

Matthieu Resurreccion

Prof. Marin

COMP484

3 December 2025

## Homework 10

### Breakpoints and Debugging

# This is your Giga Pet

---



Name: **Matt**  
Weight: **7 pounds**  
Happiness: **5 tail wags (per min)**  
Health: **3 HP**

Treat

Play

Exercise


Sleep

You gave Matt a treat!

Fig. 1 Project 2 Treat Click Event

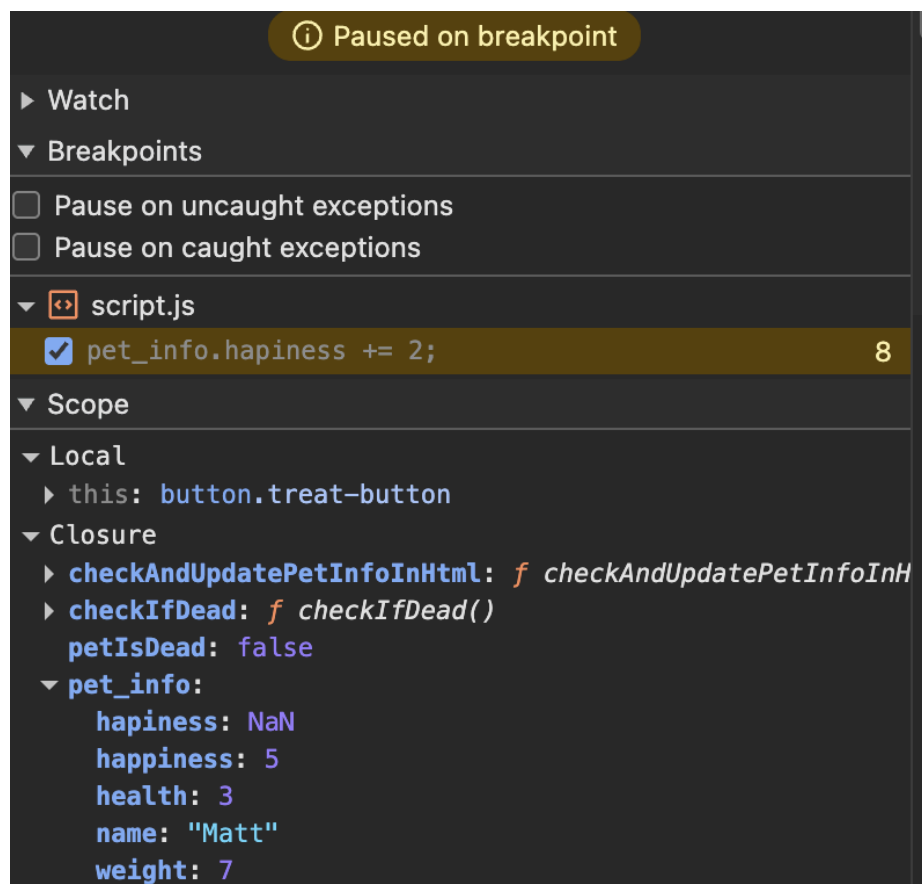
As shown above, when the user clicks the ‘Treat’ button, it displays a message and should increase ‘Weight’, decrease ‘Health’, and increase ‘Happiness’. However, here, ‘Happiness’

remains at its starting value of 5. We can use Chrome Dev Tools by setting a breakpoint at where this state is supposed to update.



```
1  $(function () {  
2      var pet_info = { name: "Matt", weight: 5, happiness: 5, health:  
3          let petIsDead = false;  
4  
5      $(".treat-button, .play-button, .exercise-button, .sleep-button"  
6  
7      $(".treat-button").click(function () {  
8          pet_info.hapiness += 2;  
9          pet_info.weight += 1;  
10         pet_info.health -= 1;  
11  
12         showNotification(`You gave ${pet_info.name} a treat!`);  
13         spawnEffect("🐾");  
14         checkAndUpdatePetInfoInHtml();  
15     });  
16 }
```

Fig. 2 Project 2 Breakpoint Set



Paused on breakpoint

Watch

Breakpoints

- ☐ Pause on uncaught exceptions
- ☐ Pause on caught exceptions

script.js

- ☒ pet\_info.hapiness += 2; 8

Scope

- Local
  - this: button.treat-button
- Closure
  - checkAndUpdatePetInfoInHtml: f checkAndUpdatePetInfoInH
  - checkIfDead: f checkIfDead()
  - petIsDead: false
  - pet\_info:
    - hapiness: NaN
    - happiness: 5
    - health: 3
    - name: "Matt"
    - weight: 7

Fig. 3 Project 2 Breakpoint Scope

Once we inspect the scope of the breakpoint, we can view the values of each object, most importantly, 'pet\_info' which are the variables that we are concerned about. Clearly, there is an additional variable 'hapiness', showing that there is a typo in the script that goes unnoticed.

## Live DOM Editing

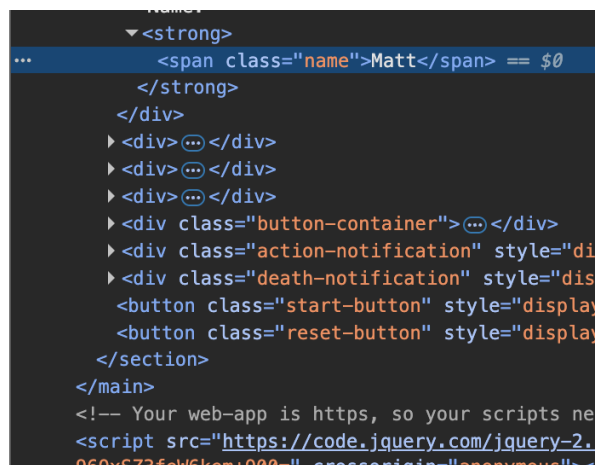
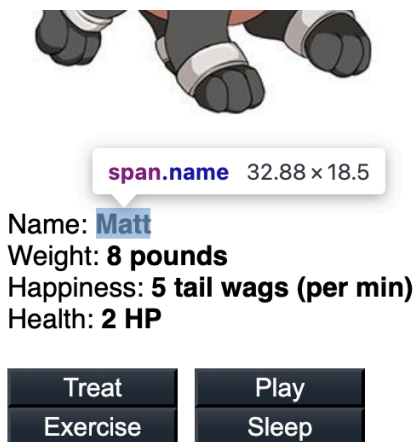


Fig. 4 Project 2 Name Span

See that we can access the DOM element that displays the name of our pet.

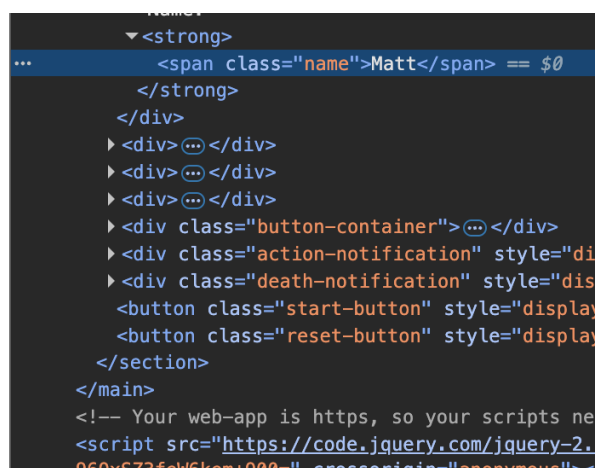
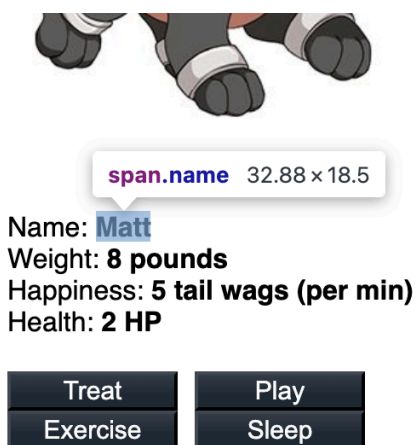


Fig. 5 Project 2 Live Editing Name Span

Through Chrome Dev Tools, we are able to edit the DOM and see the updates take place live.

## Animations Spawning



Fig. 6 Project 2 ‘zzz’ Animation

Chrome Dev Tools also allow us to inspect as animations spawn in the DOM.

```
et-image-container">
ct" style="top: 127.027px; opacity: 0.837956;">Z</div>
  image with your own pet image -->
image" src="https://cdn.glitch.com/3aa98e05-3216-497c-a251-210ae4713a83
">
t-container"></div>

ashboard">

'name">Matt</span> == $0

ton-container">⋮</div>
ion-notification" style="display: block; opacity: 0.951967;">⋮</div>
h-notification" style="display: none;">⋮</div>
```

Fig. 7 Project 2 Animation Spawning

See how the ‘top’, ‘opacity’, and ‘display’ styles change in regard to the animation.

## Adding CSS Classes

```
.pet-image {  
    height: 250px;  
}
```

style.css:48

Fig. 8 Project 2 '.pet-image' Original Style

```
.pet-image {  
    height: 250px;  
    border: 5px solid red;  
    transform: rotate(5deg);  
}
```

\*style.css:48

Fig. 9 Project 2 '.pet-image' New Style



Fig. 10 Project 2 '.pet-image' New Style Result