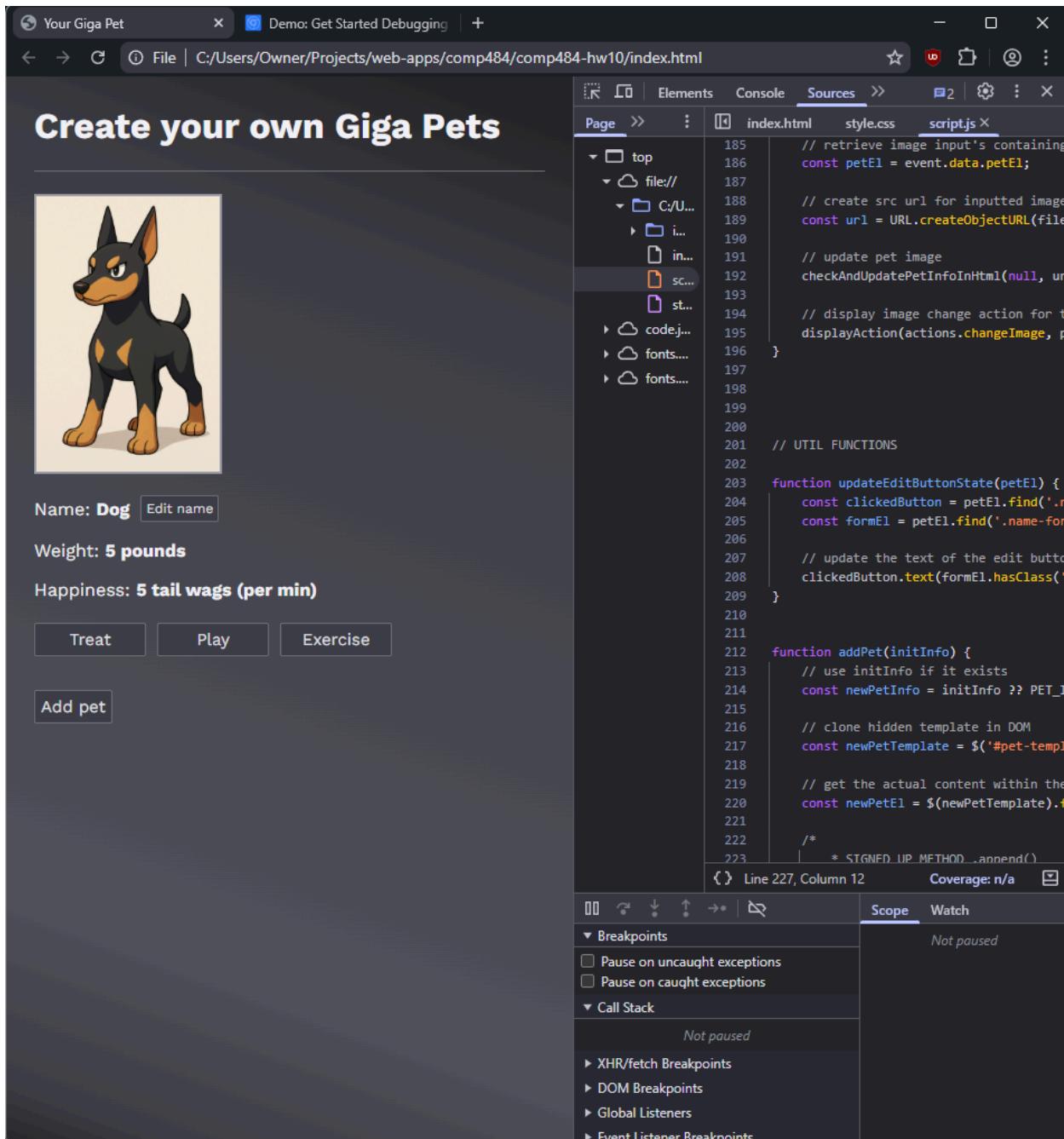


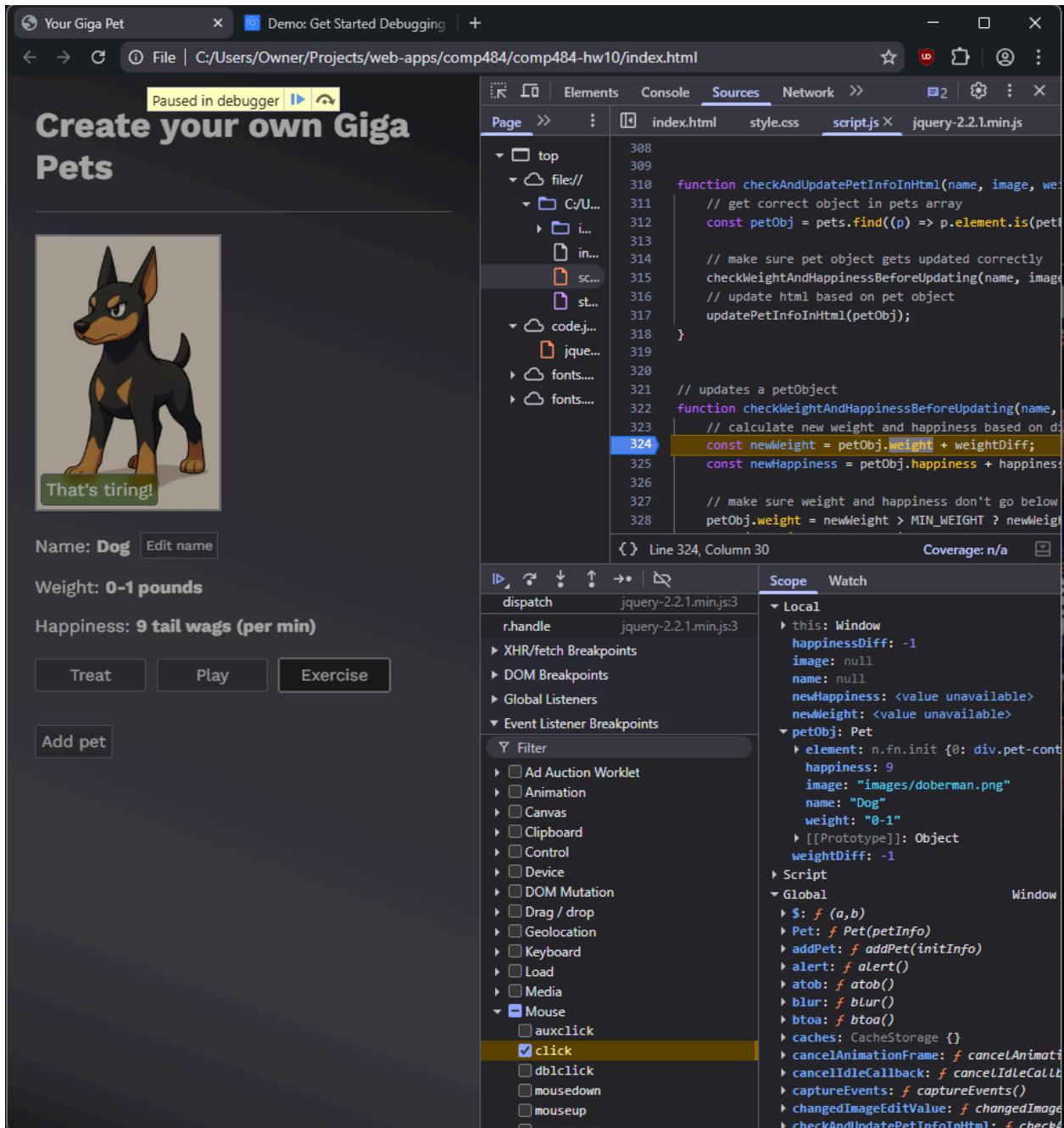
Thomas Scott
3 December 2025

Homework 10 - Chrome Dev Tools



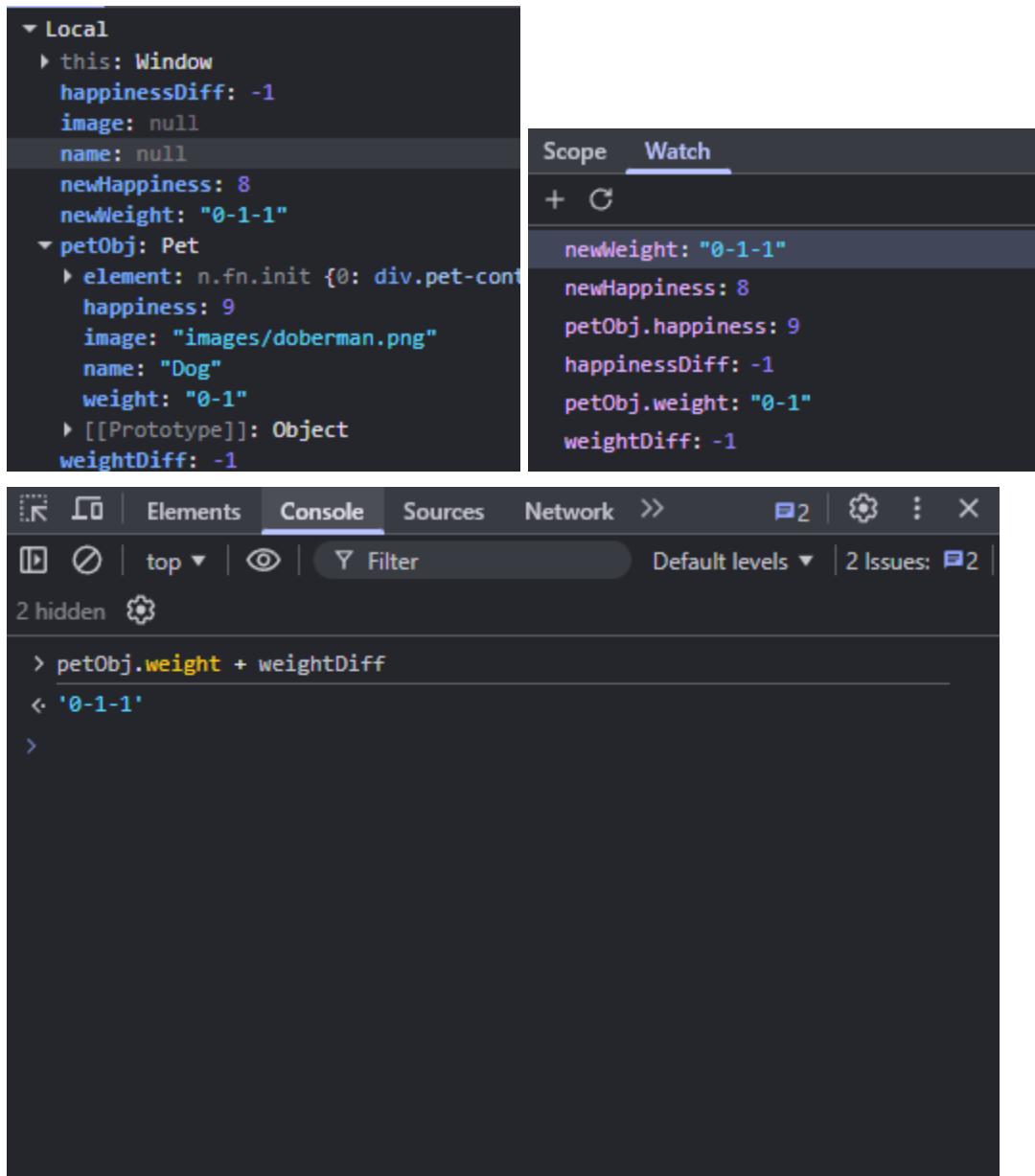
Step 1-2, Debug JavaScript

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

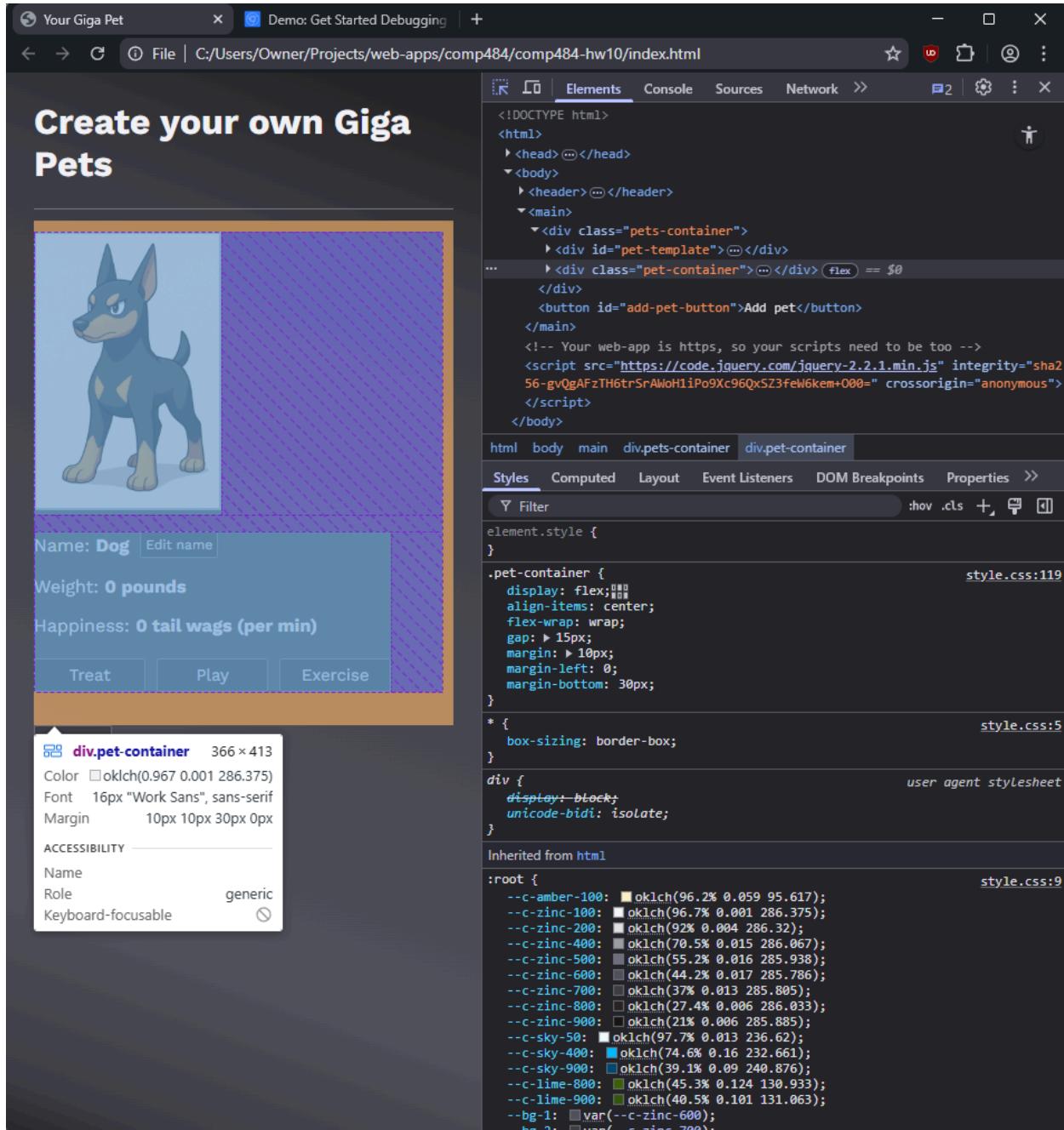
In finding the function that seemed to be problematic, I checked the values of variables in a few ways. The first method I used was to just look at the variables in the local scope at the breakpoint. The second method was setting watch variables to see how specific variables change over time. With these 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.

The screenshot shows the Chrome DevTools Sources tab. The left sidebar lists files: index.html, script.js (which has a yellow warning icon), and jquery-2.2.1.min.js. The right pane shows the code for script.js:

```
// 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;
// *****
```

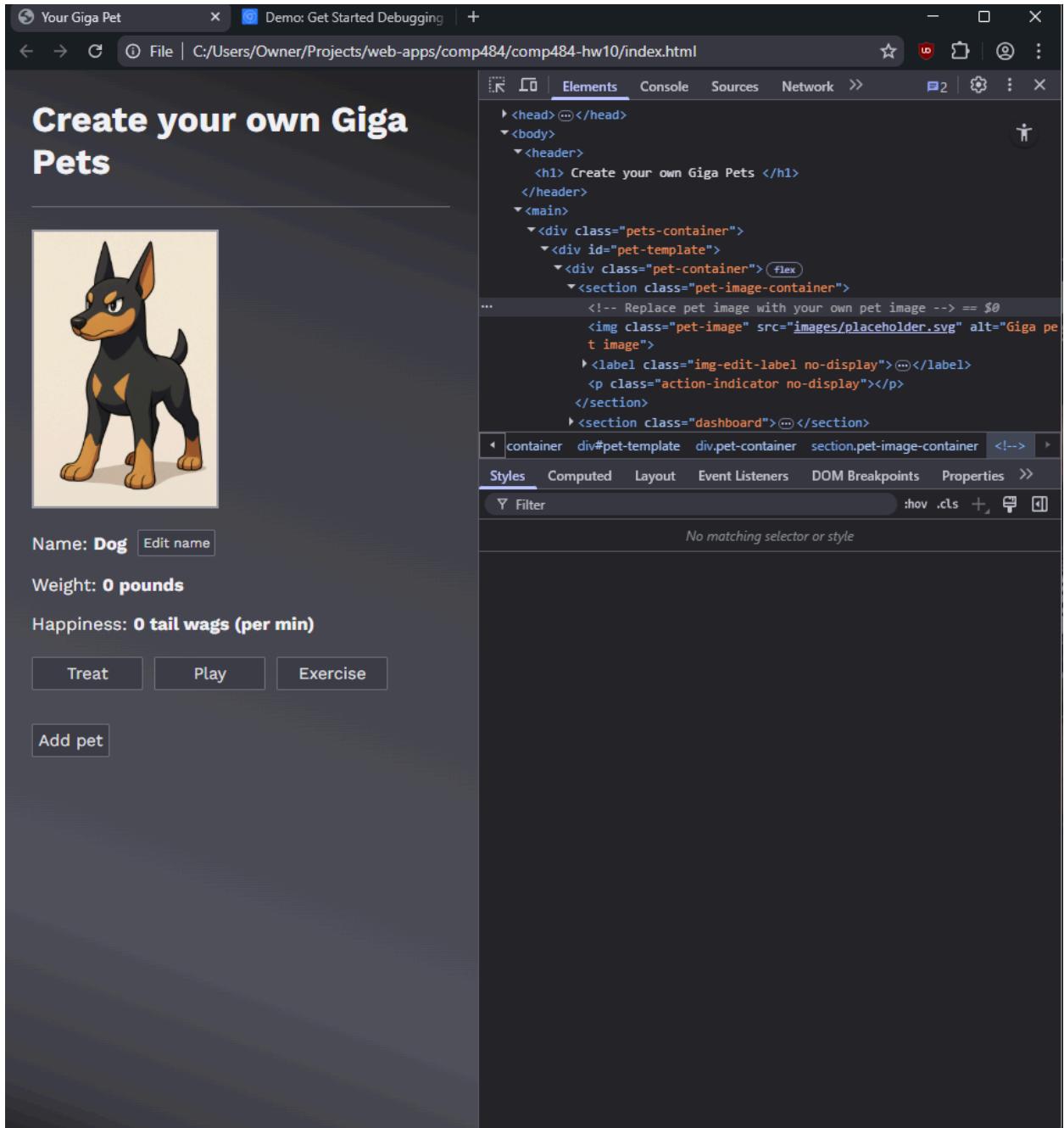
Step 7, Debug JavaScript

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.



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.

Screenshot of a web browser showing the "Create your own Giga Pets" application. The page displays a pet container with a placeholder image and a second pet container below it. The developer tools' Elements panel is open, highlighting the first pet's image element. A context menu is visible, with the "Scroll into view" option selected.

The DOM structure shown in the Elements panel:

```

<section class="dashboard">
  </section>
  </div>
  </div>
  <div class="pet-container">
    </div> (flex)
  <div class="pet-container">
    </div> (flex)
  <div class="pet-container"> (flex)
    <section class="pet-image-container">
      <!-- Replace pet image with your own pet image -->
      ...
       == $0
    </section>
  </div>
  <div>
    <button id="add-pet-button">
  
```

The context menu options include:

- 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
- Store as global variable

The CSS styles for the highlighted element are:

```

element.style {
}
.pet-image {
  height: 250px;
  border: 2px solid var(--c-zinc-400);
}
* {
  box-sizing: border-box;
}
img {
  overflow-clip-margin: content;
  overflow: clip;
}

```

Inherited styles from the root node:

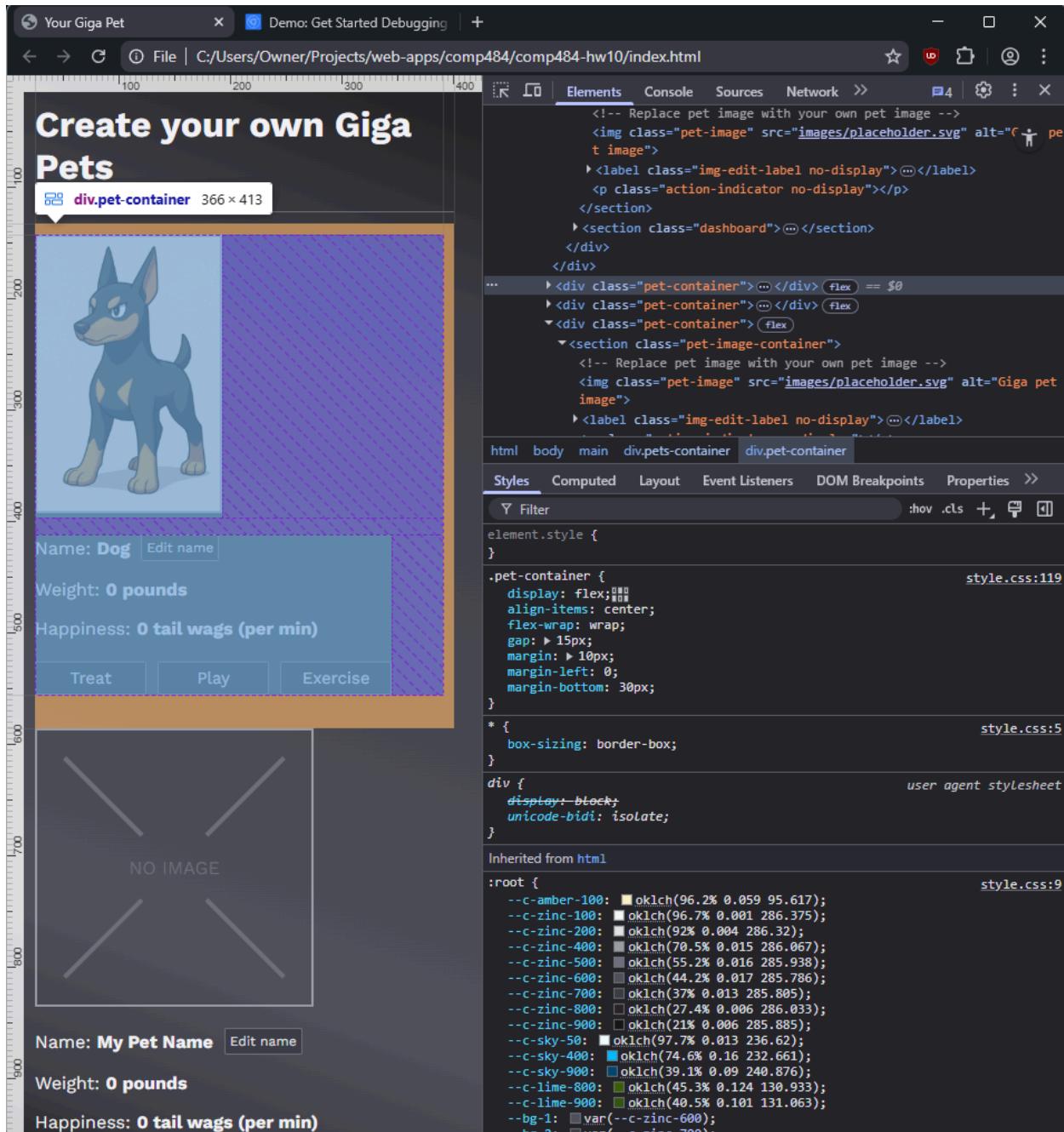
```

:root {
  --c-amber-100: oklch(96.2% 0.000 0.000);
  --c-zinc-100: oklch(96.7% 0.000 0.000);
  --c-zinc-200: oklch(92% 0.000 0.000);
  --c-zinc-400: oklch(70.5% 0.000 0.000);
  --c-zinc-500: oklch(55.2% 0.000 0.000);
  --c-zinc-600: oklch(44.2% 0.000 0.000);
  --c-zinc-700: oklch(37% 0.013 285.805);
  --c-zinc-800: oklch(27.4% 0.000 286.033);
  --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);
}

```

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.

Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



Name: **Dog** [Edit name](#)

Weight: **0 pounds**

Happiness: **0 tail wags (per min)**

[Treat](#) [Play](#) [Exercise](#)



Name: **My Pet Name** [Edit name](#)

Weight: **0 pounds**

Happiness: **0 tail wags (per min)**

Elements Console Sources Network

```

<div>
  "Happiness: "
    <strong>
      ...
        <span class="happiness"></span> == $0
        " tail wags (per min)"
      </strong>
    </div>
  </div>
  <div class="button-container">...</div> (flex)
<section>
</div>
<div class="pet-container">...</div> (flex)
<div class="pet-container">...</div> (flex)
<div class="pet-container">...</div> (flex)

```

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter :hover .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);
height: 100%;

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.”

Screenshot of a web browser showing a pet creation application and its developer tools.

The application interface includes:

- Create your own Giga Pets** heading.
- Dog** image.
- Name: Dog** input field.
- Weight: 0 p** text.
- Happiness: 10000 tail wags (per min)** text.
- Treat**, **Play**, and **Exercise** buttons.
- NO IMAGE** placeholder for another pet's image.
- Name: My Pet Name** input field.
- Weight: 0 pounds** text.
- Happiness: 0 tail wags (per min)** text.

The developer tools (Elements tab) show the DOM structure and styles for the happiness element:

```

<strong>
  <span class="happiness">10000</span> == $0
  " tail wags (per min)"
</strong>

```

Styles panel (style.css:5) shows the following rule:

```

strong {
  font-weight: bolder;
}

```

Inherited from `strong` (user agent stylesheet):

```

strong {
  font-weight: bolder;
}

```

Inherited from `html` (style.css:9):

```

: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);
  height: 100px;
}

```

Edit content, Edit the DOM

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

Screenshot of a web browser showing a pet creation application. The main content area displays a dog named "Dog" with a black and tan coat. Below the name is a text input field for editing the name. Underneath the dog's image, there are three buttons: "Treat", "Play", and "Exercise". To the right of the dog's image, there is a text box showing "Weight: 0 pounds" and "Happiness: 10000 tail wags (per min)". A tooltip or callout bubble points from the "Happiness" text to the value "10000". Below this, there is another section for a pet named "My Pet Name" with similar fields for weight and happiness.

The right side of the screen shows the browser's developer tools, specifically the Elements tab. It highlights the "happiness" span element under the "Dashboard" section. The element's style is shown as "background: red". The developer tools also show the full DOM structure and the CSS styles applied to the page, including "style.css:5" and "style.css:9".

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.

S Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



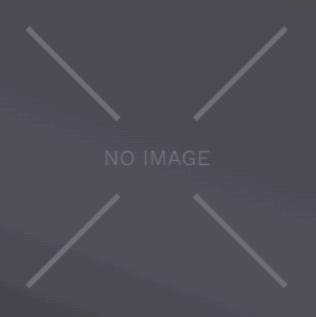
div.pet-name-container 320x24

Name: Dog [Edit name](#)

Weight: 0 pounds

Happiness: 10000 tail wags (per min)

Treat Play Exercise



Name: My Pet Name [Edit name](#)

Weight: 0 pounds

Happiness: 0 tail wags (per min)

Elements Console Sources Network

```

<section class="pet-image-container"></section>
  <div class="info-container">(flex)
    <div class="pet-name-container">...</div>
    <div>...</div>
    <div>
      "Happiness: "
      <strong>
        ...<br/>
        <blockquote class="happiness" style="background: red">10000<br/>
        </blockquote> == $0
        " tail wags (per min)"
      </strong>
    </div>
    <div class="button-container">...</div> (flex)
  </div>

```

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter :hover .cls + □

element.style { background: red; }

* { 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 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.

Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



Name: Dog [Edit name](#)

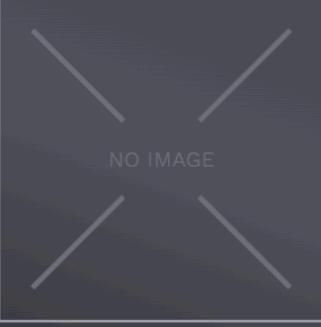
Weight: 0 pounds

blockquote.happiness 240 × 19

Happiness: 10000

tail wags (per min)

Treat Play Exercise

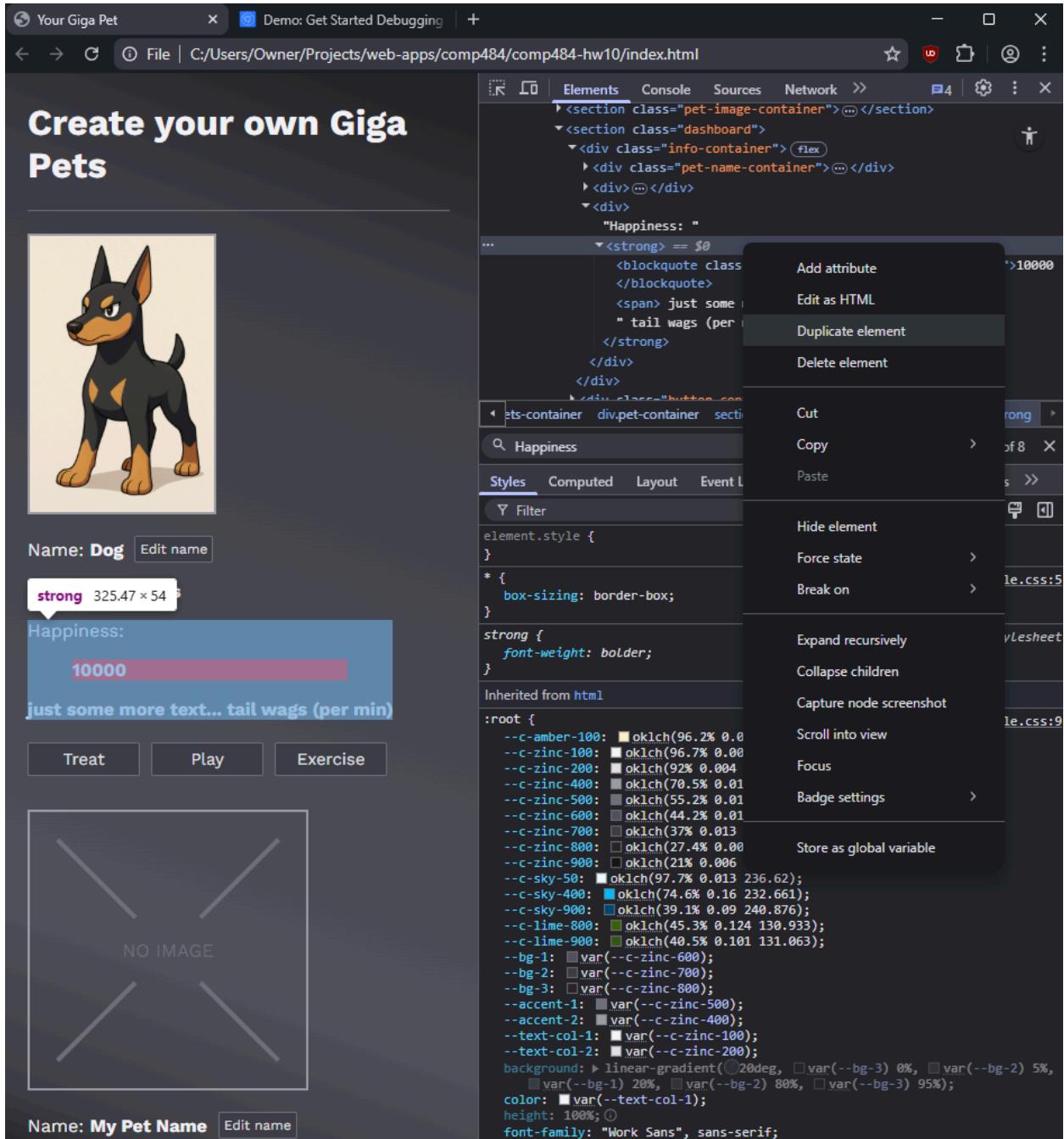


Name: My Pet Name [Edit name](#)

The screenshot shows a web browser window displaying a pet creation application. On the left, there's a section for a dog named 'Dog' with a 'Name' input field. Below it, a 'Weight' section shows '0 pounds'. A 'Happiness' section displays a value of '10000' with a progress bar and the text 'tail wags (per min)'. At the bottom, there are buttons for 'Treat', 'Play', and 'Exercise'. On the right, the browser's developer tools are open, specifically the Elements tab. A blockquote element under the 'happiness' class is selected, containing the text 'just some more text...'. The CSS panel shows the styles for this element, including a red background color. The bottom part of the developer tools shows the DOM tree and various CSS properties for other elements like strong and html.

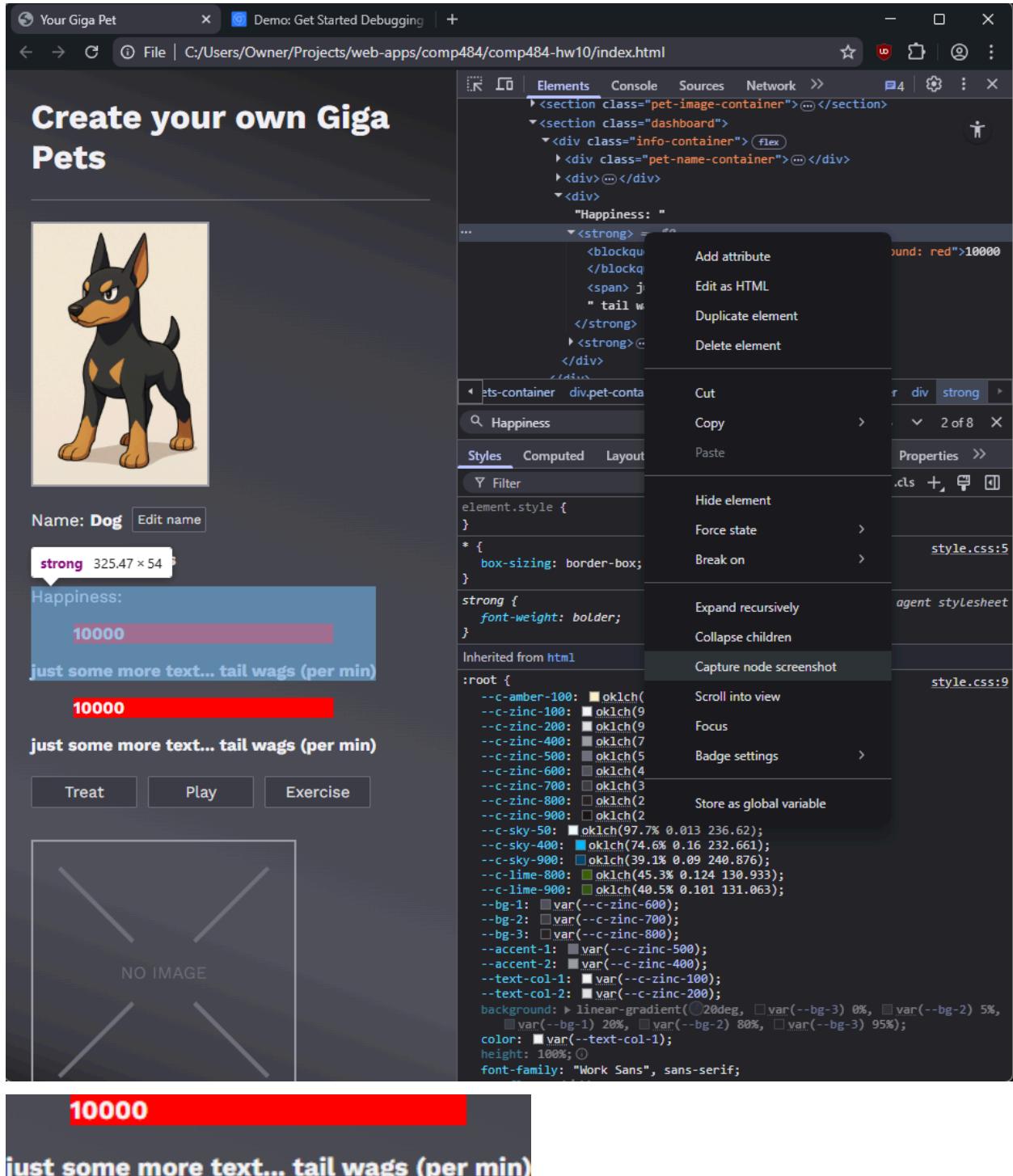
Edit as HTML, Edit the DOM

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.

S Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



Name: **Dog** [Edit name](#)

Weight: **0 pounds**

Happiness:

10000

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

10000

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

[Treat](#) [Play](#) [Exercise](#)

NO IMAGE

Elements Console Sources Network

```

<section class="dashboard">
  <div class="info-container">(flex)
    <div class="pet-name-container">::</div>
    <div>::</div>
    "Happiness: "
    <strong>
      <blockquote class="happiness" style="background: red">10000
        </blockquote> == $0
      <span> just some more text...</span>
      " tail wags (per min)"</strong>
    </strong>
    <strong>::</strong>
  </div>
  <div class="button-container">::</div> (flex)
</section>

```

Styles Computed Layout Event Listeners DOM Breakpoints Properties

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; }

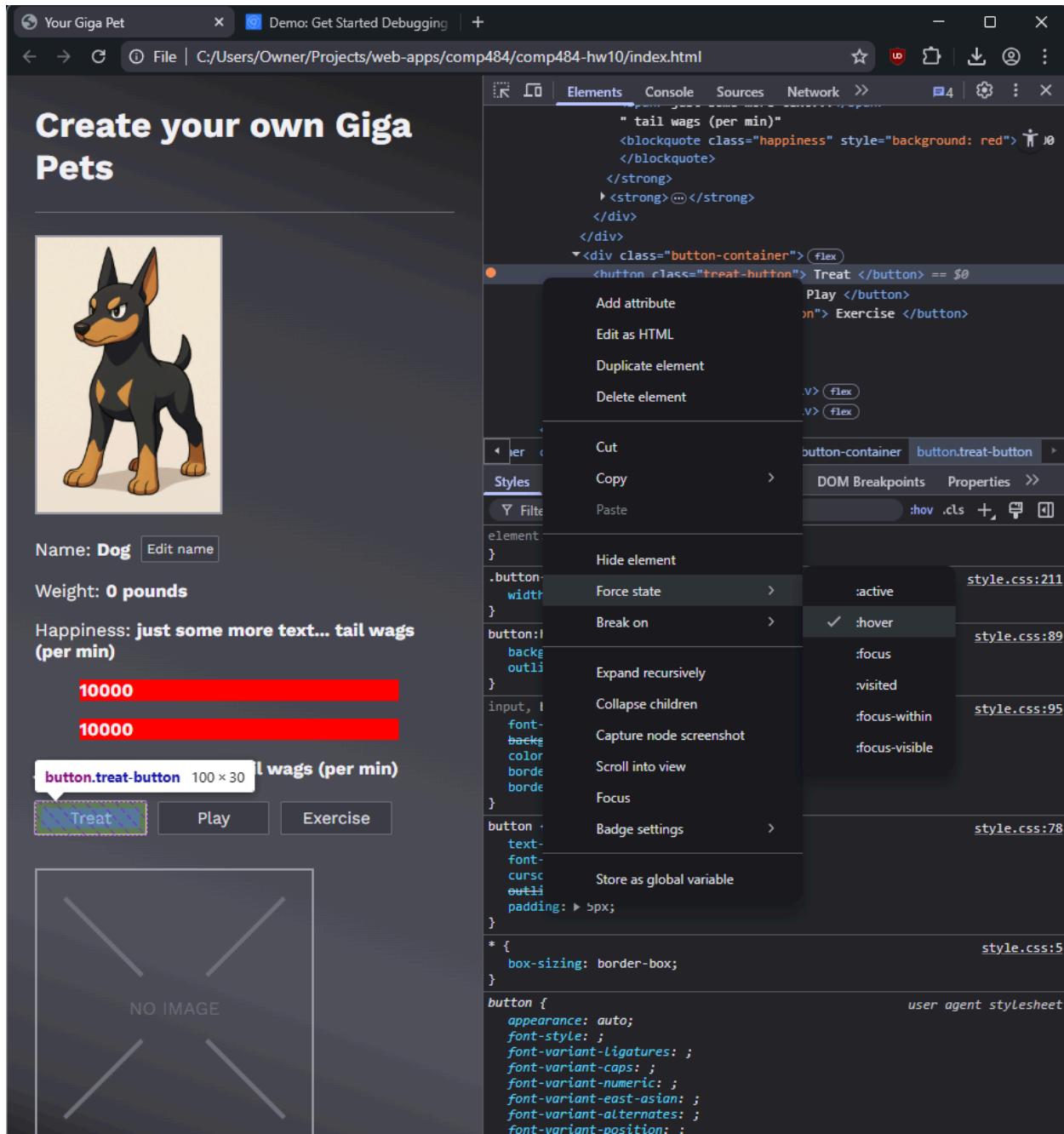
Inherited from html

: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).

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.

S Your Giga Pet X Demo: Get Started Debugging | +

← → C File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



Name: **Dog** [Edit name](#)

Weight: **0 pounds**

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

10000 

10000 

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



tail wags (per min)"
`blockquote class="happiness" style="background: red">
</blockquote>
`
`...
`</div>
`</div>
`<div class="button-container __web-inspector-hide-shortcut__" style="flex" = \$0
`<button class="treat-button"> Treat </button>
`<button class="play-button"> Play </button>
`<button class="exercise-button"> Exercise </button>
`</div>
`</section>
`</div>
`<div class="pet-container">...</div> 
`<div class="pet-container">...</div> 

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter :hover .cls +, □

```
element.style {  
}  
.__web-inspector-hide-shortcut__, __web-inspector-hide-shortcut__ * {  
  visibility: hidden !important;  
}  
.button-container {  
  margin-top: 20px;  
  display: flex;flex-wrap: wrap;  
  gap: 10px;  
}  
* {  
  box-sizing: border-box;  
}  
div {  
  display: block;  
  unicode-bidi: isolate;  
}  
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);
```

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.

Screenshot of a web browser showing the "Create your own Giga Pets" application. The page displays a Doberman Pinscher, a name input field ("Name: Dog"), a weight input field ("Weight: 0 p"), and a happiness progress bar ("Happiness: just some more text... tail wags (per min) 10000"). Below the progress bar are three buttons: "Treat", "Play", and "Exercise". A modal window titled "NO IMAGE" is visible at the bottom left. The browser's developer tools are open, specifically the Elements tab, where a context menu is displayed over the first span element under the "Happiness" heading. The menu includes options like "Edit text", "Edit as HTML", "Duplicate element", and "Delete element". The "Delete element" option is highlighted.

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.

S Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



section.dashboard 325.47 x 249

Name: Dog [Edit name](#)

Weight: 0 pounds

Happiness: tail wags (per min)

10000

10000

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

Treat Play Exercise

NO IMAGE

Elements Console Sources Network

```


<label class="img-edit-label no-display"></label>
<p class="action-indicator no-display" style="opacity: 1;">That's
tiring!</p>
</section>
<section class="dashboard" == $0
  > <div class="info-container"><flex>
    > <div class="pet-name-container"><..></div>
    > <div><..></div>
  </div>
  > <div>
    > "Happiness: "
    > <strong>
      > " tail wags (per min)"
      > <blockquote class="happiness" style="background: red">10000
        </blockquote>
    > </strong>
    > <strong><..></strong>
  </div>
  > <div class="button-container"><flex>
    > <button class="treat-button"> Treat </button>
    > <button class="play-button"> Play </button>
    > <button class="exercise-button"> Exercise </button>
  </div>
</div>
<div class="pet-container"><..></div> (flex)
<div class="pet-container"><..></div> (flex)
</div>
<button id="add-pet-button">Add pet</button>
</main>
<!-- Your web-app is https, so your scripts need to be too -->
<script src="https://code.jquery.com/jquery-2.1.1.min.js" integrity="sha256-gvQAFzTH6trSrAWoH1iPo9Xc96QxSZ3feW6kem+000--" crossorigin="anonymous">
</script>
</body>
</html>

```

html body main div.pets-container div.pet-container section.dashboard

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter

Console

\$0

< <section class="dashboard"><..></section>

|

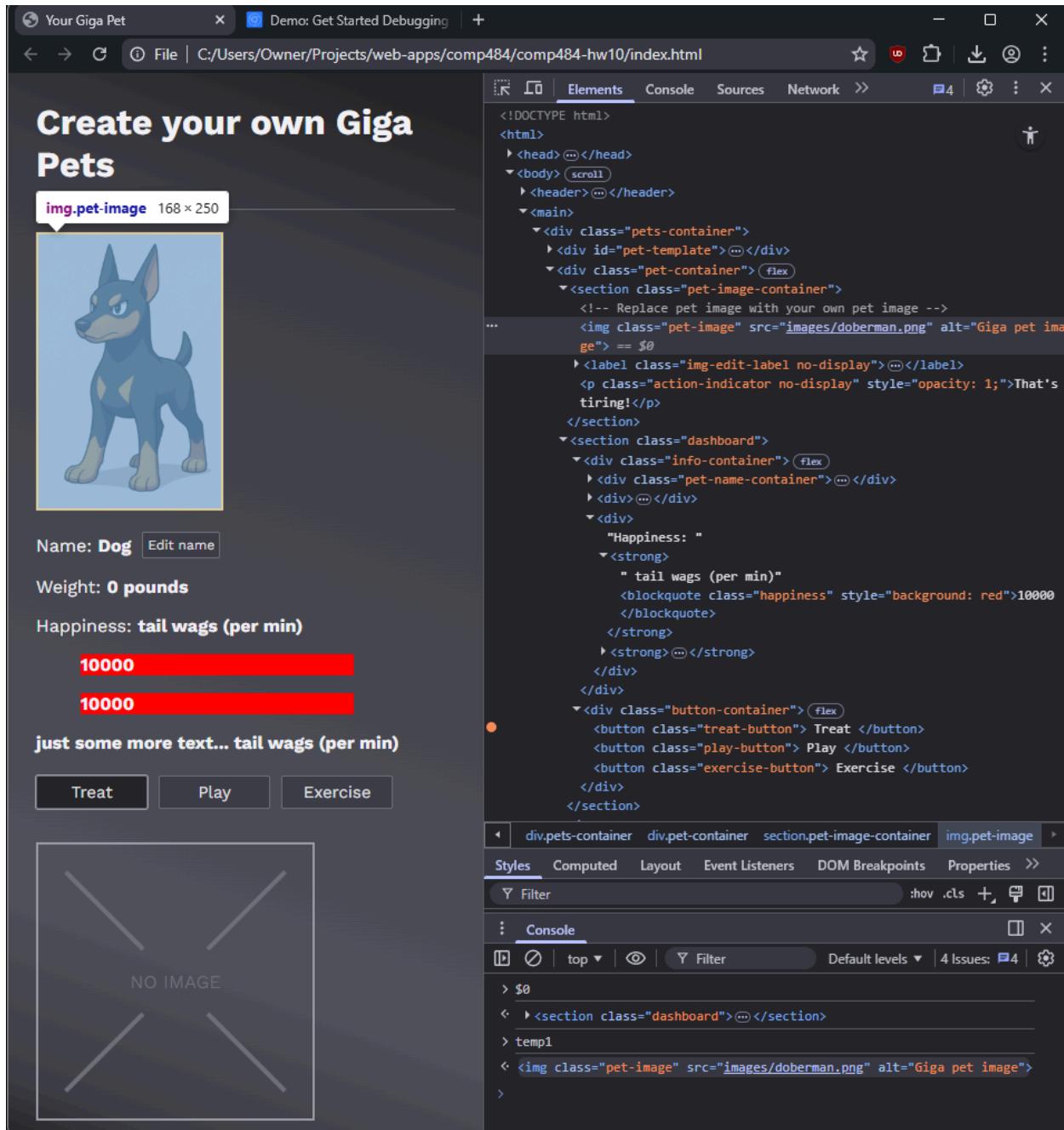
Reference currently selected node with \$0, Access nodes in the Console

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

Your Giga Pet Demo: Get Started Debugging

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets



Name: **Dog** [Edit name](#)

Weight: **0 pounds**

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

10000

10000

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

Treat Play Exercise

NO IMAGE

Elements Console Sources Network

```
<!DOCTYPE html>
<html>
  <head> ...
  <body> (scroll)
    <header> ...
    <main>
      <div class="pets-container">
        <div id="pet-template"> ...
        <div class="pet-container" style="flex">
          <section class="pet-image-container">
            <!-- Replace pet image with your own pet image -->
            
            <label class="img-edit-label no-display"> ...
            <p class="action-indicator no-display" style="opacity: 1;">That's tiring!</p>
          </section>
          <section class="dashboard">
            <div class="info-container" style="flex">
              <div class="pet-name-container"> ...
              <div>
                "Happiness: "
                <strong>
                  " tail wags (per min)"<br/>
                  <blockquote class="happiness" style="background: red">10000</blockquote>
                </strong>
                <strong> ...
                </strong>
              </div>
            </div>
            <div class="button-container" style="flex">
              <button class="treat-button"> Treat </button>
              <button class="play-button"> Play </button>
              <button class="exercise-button"> Exercise </button>
            </div>
          </section>
        </div>
      </div>
    </main>
  </body>
</html>
```

div.pets-container div.pet-container section.pet-image-container img.pet-image

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter :hover .cls + ↻

Console

Default levels 4 Issues: P4

```
> $0
< <section class="dashboard"> ...
> temp1
< 
```

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.

S Your Giga Pet Demo: Get Started Debugging | +

File | C:/Users/Owner/Projects/web-apps/comp484/comp484-hw10/index.html

Create your own Giga Pets

div.pet-name-container 325.47 × 24

Name: Dog Edit name

Weight: 0 pounds

Happiness: tail wags (per min)

10000

10000

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

Treat Play Exercise

NO IMAGE

Elements Console Sources Network

```
<div class="pets-container">
  <div id="pet-template">...</div>
  <div class="pet-container">(flex)
    <section class="pet-image-container">
      <!-- Replace pet image with your own pet image -->
      
    <label class="img-edit-label no-display">...</label>
    <p class="action-indicator no-display" style="opacity: 1;">That's tiring!</p>
  </section>
  <section class="dashboard">
    <div class="info-container">(flex)
      <div class="pet-name-container"> == $0
        <div class="pet-name-and-button">...</div>(flex)
        <form class="name-form no-display">...</form>
      </div>
      <div>
        "Happiness: "
        <strong>
          " tail wags (per min)"
          <blockquote class="happiness" style="background: red">10000
            </blockquote>
        </strong>
        <strong>...</strong>
      </div>
      <div class="button-container">(flex)
        <button class="treat-button"> Treat </button>
        <button class="play-button"> Play </button>
        <button class="exercise-button"> Exercise </button>
      </div>
    </section>
  </div>
  <div class="pet-container">...</div>(flex)
  <div class="pet-container">...</div>(flex)

```

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Y Filter

Console

allow pasting below and press Enter to allow pasting.

allow pasting

> document.querySelector("body > main > div > div:nth-child(2) > section.dashboard > div.info-container > div.pet-name-container")

<

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.