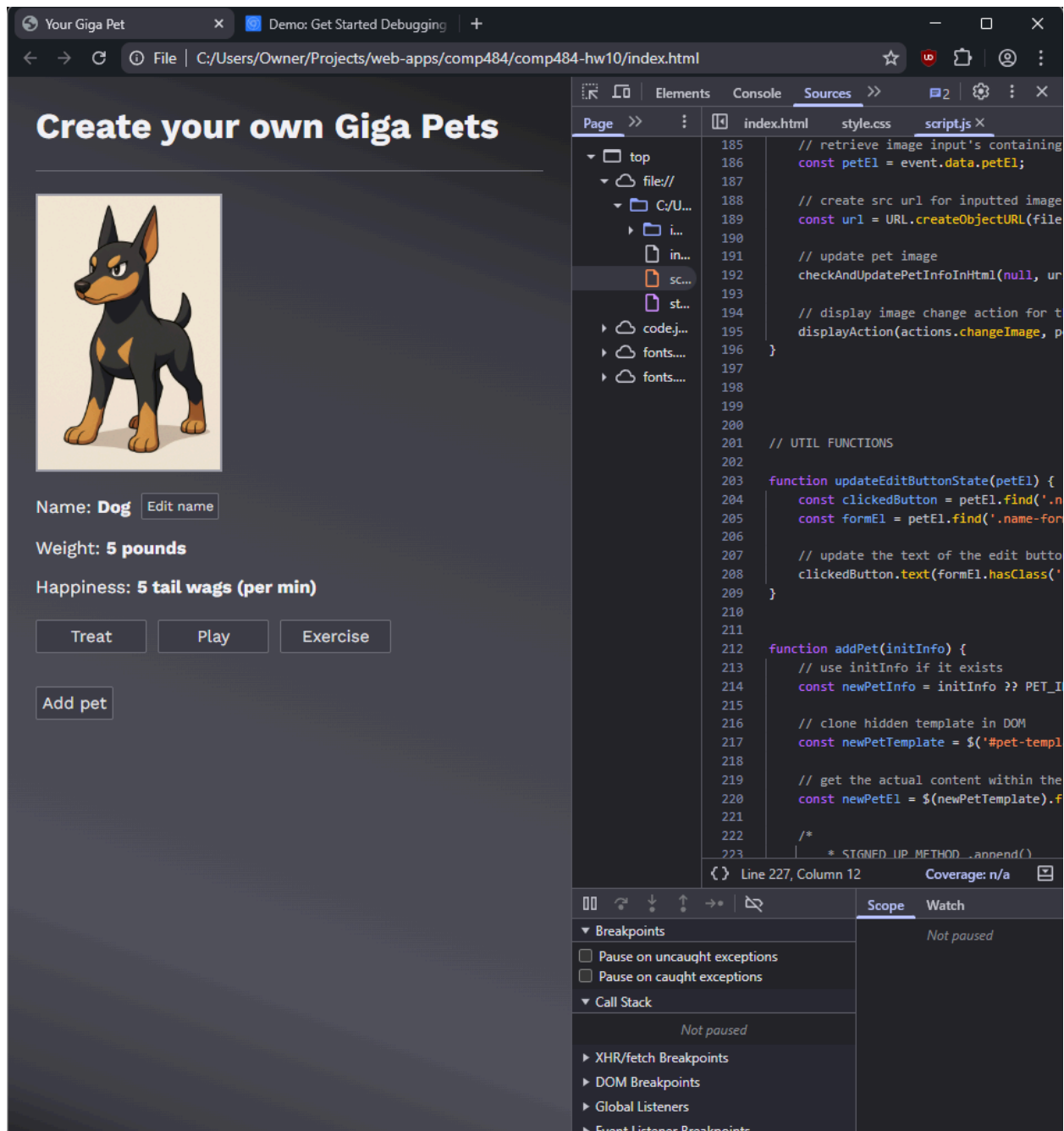


Thomas Scott
3 December 2025

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.

Create your own Giga Pets

That's tiring!

Name: **Dog**

Weight: **0-1 pounds**

Happiness: **9 tail wags (per min)**

```

308
309
310 function checkAndUpdatePetInfoInHtml(name, image, we
311 // get correct object in pets array
312 const petObj = pets.find((p) => p.element.is(petI
313
314 // make sure pet object gets updated correctly
315 checkWeightAndHappinessBeforeUpdating(name, image
316 // update html based on pet object
317 updatePetInfoInHtml(petObj);
318 }
319
320
321 // updates a petObject
322 function checkWeightAndHappinessBeforeUpdating(name,
323 // calculate new weight and happiness based on d
324 const newWeight = petObj.weight + weightDiff;
325 const newHappiness = petObj.happiness + happiness
326
327 // make sure weight and happiness don't go below
328 petObj.weight = newWeight > MIN_WEIGHT ? newWeigh
  
```

Line 324, Column 30 Coverage: n/a

Scope Watch

Local

- this: Window
- happinessDiff: -1
- image: null
- name: null
- newHappiness: <value unavailable>
- newWeight: <value unavailable>
- petObj: Pet
 - element: n.fn.init {0: div.pet-cont
 - happiness: 9
 - image: "images/doberman.png"
 - name: "Dog"
 - weight: "0-1"
 - [[Prototype]]: Object
 - weightDiff: -1

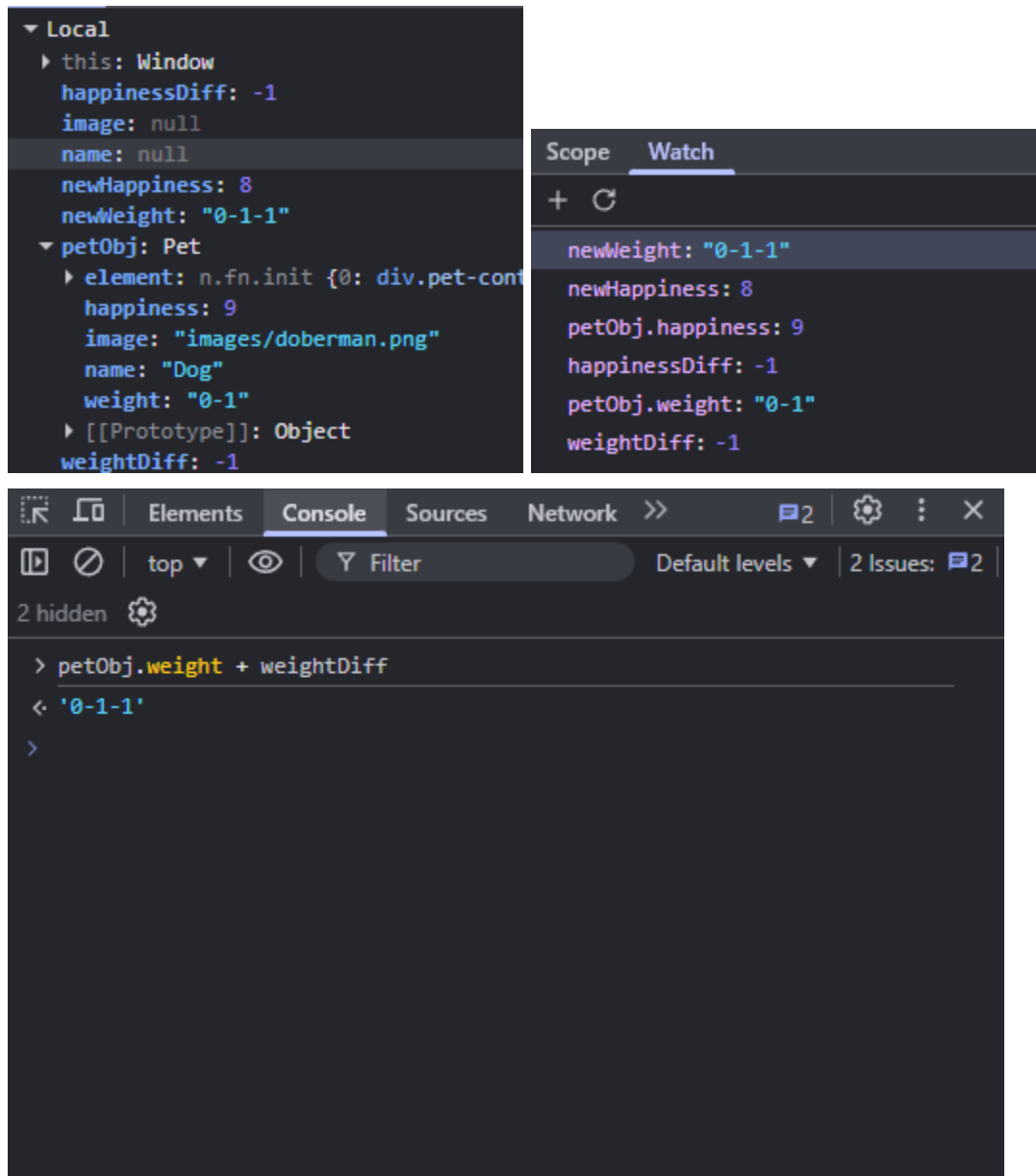
Script

Global

- Window
- \$: f (a,b)
- Pet: f Pet(petInfo)
- addPet: f addPet(initInfo)
- alert: f alert()
- atob: f atob()
- blur: f blur()
- btoa: f btoa()
- caches: CacheStorage {}
- cancelAnimationFrame: f cancelAnimati
- cancelIdleCallback: f cancelIdleCallb
- captureEvents: f captureEvents()
- changedImageEditValue: f changedImage
- checkAndUpdatePetInfoInHtml: f checkA

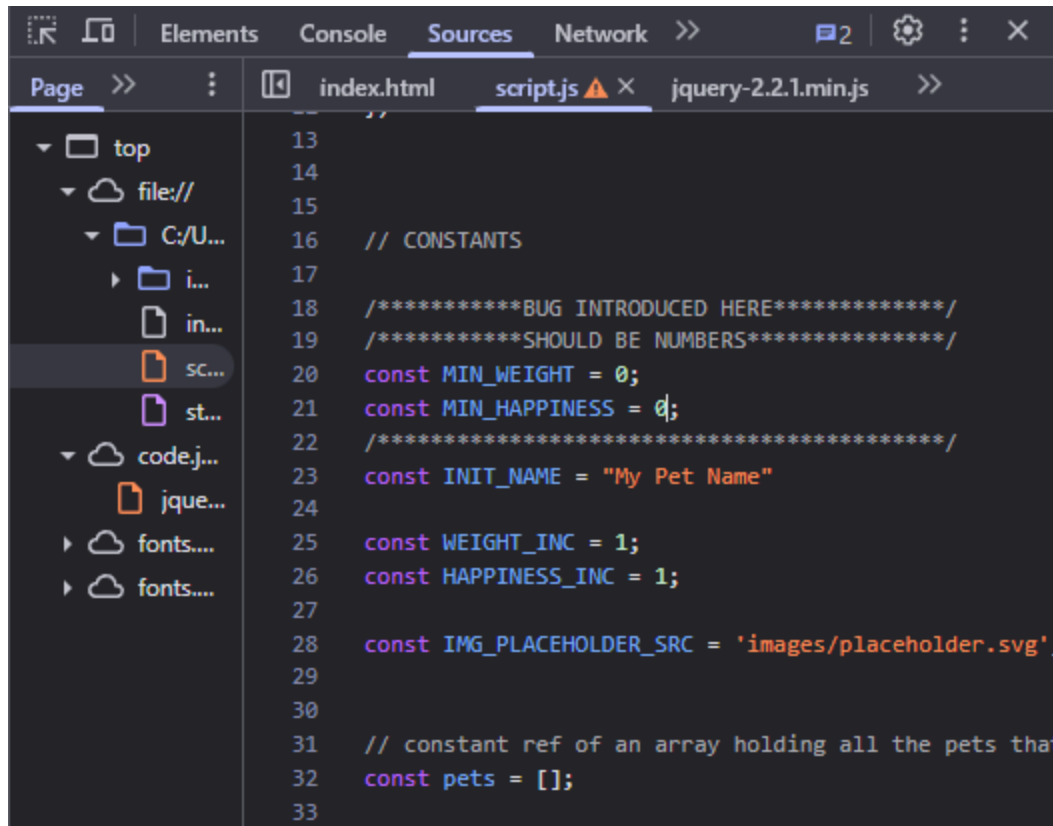
Step 3-5, Debug JavaScript

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.



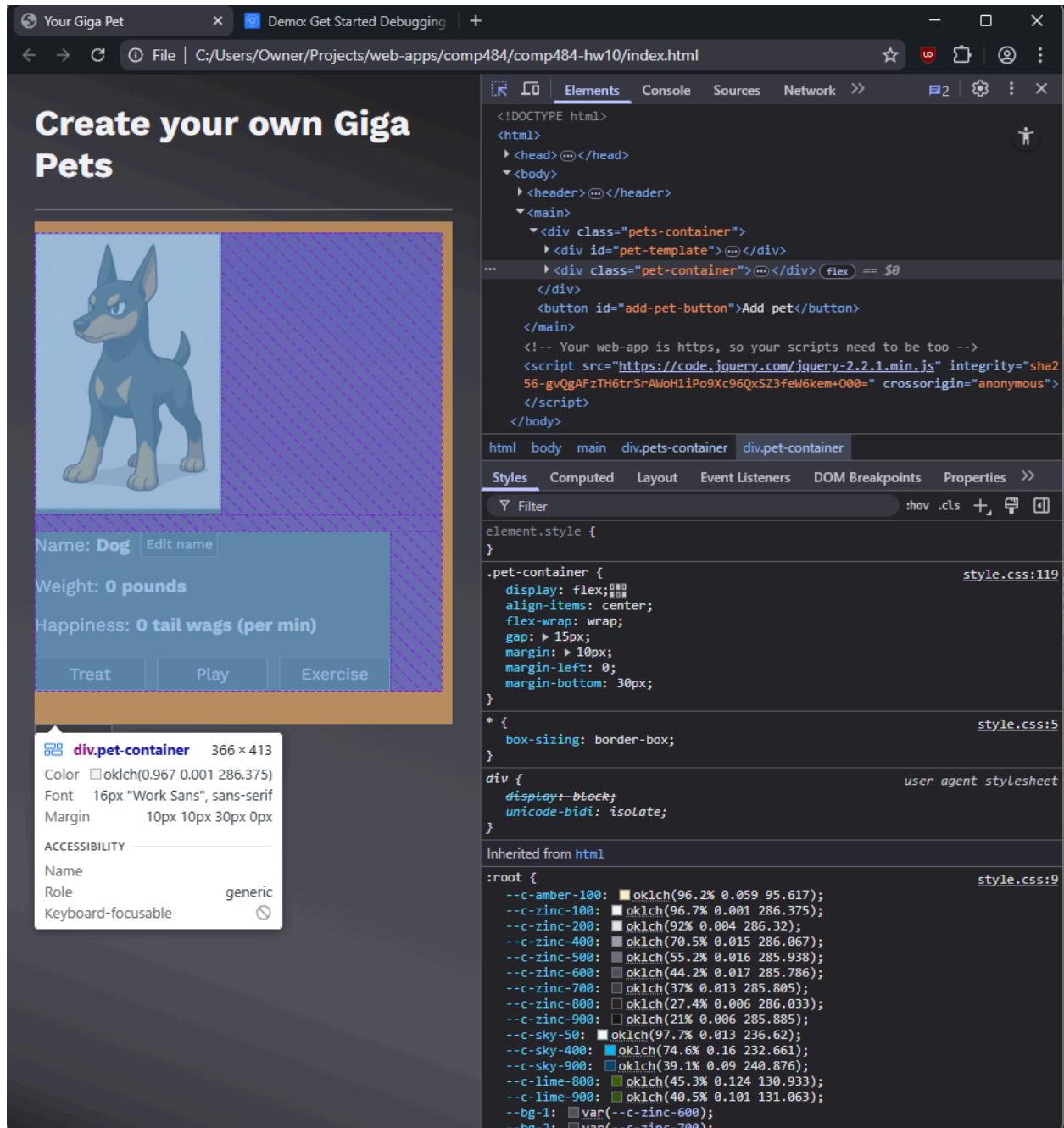
```
// CONSTANTS

/*****BUG INTRODUCED HERE*****/
/*****SHOULD BE NUMBERS*****/
// const MIN_WEIGHT = "0";
// const MIN_HAPPINESS = "0";
/*****BUG FIX HERE*****/
const MIN_WEIGHT = 0;
const MIN_HAPPINESS = 0;
/*****
```

Step 7, Debug JavaScript

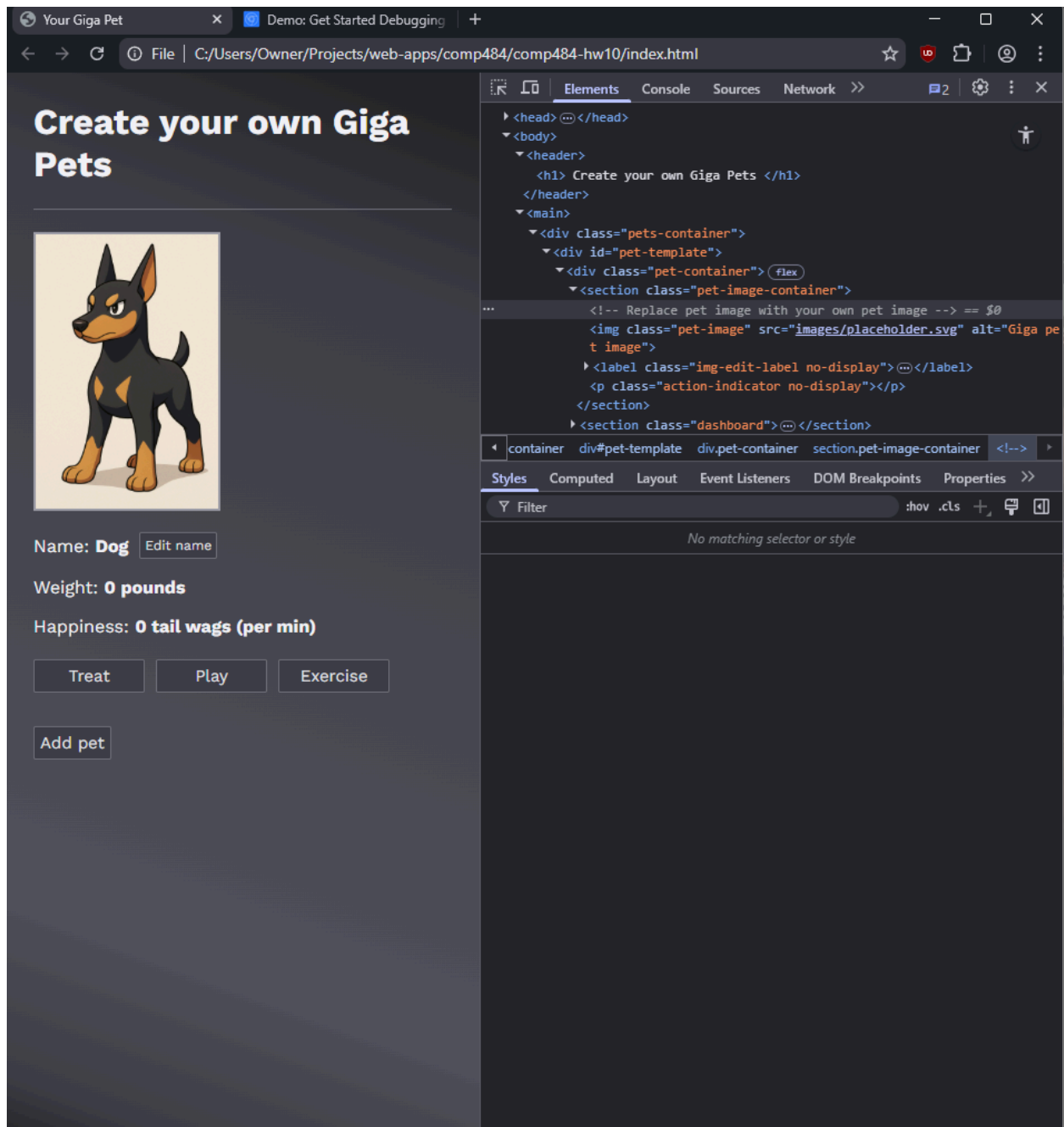
The dev tools allow you to directly change the code that has been sent to the browser.

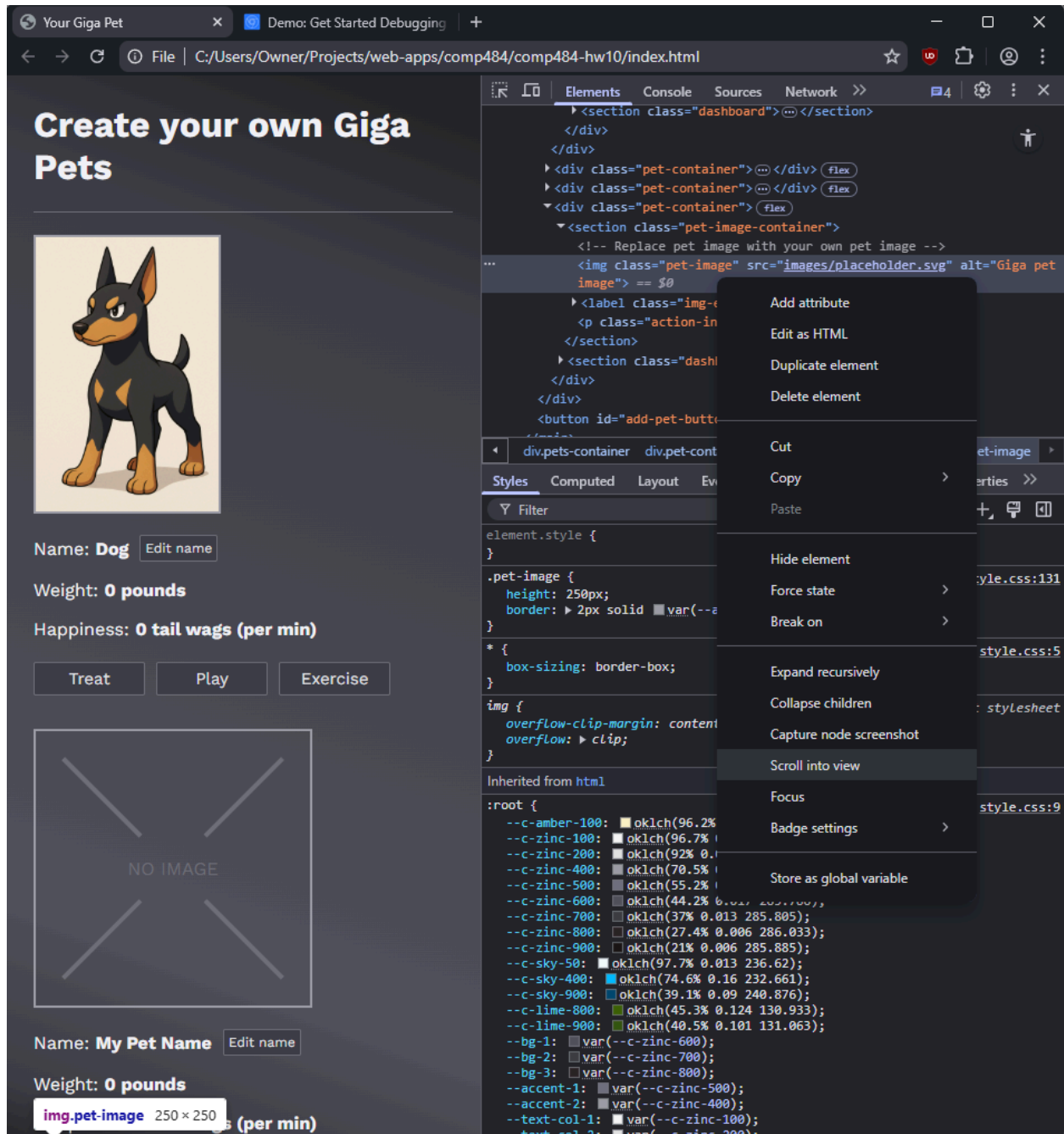
Understanding that the bug existed because weights were being processed as strings, I changed the initial value of weight to be a number rather than a string. However, since the pet object on the screen had already been loaded, this fix within the dev tools didn't work as the change would need to occur 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

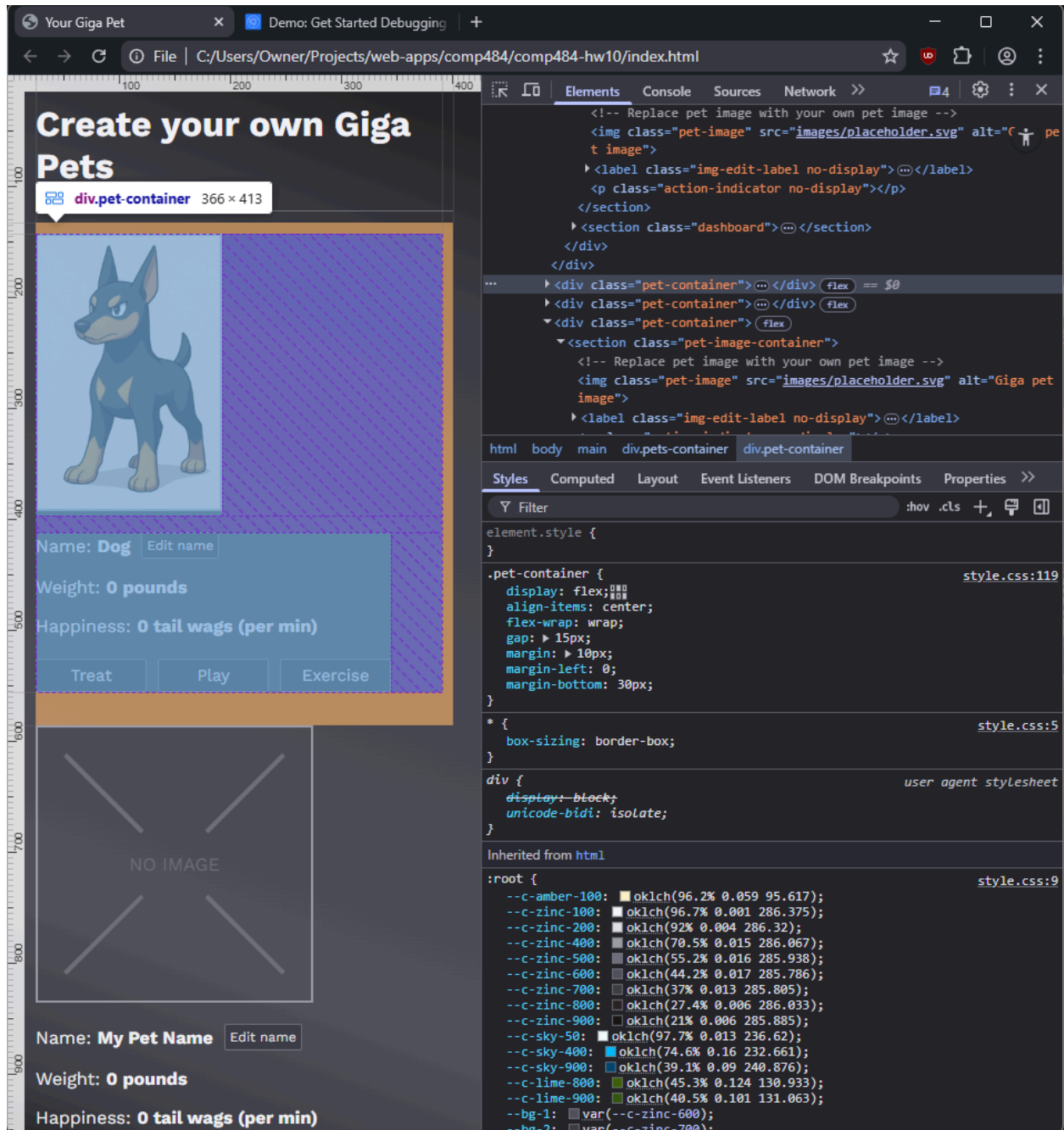
Here you can see me selecting a pet container using the arrow symbol that is on the top left of the dev tools.





Scroll into view, View DOM nodes

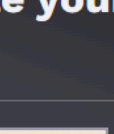
On the bottom left, you can see that I have selected a pet-image that is out of view. In order to quickly get it into view, I selected the “Scroll into view” option when right clicking the element in the DOM.



Show ruler, View DOM nodes

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

Create your own Giga Pets




Name: Dog

Weight: 0 pounds

Happiness: 0 tail wags (per min)

TreatPlayExercise



NO IMAGE

Name: My Pet Name

Weight: 0 pounds

Happiness: 0 tail wags (per min)

ElementsConsoleSourcesNetwork

<div>
 "Happiness: "

 == \$0
 " tail wags (per min)"

 </div>
</div>
> <div class="button-container">...</div> flex
</section>
</div>
</div>
> <div class="pet-container">...</div> flex
> <div class="pet-container">...</div> flex
> <div class="pet-container">...</div> flex

/pet-container section.dashboard div.info-container div strong span.happiness2 of 8X

Happiness

StylesComputedLayoutEvent ListenersDOM BreakpointsProperties

Filter: :hov .cls + - []

element.style { }

* { box-sizing: border-box; style.css:5 }

Inherited from strong

strong { font-weight: bolder; user agent stylesheet }

Inherited from html

:root { style.css:9
--c-amber-100: oklch(96.2% 0.059 95.617);
--c-zinc-100: oklch(96.7% 0.001 286.375);
--c-zinc-200: oklch(92% 0.004 286.32);
--c-zinc-400: oklch(70.5% 0.015 286.067);
--c-zinc-500: oklch(55.2% 0.016 285.938);
--c-zinc-600: oklch(44.2% 0.017 285.786);
--c-zinc-700: oklch(37% 0.013 285.805);
--c-zinc-800: oklch(27.4% 0.006 286.033);
--c-zinc-900: oklch(21% 0.006 285.885);
--c-sky-50: oklch(97.7% 0.013 236.62);
--c-sky-400: oklch(74.6% 0.16 232.661);
--c-sky-900: oklch(39.1% 0.09 240.876);
--c-lime-800: oklch(45.3% 0.124 130.933);
--c-lime-900: oklch(40.5% 0.101 131.063);
--bg-1: var(--c-zinc-600);
--bg-2: var(--c-zinc-700);
--bg-3: var(--c-zinc-800);
--accent-1: var(--c-zinc-500);
--accent-2: var(--c-zinc-400);
--text-col-1: var(--c-zinc-100);
--text-col-2: var(--c-zinc-200);
background: linear-gradient(20deg, var(--bg-3) 0%, var(--bg-2) 5%, var(--bg-1) 20%, var(--bg-2) 80%, var(--bg-3) 95%);
color: var(--text-col-1);

Search nodes, View DOM nodes

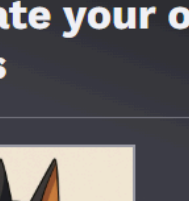
Here I use the Ctrl+F shortcut to search for keywords within the DOM. I searched for “Happiness,” and one of the results that I was given was a span with a class of “happiness.”

Your Giga Pet

Demo: Get Started Debugging

C:\Users\Owner\Projects\web-apps\comp484-hw10\index.html

Create your own Giga Pets



Name: Dog Edit name

Weight: 0 poundsspan.happiness 48.13 × 19

Happiness: 10000 tail wags (per min)

TreatPlayExercise

NO IMAGE

Name: My Pet Name Edit name

Weight: 0 pounds

Happiness: 0 tail wags (per min)

ElementsConsoleSourcesNetwork

<section class= "pet-image-container"></section>
<section class= "dashboard">
 <div class= "info-container"> flex
 <div class= "pet-name-container"></div>
 <div></div>
 <div>
 "Happiness: "

 10000 == \$0
 " tail wags (per min)"

 </div>
 <div class= "button-container"></div> flex
 </section>
</div>

pet-container section.dashboard div.info-container div strong span.happiness

Happiness

StylesComputedLayoutEvent ListenersDOM BreakpointsProperties

Filter :hov .cls +, , ,

element.style {
}

* {
 box-sizing: border-box;
}

Inherited from strong

strong {
 font-weight: bolder;
}

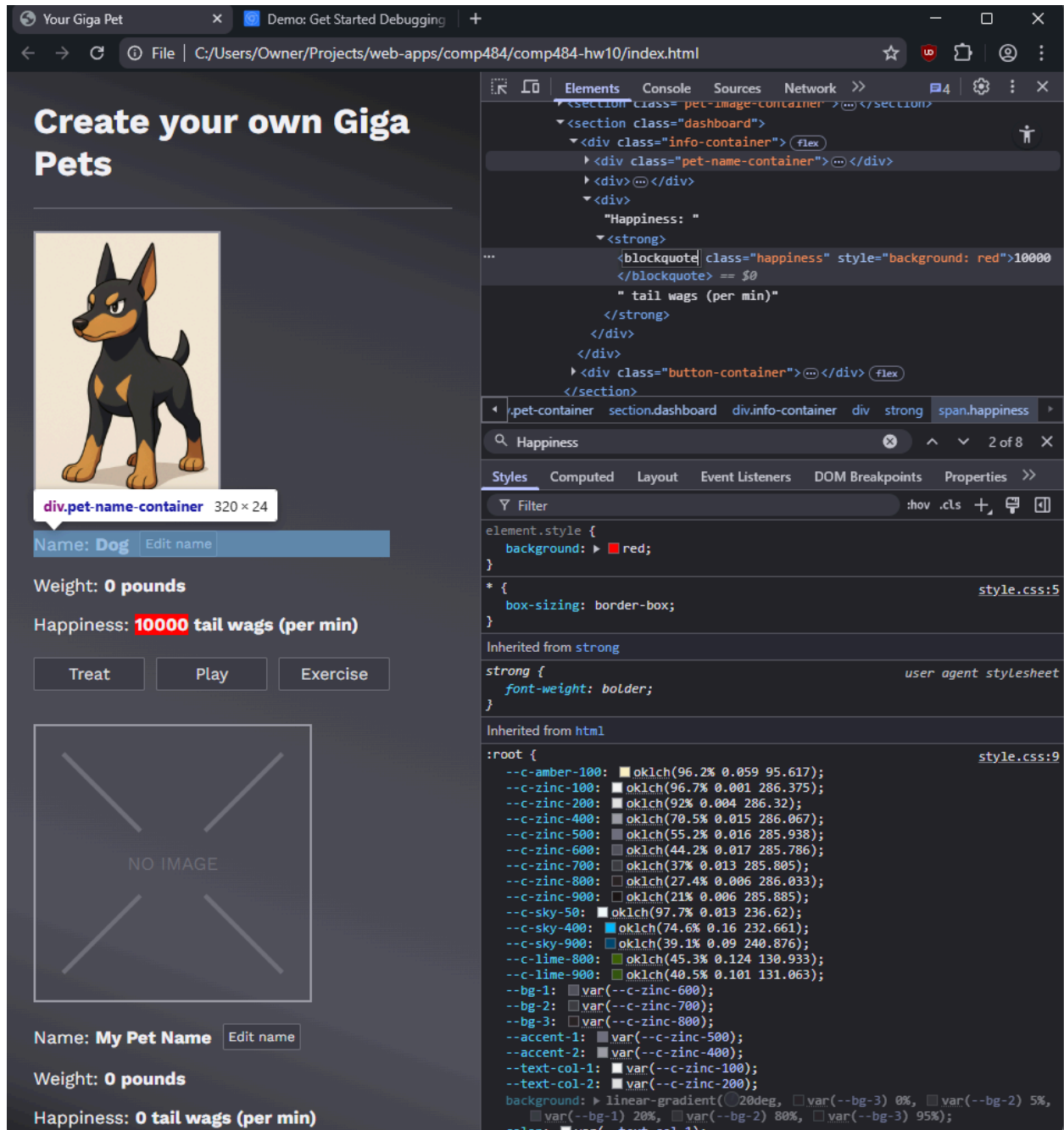
Inherited from html

:root {
 --c-amber-100: oklch(96.2% 0.059 95.617);
 --c-zinc-100: oklch(96.7% 0.001 286.375);
 --c-zinc-200: oklch(92% 0.004 286.32);
 --c-zinc-400: oklch(70.5% 0.015 286.067);
 --c-zinc-500: oklch(55.2% 0.016 285.938);
 --c-zinc-600: oklch(44.2% 0.017 285.786);
 --c-zinc-700: oklch(37% 0.013 285.805);
 --c-zinc-800: oklch(27.4% 0.006 286.033);
 --c-zinc-900: oklch(21% 0.006 285.885);
 --c-sky-50: oklch(97.7% 0.013 236.62);
 --c-sky-400: oklch(74.6% 0.16 232.661);
 --c-sky-900: oklch(39.1% 0.09 240.876);
 --c-lime-800: oklch(45.3% 0.124 130.933);
 --c-lime-900: oklch(40.5% 0.101 131.063);
 --bg-1: var(--c-zinc-600);
 --bg-2: var(--c-zinc-700);
 --bg-3: var(--c-zinc-800);
 --accent-1: var(--c-zinc-500);
 --accent-2: var(--c-zinc-400);
 --text-col-1: var(--c-zinc-100);
 --text-col-2: var(--c-zinc-200);
 background: linear-gradient(20deg, var(--bg-3) 0%, var(--bg-2) 5%, var(--bg-1) 20%, var(--bg-2) 80%, var(--bg-3) 95%);
 color: var(--text-col-1);
}

Edit content, Edit the DOM

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


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.

Create your own Giga Pets



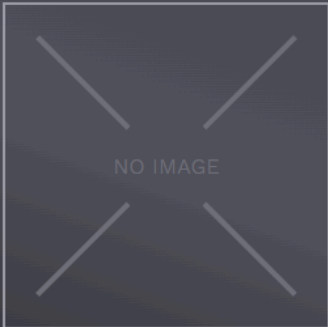
Name: **Dog**

Weight: **0 pounds**

10000 240 × 19

10000

tail wags (per min)



Name: **My Pet Name**

```

<section class="pet-image-container">...</section>
<section class="dashboard">
  <div class="info-container"> flex
    <div class="pet-name-container">...</div>
    <div>...</div>
  <div>
    "Happiness: "
    <strong>
      <blockquote class="happiness" style="background:
        red">10000</blockquote>
      <span> just some more text...</span>
    </strong>
    " tail wags (per min)"
  </div>
</div>

```

Happiness

Styles Computed Layout Event Listeners DOM Breakpoints Properties >>

Filter :hov .cls + -

```

element.style {
  background: red;
}
* {
  box-sizing: border-box;
}
blockquote {
  display: block;
  margin-block-start: 1em;
  margin-block-end: 1em;
  margin-inline-start: 40px;
  margin-inline-end: 40px;
  unicode-bidi: isolate;
}

```

Inherited from strong

```

strong {
  font-weight: bolder;
}

```

Inherited from html

```

:root {
  --c-amber-100: oklch(96.2% 0.059 95.617);
  --c-zinc-100: oklch(96.7% 0.001 286.375);
  --c-zinc-200: oklch(92% 0.004 286.32);
  --c-zinc-400: oklch(70.5% 0.015 286.067);
  --c-zinc-500: oklch(55.2% 0.016 285.938);
  --c-zinc-600: oklch(44.2% 0.017 285.786);
  --c-zinc-700: oklch(37% 0.013 285.805);
  --c-zinc-800: oklch(27.4% 0.006 286.033);
  --c-zinc-900: oklch(21% 0.006 285.885);
  --c-sky-50: oklch(97.7% 0.013 236.62);
  --c-sky-400: oklch(74.6% 0.16 232.661);
  --c-sky-900: oklch(39.1% 0.09 240.876);
  --c-lime-800: oklch(45.3% 0.124 130.933);
  --c-lime-900: oklch(40.5% 0.101 131.063);
  --bg-1: var(--c-zinc-600);
}

```

Edit as HTML, Edit the DOM

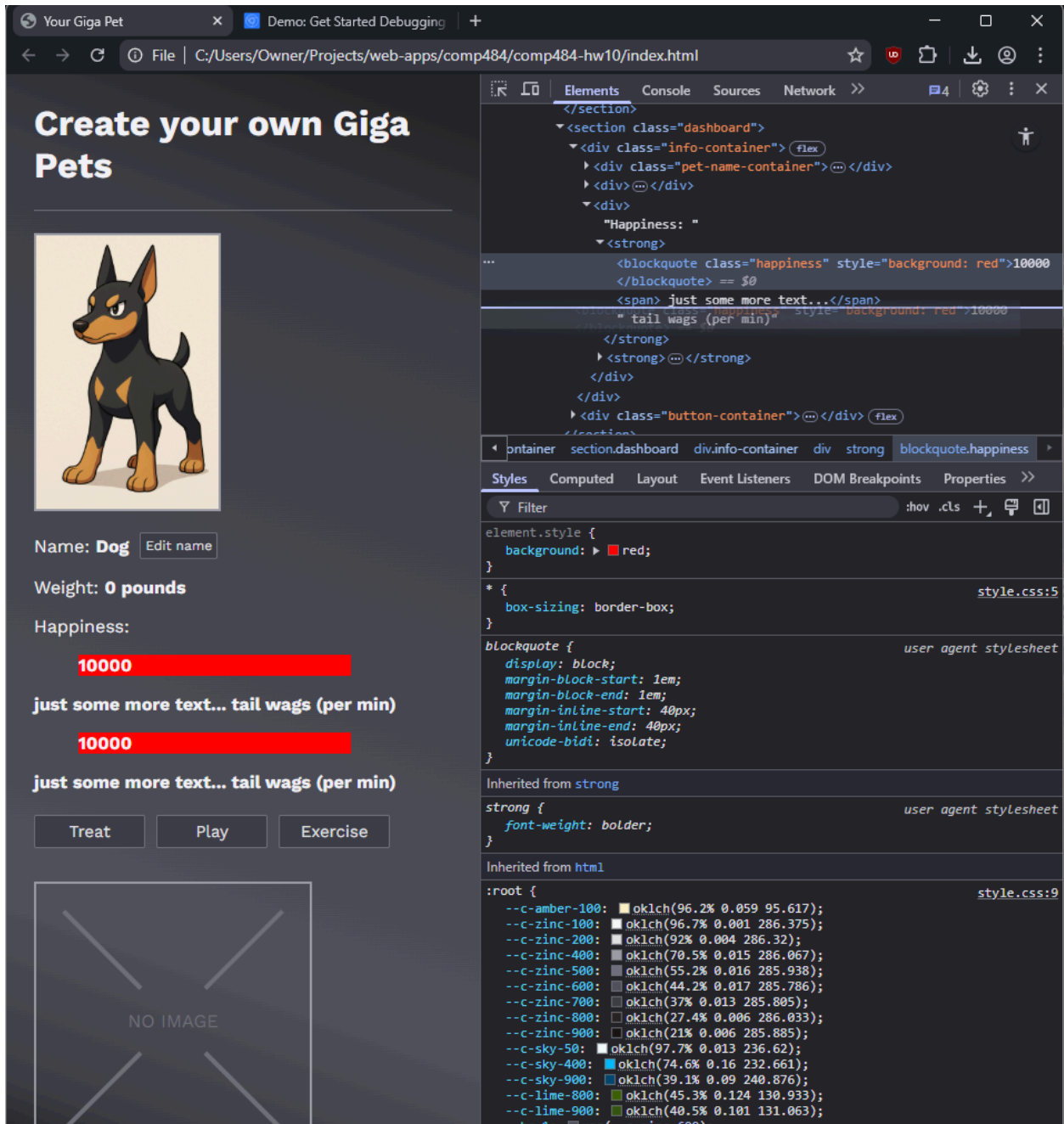
After right clicking on the blockquote, I was able to select the option to edit it as HTML. What this did was open up a mini editor in which I was able to freely write HTML. As you can see, I added another span element after the blockquote using this feature.

The screenshot shows a web browser window with the title "Your Giga Pet" and a tab "Demo: Get Started Debugging". The address bar shows the file path: `C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html`. The web page displays a form to create a pet. It includes a dog image, a name input field with "Dog" and an "Edit name" button, a happiness meter showing "10000", and buttons for "Treat", "Play", and "Exercise". A placeholder for a pet image shows "NO IMAGE".

The browser's developer tools are open, showing the DOM tree. The selected element is a `strong` element with the text "Happiness: ". A context menu is open over this element, showing options like "Add attribute", "Edit as HTML", "Duplicate element", "Delete element", "Cut", "Copy", "Paste", "Hide element", "Force state", "Break on", "Expand recursively", "Collapse children", "Capture node screenshot", "Scroll into view", "Focus", "Badge settings", and "Store as global variable". The "Duplicate element" option is highlighted.

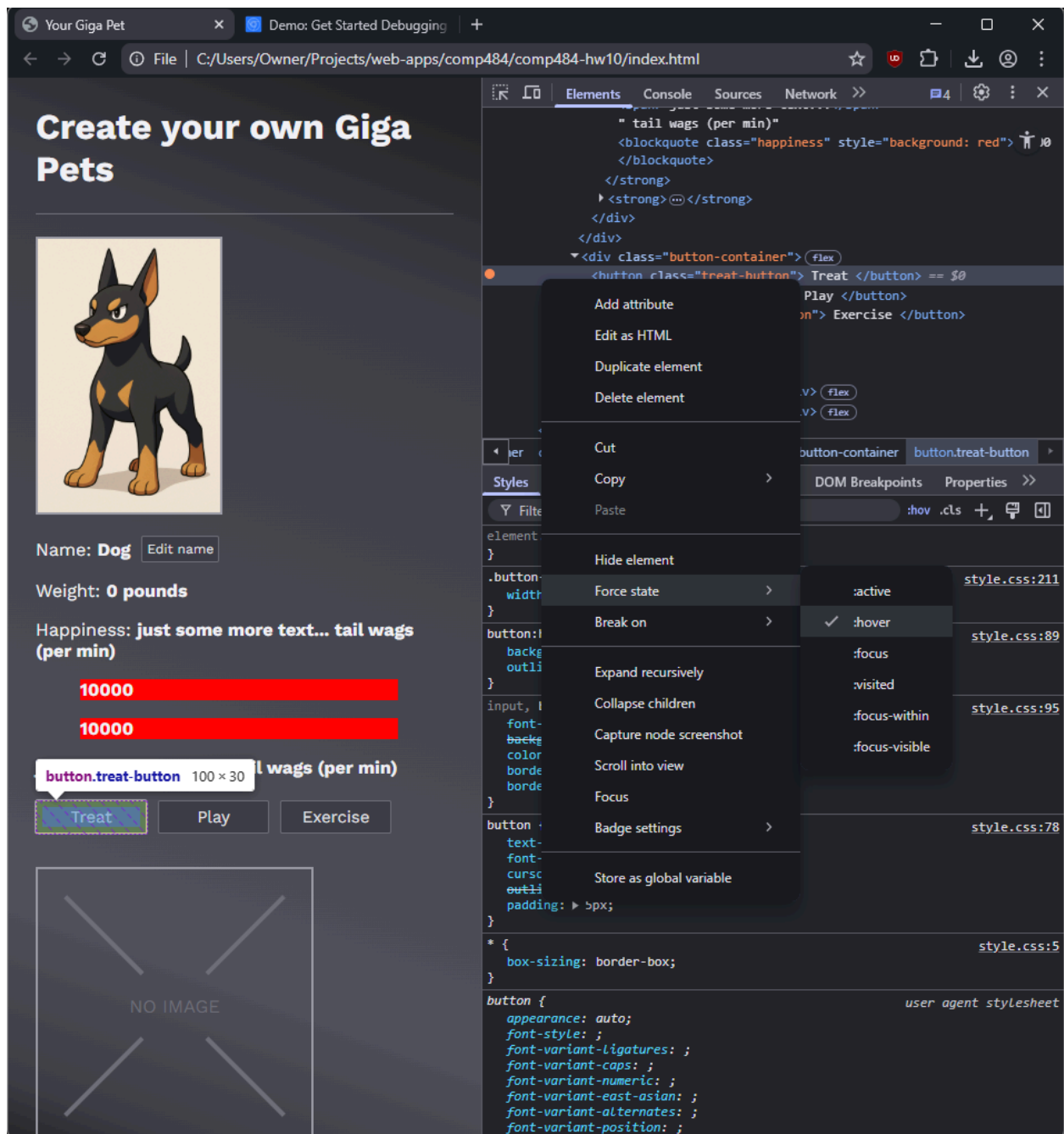
Duplicate a node, Edit the DOM

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.



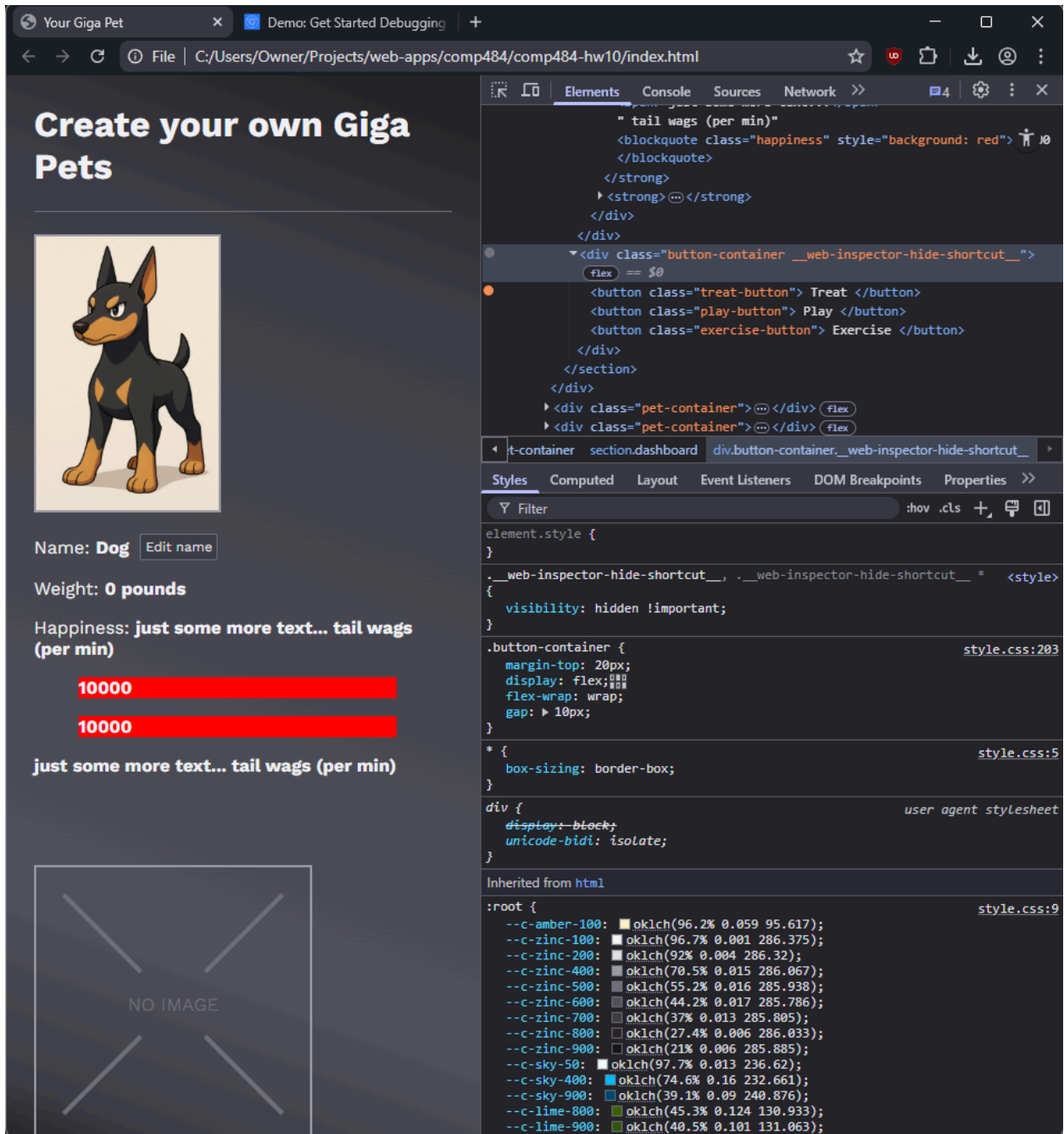
Reorder DOM nodes, Edit the DOM

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.

Create your own Giga Pets



Name: **Dog**

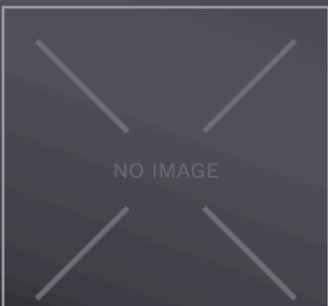
Weight: **0** 173.02 × 19

Happiness: **just some more text...** tail wags (per min)

10000

10000

just some more text... tail wags (per min)



Developer Tools:

- Elements:**

```

<div class="pet-name-container"></div>
<div></div>
<div>
  "Happiness: "
  <strong>
    <span> just some
    " tail wags (per
    <blockquote clas
    </blockquote>
  </strong>
  <strong></strong>
</div>
<div class="button-co
  <button class="treat
  <button class="play-
  <button class="exerc

```
- Styles:**

```

element.style {
}
* {
  box-sizing: border-box;
}
Inherited from strong
strong {
  font-weight: bold;
}
Inherited from html
:root {
  --c-amber-100: oklch(96.2% 0.059 95.617);
  --c-zinc-100: oklch(96.7% 0.001 286.375);
  --c-zinc-200: oklch(92% 0.004 286.32);
  --c-zinc-400: oklch(70.5% 0.015 286.067);
  --c-zinc-500: oklch(55.2% 0.016 285.938);
  --c-zinc-600: oklch(44.2% 0.017 285.786);
  --c-zinc-700: oklch(37% 0.013 285.805);
  --c-zinc-800: oklch(27.4% 0.006 286.033);
  --c-zinc-900: oklch(21% 0.006 285.885);
  --c-sky-50: oklch(97.7% 0.013 236.62);
  --c-sky-400: oklch(74.6% 0.16 232.661);
  --c-sky-900: oklch(39.1% 0.09 240.876);
  --c-lime-800: oklch(45.3% 0.124 130.933);
  --c-lime-900: oklch(40.5% 0.101 131.063);
  --bg-1: var(--c-zinc-600);
  --bg-2: var(--c-zinc-700);
  --bg-3: var(--c-zinc-800);
  --accent-1: var(--c-zinc-500);
  --accent-2: var(--c-zinc-400);
  --text-col-1: var(--c-zinc-100);
  --text-col-2: var(--c-zinc-200);
  background: linear-gradient(20deg, var(--bg-3) 0%, var(--bg-2) 5%,
    var(--bg-1) 20%, var(--bg-2) 80%, var(--bg-3) 95%);
  color: var(--text-col-1);

```
- Console:**

```

style.css:5
style.css:9

```

Delete a node, Edit the DOM

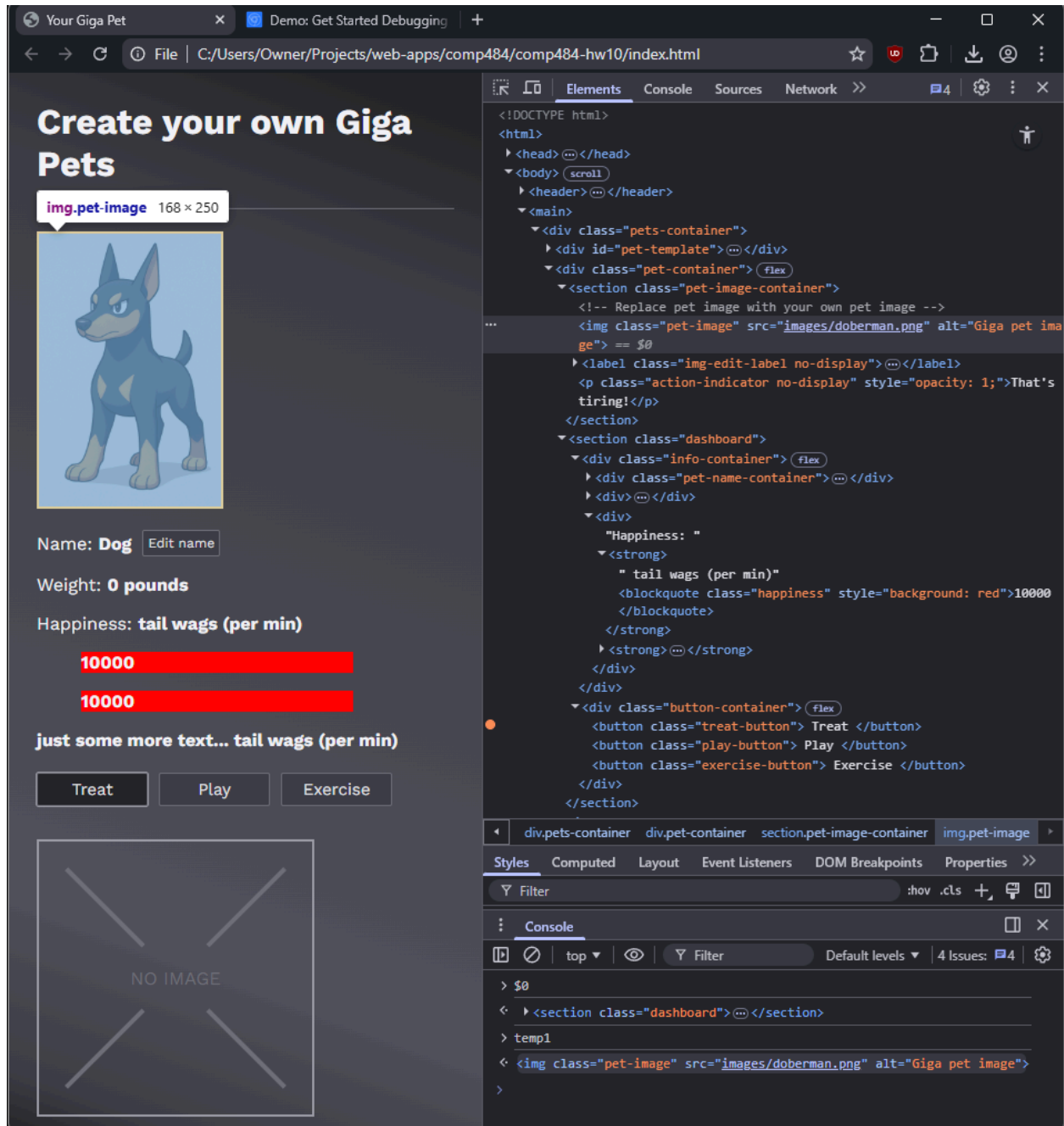
After right clicking the first span that says “just some more text...”, I was able to delete it by selecting one of the available options.

The screenshot shows a web browser window with the URL `C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html`. The page title is "Your Giga Pet". The main heading is "Create your own Giga Pets". Below the heading is a dog image with a tooltip showing `section.dashboard` and dimensions `325.47 x 249`. The form includes fields for Name (Dog), Weight (0 pounds), and Happiness (tail wags (per min)). The Happiness field has a red progress bar and a value of 10000. Below the form are buttons for Treat, Play, and Exercise. At the bottom is a placeholder for a pet image with the text "NO IMAGE".

The developer console is open, showing the Elements tab. The selected node is `<section class="dashboard"> == $0`. The console also shows the HTML structure of the page, including the `<main>` and `<script>` tags.

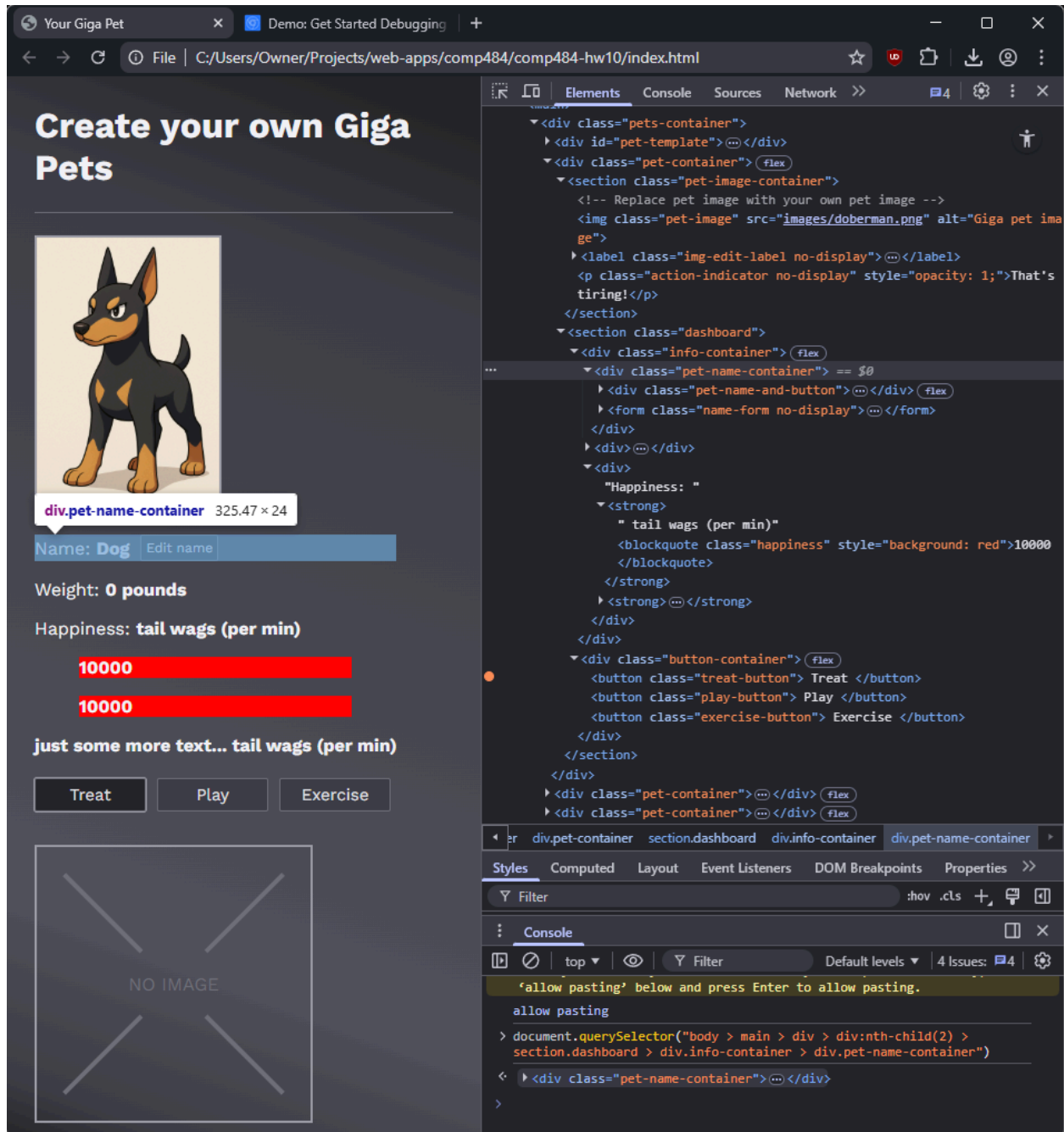
Reference currently selected node with \$0, Access nodes in the Console

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.