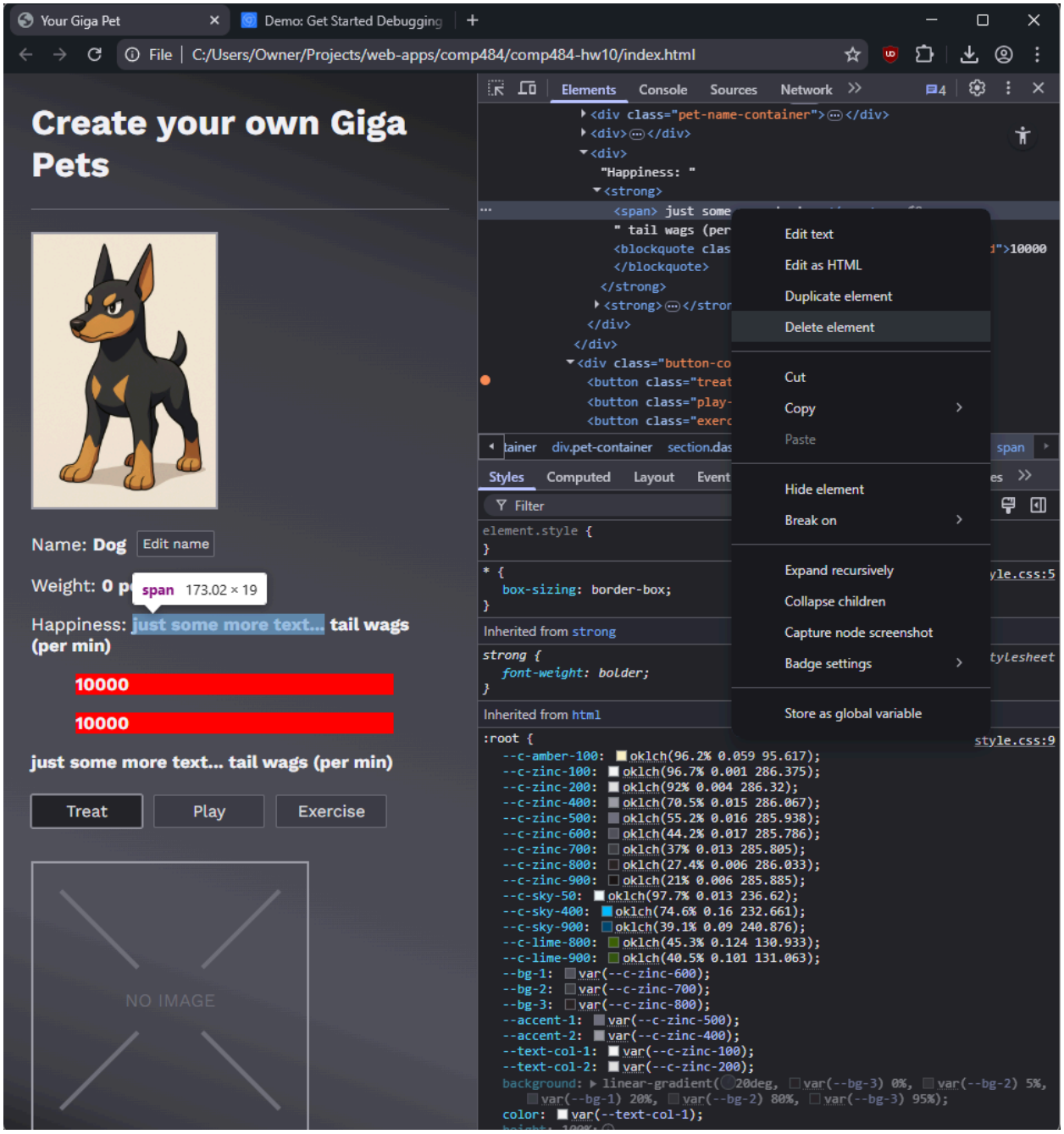
Force State, Edit the DOM

After selecting and right clicking the treat button element, I was able to force a hover state on it. Since the button has extra styling when it is hovered, this option made it so that styling was on regardless of if I was hovering the button or not.



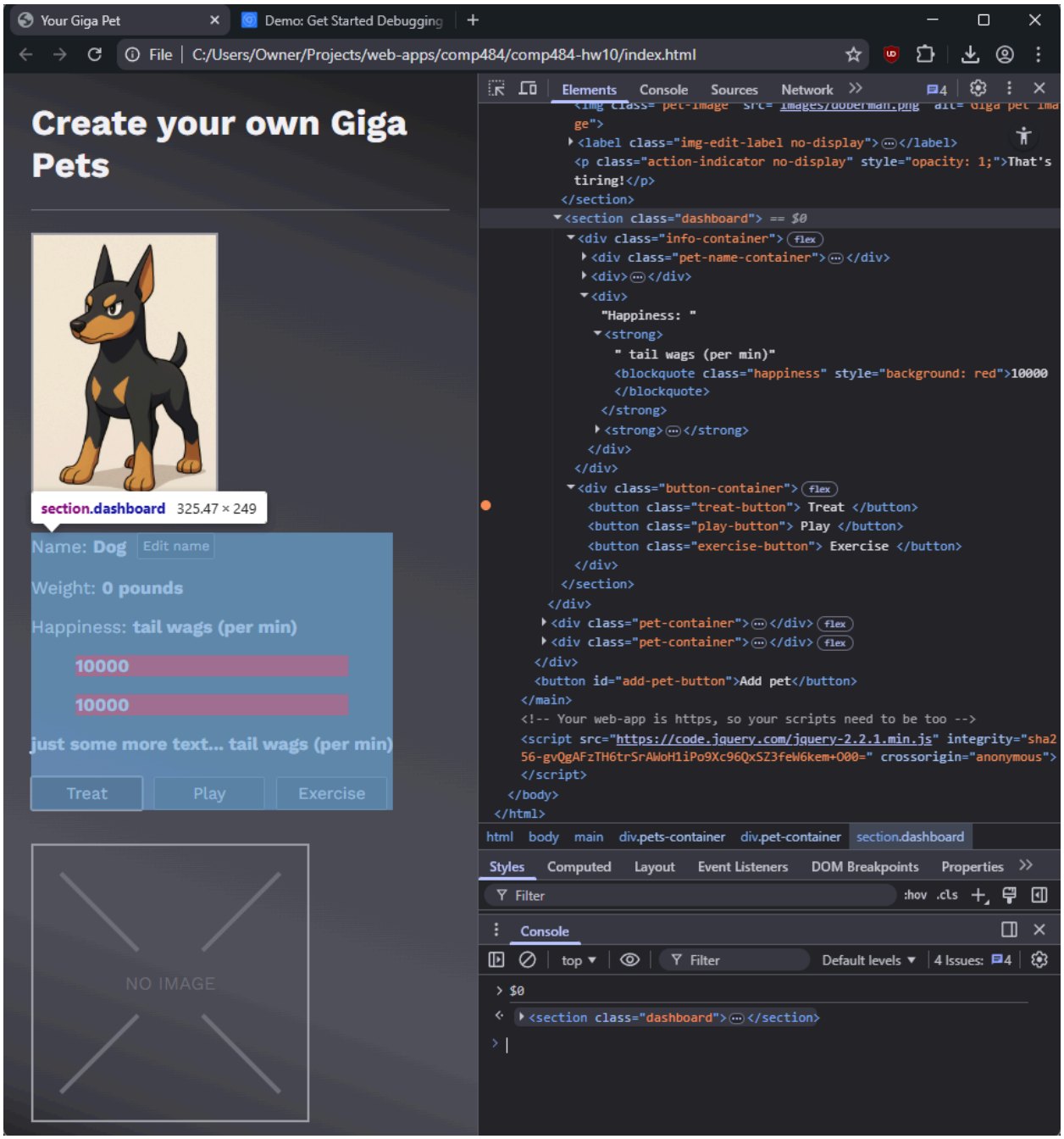
Hide a node, Edit the DOM

Here I selected the button container, which contains the main action buttons for each pet. After hitting the shortcut “H”, the dev tools added a special class to the element which set its visibility to hidden.



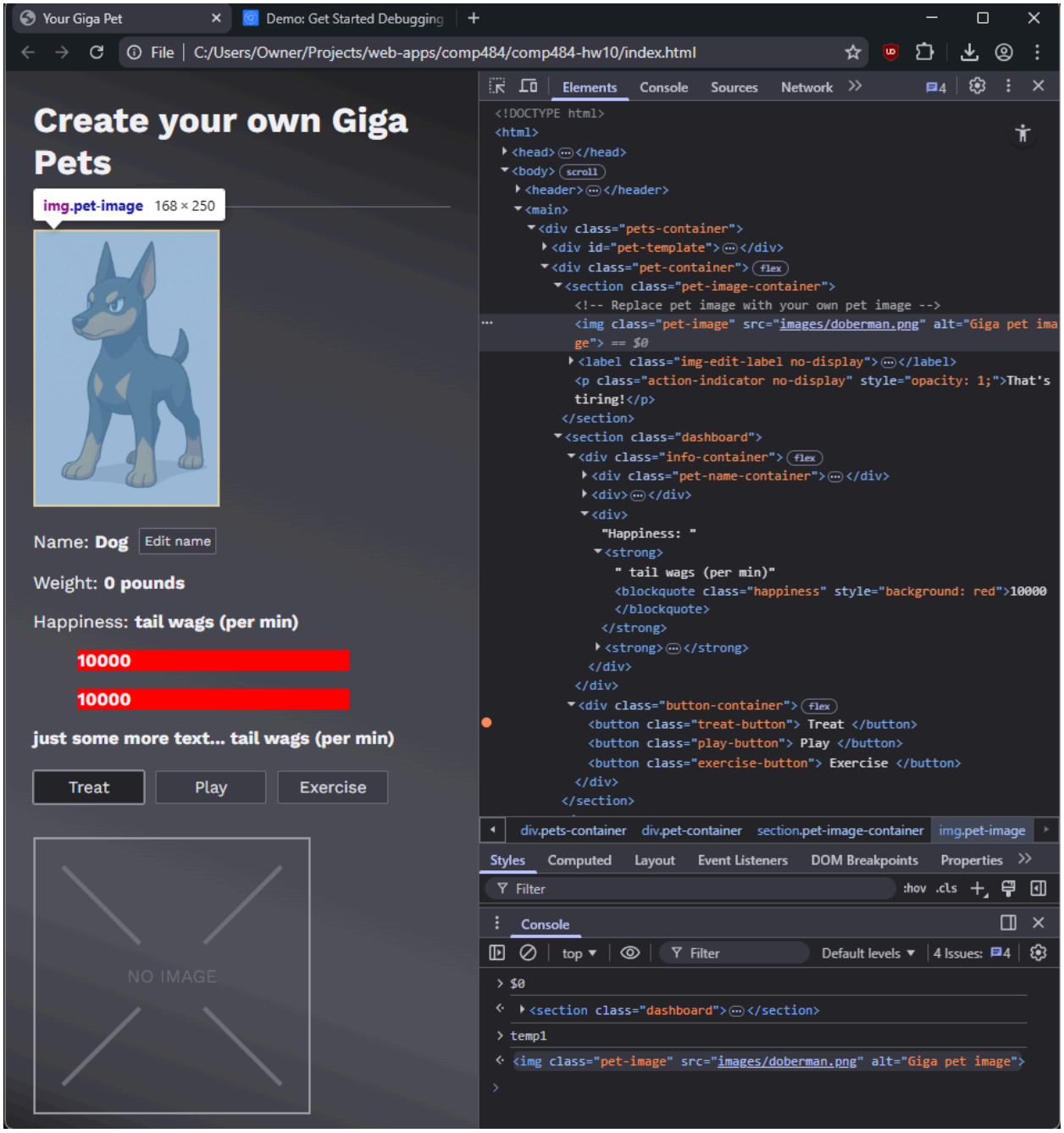
Delete a node, Edit the DOM

After right clicking the first span that says “just some more text…”, I was able to delete it by selecting one of the available options.



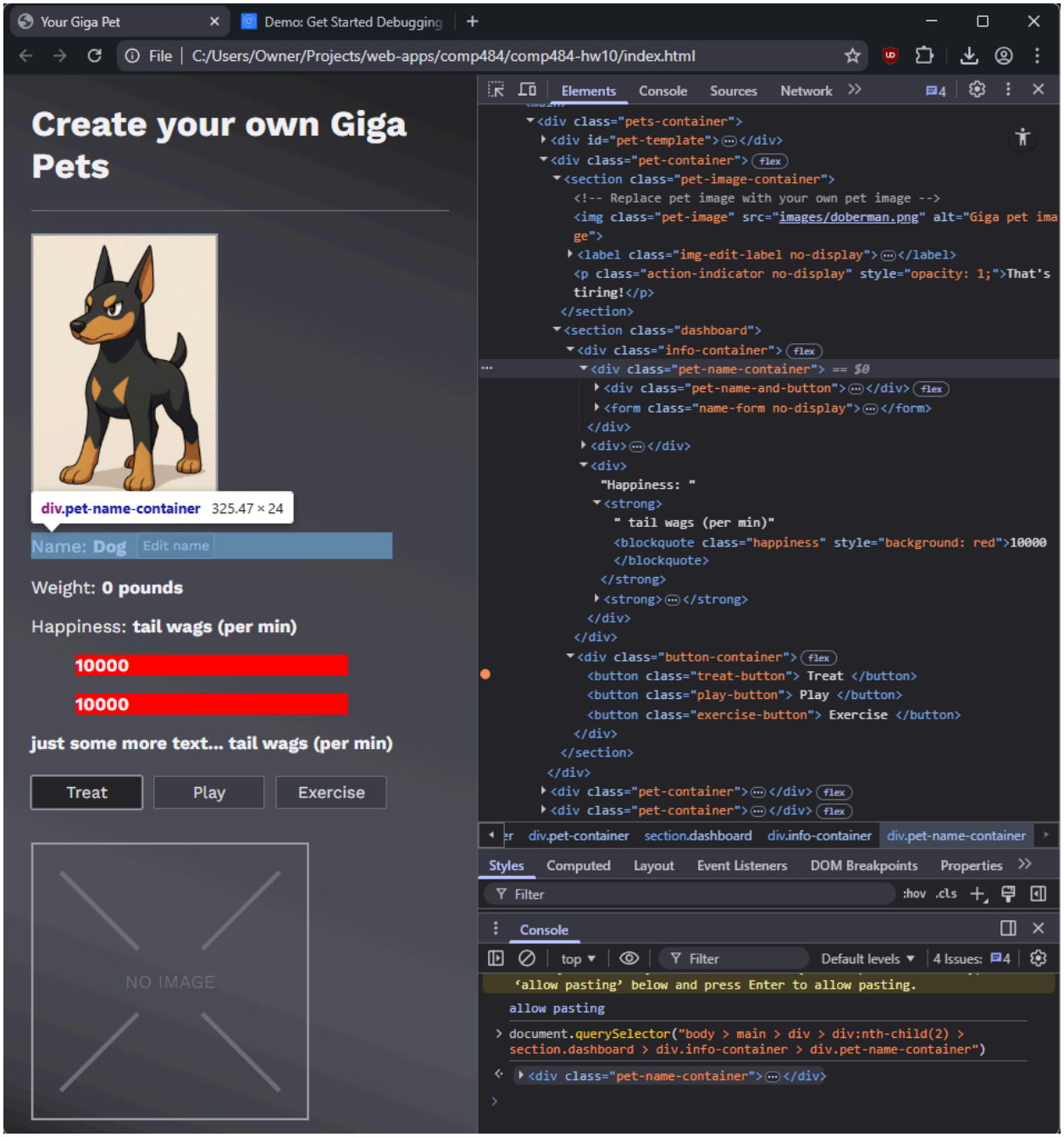
Reference currently selected node with $0, Access nodes in the Console

After selecting a node in the elements tab, I was able to access it in the console using the special "$0" variable.



Store as global variable, Access nodes in the Console

After right clicking the pet-image element and selecting “Store as global variable,” I was able to access the element with the “temp1” global variable.



Copy JS path, Access nodes in the Console

After right clicking the pet-name-container element, I was able to generate a document.querySelector path that references that specific element using the "Copy JS Path" option.