

Joshua Gonzalez-Hahn  
Comp 484  
Homework 10

# Debug JavaScript

So, for implementing the Debug JavaScript portion, I went ahead and added two input fields that will have you input your pet's weight and happiness values, and then you click a button to add them together to get your pet's power level. However, the result will be a string and show the two values merged instead of added.

## 1. Reproduce the bug

So, for this part, the user will enter two values based on stats from their pet to get a power level calculation.



The result will be wrong. For this example, the result should be 70.

## 2. Familiarity with Sources UI for my project

Here we are, the three important sections: the page tab, code editor, and the debugger section.

This is your Giga Pet!

Name: Pakkun  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator

script.js X

```
//This is where I create the function to add the pet's weight and happiness
$("#calc-button").click(function () {
  var num1 = $("#input1").val();
  var num2 = $("#input2").val();

  //add num1 and num2
  var sum_result = num1 + num2;

  //show the result
  $("#calc-result").text(sum_result);
});

// Add a variable "pet_info" equal to a object with the name and weight/happiness
var pet_info = {
  name: "Pakkun",
  weight: 20,
  happiness: 50
};
```

Breakpoints

- Pause on uncaught exceptions
- Pause on caught exceptions

Call Stack

Not paused

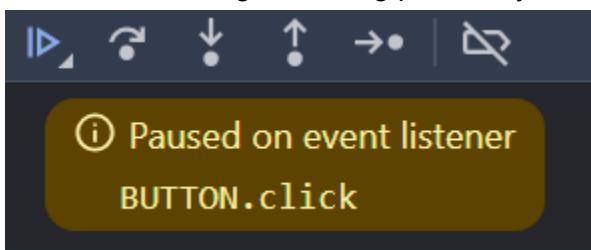
- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints

Scope Watch

Console AI assistance

### 3. Pause the code with breakpoints

We are first looking at the bug potentially being at the time we click the button.



I'm able to scroll to the next function call and such using these buttons.

This approach wasn't too helpful as it wouldn't highlight in the "script.js" file.

However, placing a breakpoint did!

## 5. Set a line-of-code breakpoint

The screenshot shows a browser developer tools interface with the "Sources" tab selected. The code editor displays a file named "script.js" with the following content:

```
// Add a variable "pet_info" equal to a object with the name
var pet_info = { name: "Pakkun", weight: 20, happiness: 50 }

function clickedPowerLevelCalculator() {
    var num1 = $("#input1").val();
    var num2 = $("#input2").val();

    //add num1 and num2
    var sum_result = num1 + num2;

    //show the result
    $("#calc-result").text(sum_result);
}

function clickedTreatButton() {
    // Increase pet happiness
    pet_info.happiness += 1;
}
```

A blue highlight on line 88 indicates a breakpoint has been set at `$("#calc-result").text(sum_result);`. The right-hand sidebar shows the "Event Listener Breakpoints" section, where the "click" event on the element `#calc-result` is highlighted.

## 6. Check variable values

1. We can check the “scope” to check variable values.

Paused in debugger

Name: Pakkun  
Weight: 22 pounds  
Happiness: 54 tail wags (per min)  
Comment from your pet: Thank you for the treat!

Your Pet's powerlevel calculator

Pet's Power level is weight + happiness

Your Pet's Weight 20 Your Pet's Happiness 57  
Calculate your Pet's Power Level

Result is: 2057

Treat Play Exercise Special Weight Loss Treat

Special Food Menu Sleep

```

Page >> : script.js X jquery-2.2.1.min.js
    ↴
    ▾ top
    ▾ 127...
    ▾ i...
    ▾ i...
    ▾ S...
    ▾ S...
    ▾ cod...
    ▾ j...
    ▾ fon...
    ▾ fon...
    ↴
    74 // Add a variable "pet_info" equal to a object
    75 var pet_info = { name: "Pakkun", weight: 20, h
    76
    77
    78
    79 function clickedPowerLevelCalculator() {
    80
    81     var num1 = $("#input1").val(); num1 = "20"
    82     var num2 = $("#input2").val(); num2 = "57"
    83
    84     //add num1 and num2
    85     var sum_result = num1 + num2; sum_result =
    86
    87     //show the result
    88     $("#calc-result").Dtext(sum_result);
    89 }
    90
    91
    92
    93 function clickedTreatButton() {
    94     // Increase pet happiness
    95     pet_info.happiness += 1;
    96 }
```

Line 88, Column 3 Coverage: n/a

Console AI assistance

Default levels ▾ No Issues 4 hidden

We have num1: "20"

num2: "57"

sum\_result: "2057"

## 2. We can use watch expressions.

Paused on breakpoint

Watch

typeof sum\_result

Breakpoints

Pause on uncaught exceptions  
Pause on caught exceptions

script.js

\$("#calc-result").text(sum\_result);

Scope

Local

this: button#calc-button  
num1: "20"  
num2: "57"  
sum\_result: "2057"

Global

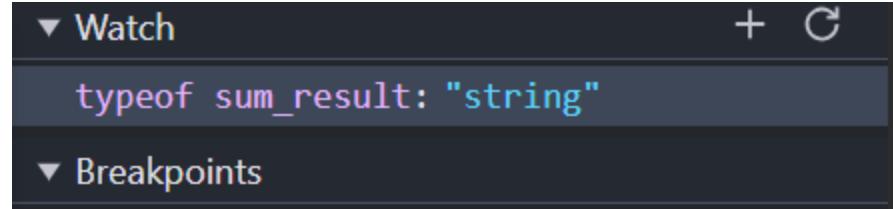
Call Stack

clickedPowerLevelCalculator script.js:88  
dispatch jquery-2.2.1.min.js:3  
(anonymous) jquery-2.2.1.min.js:3

Console AI assistance

Default levels ▾ No Issues 4 hidden

I made a new watch expression and named it sum\_result.



After hitting enter, we see sum\_result is of type String.

3. We can also use the console to evaluate JavaScript statements.

The screenshot shows the Chrome DevTools with the debugger paused at line 78 of the `script.js` file. The code at line 78 is `\$( "#calc-result" ).text( sum\_result );`. The `sum\_result` variable is highlighted in yellow and shows its value as "2057". The local scope shows `num1` as "20", `num2` as "57", and `sum\_result` as "2057". The global scope shows the function `clickedPowerLevelCalculator` and variables `dispatch`, `r.handle`, and `XHR/fetch Breakpoints`.

From this screenshot, we know the issue is that our numbers being added are actually strings. So I tested `parseInt(num1) + parseInt(num2)`, and the result is correct.

## 7. Apply the fix

I now added the new code with `parseInt(num1) + parseInt(num2)` (line 76)

Screenshot of the Chrome DevTools Sources tab showing the script.js file. The code contains two functions: clickedPowerLevelCalculator and clickedTreatButton. The clickedPowerLevelCalculator function adds the values of two input fields and displays the result. The clickedTreatButton function increases the pet's happiness by 4. The DevTools sidebar shows various breakpoints and scopes.

```

script.js
jquery-2.2.1.min.js

66
67
68 //This is where I create the function to add the pet's v
69 function clickedPowerLevelCalculator() {
70
71     var num1 = $("#input1").val();
72     var num2 = $("#input2").val();
73
74     //add num1 and num2
75     //var sum_result = num1 + num2;
76     var sum_result = parseInt(num1) + parseInt(num2)
77     //show the result
78     $("#calc-result").text(sum_result);
79 }
80
81
82
83 function clickedTreatButton() {
84     // Increase pet happiness
85     pet_info.happiness += 4;
86     // Increase net weight
87 }

Line 77, Column 20
Coverage: n/a

```

Console tab:

```

> parseInt(num1) + parseInt(num2)
< 77
>

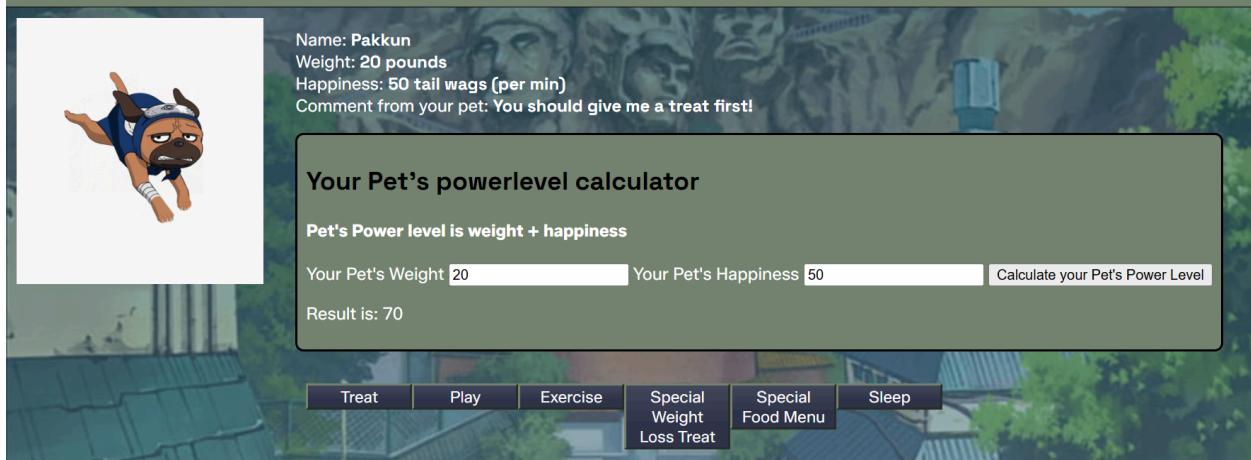
```

It now gives the corrected output.

Screenshot of the browser showing the results of the calculation. The page displays a pet named Pakkun with a weight of 20 pounds and a happiness of 50 tail wags per minute. A message encourages giving a treat. Below the message is a calculator interface titled "Your Pet's powerlevel calculator". It shows the formula "Pet's Power level is weight + happiness". The user has entered a weight of 20 and a happiness of 57, resulting in a power level of 77. Navigation buttons for Treat, Play, Exercise, Special Weight, and Special Food Menu are visible at the bottom.

This is a screenshot of the bug fixed and with the true values that should be inputted.

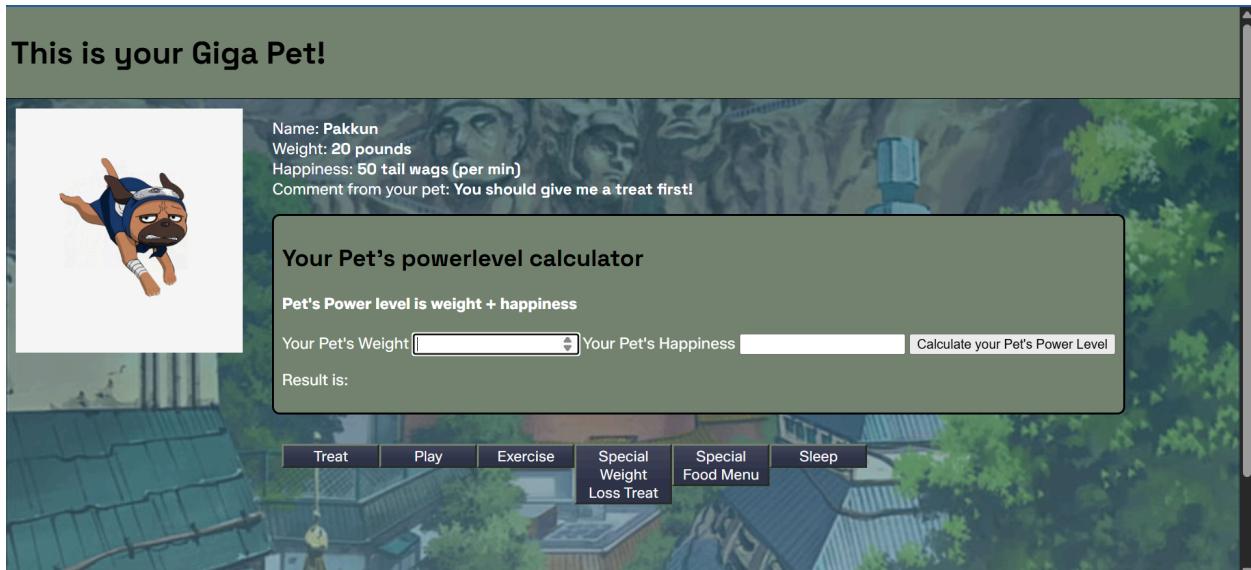
## This is your Giga Pet!



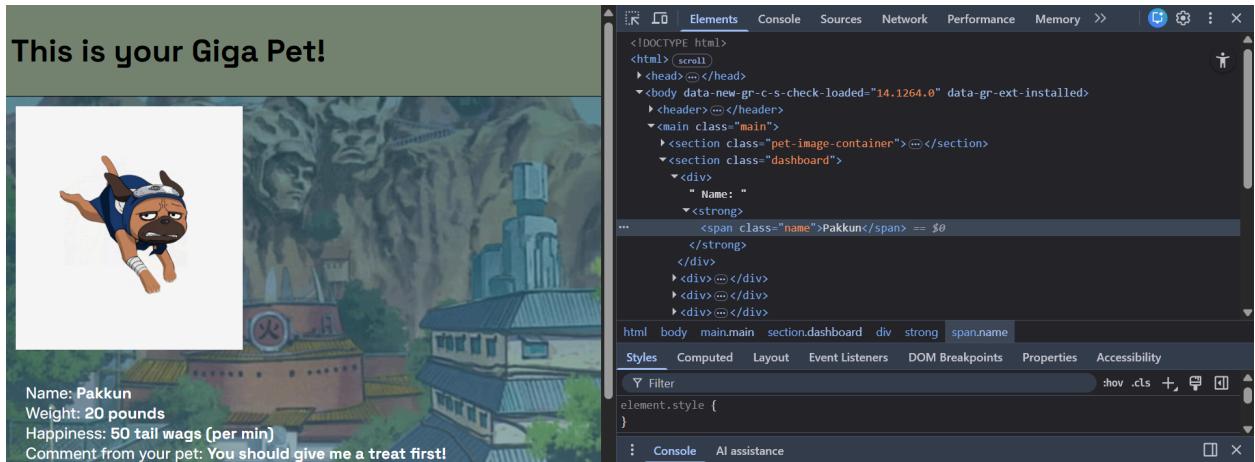
# Viewing and changing the DOM

The first section is **View DOM nodes**

The first is to **inspect a node**. For my project, we have this page:



Highlighting or hovering over the name "Pakkun", then right-click and select inspect, will open Devtools and highlight <span class="name">Pakkun</span>.



Using the select an element button, I can inspect any element.

Next is to **navigate the DOM Tree with a keyboard**. When I inspect the group of buttons(sleep button), I see this:

The screenshot shows a web browser window with the URL `127.0.0.1:5500/index.html`. On the left is the application interface, which includes a pet profile for 'Pakkun' (Name: Pakkun, Weight: 20 pounds, Happiness: 50 tail wags (per min)) and a 'Your Pet's powerlevel calculator' section. On the right is the browser's developer tools, specifically the Elements tab. The DOM tree is expanded to show a `<div class="button-container">` element containing several buttons: `<button class="treat-button">Treat</button>`, `<button class="play-button">Play</button>`, `<button class="exercise-button">Exercise</button>`, `<button class="special-treat-button"> Special Weight Loss Treat </button>`, and `<button class="sleep-button">Sleep</button>`. The `sleep-button` is highlighted in yellow. The developer tools also show the CSS styles for the `button.sleep-button` class.

Clicking up 5 times gets me to the div the buttons are in. Clicking left once collapses it. And left again took me to the parent that div node, which is the section element with class= "dashboard".

This screenshot shows the same browser setup as the previous one, but the DOM tree has been navigated further up. The `<section class="dashboard">` node is now expanded, showing its contents: two `<div>` elements, each containing a `<strong>` tag with the text 'Name:' and 'Weight:'. This demonstrates how navigating through the DOM tree with keyboard arrows can reveal the overall structure of the page's HTML.

Going back to the div and clicking the right one time reopens the group of buttons.

This screenshot shows the developer tools with the `<div class="button-container">` node expanded. The list of buttons—`Treat`, `Play`, `Exercise`, `Special Weight Loss Treat`, and `Sleep`—is clearly visible, demonstrating that the user can collapse and expand parts of the DOM tree to inspect specific sections of the page's structure.

Next is to scroll into view.

Selecting to inspect the title.

This is your Giga Pet!

Name: Pakkun  
Weight: 20 pounds  
Happiness: 50 tall wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator

```
<!DOCTYPE html>
<html> (scroll)
  > <head> ... </head>
  > <body data-new-gr-c-s-check-loaded="14.1264.0" data-gr-ext-installed>
    > <header>
      ... <h1>this is your Giga Pet!</h1> == $0
    </header>
    > <main class="main">
      > <section class="pet-image-container"> @</section>
      > <section class="dashboard">
        > <div>
          " Name: "
          > <strong>
            <span class="name">Pakkun</span>
          </strong>
        </div>
      </div>
    </main>
  </body>
```

Now, if I were to scroll down as such:

Name: Pakkun  
Weight: 20 pounds  
Happiness: 50 tall wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator

Pet's Power level is weight + happiness

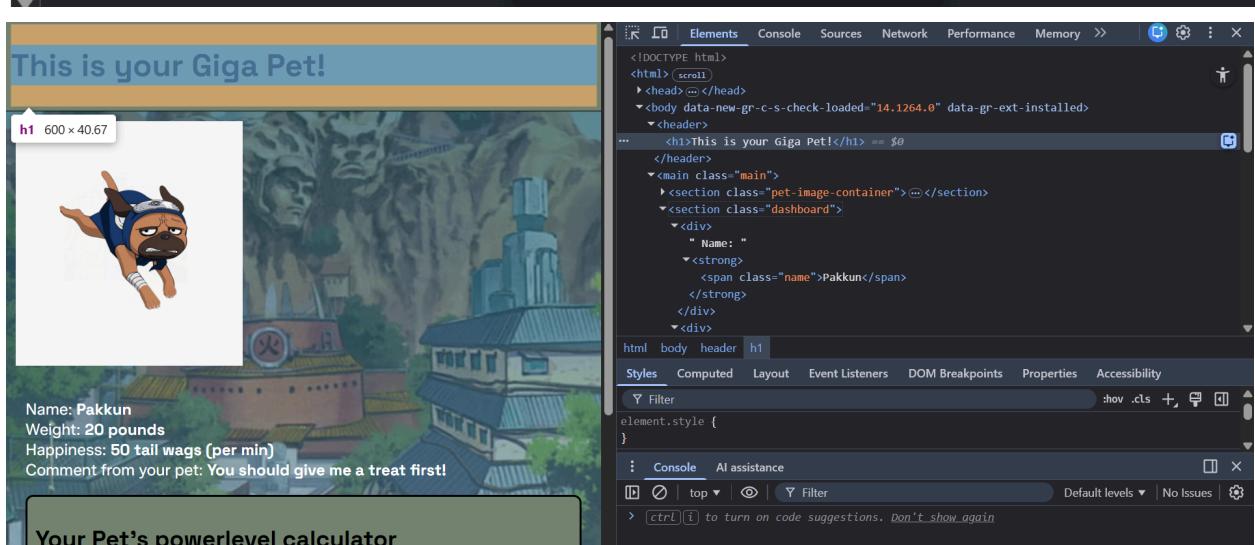
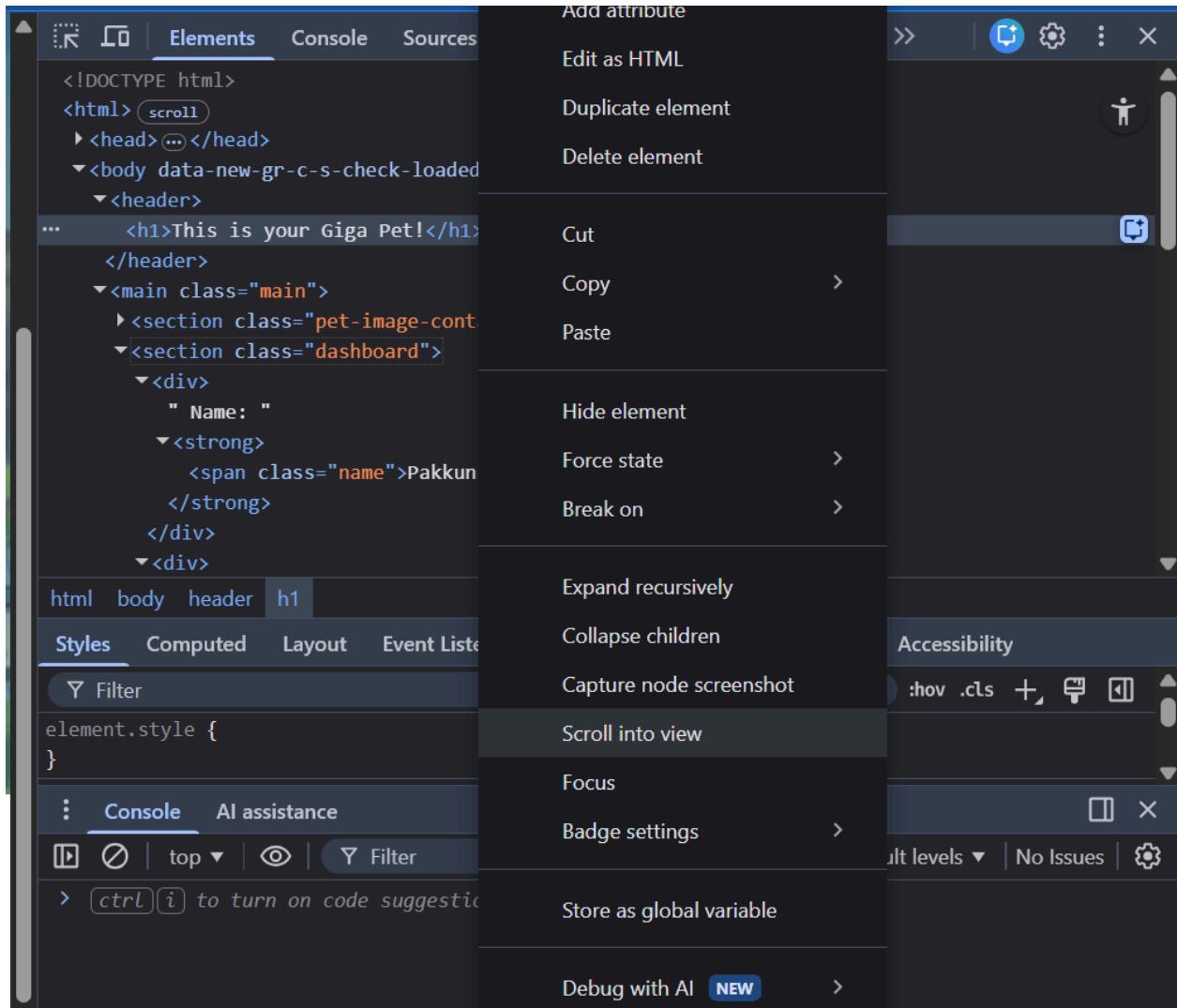
Your Pet's Weight  Your Pet's Happiness

Result is:

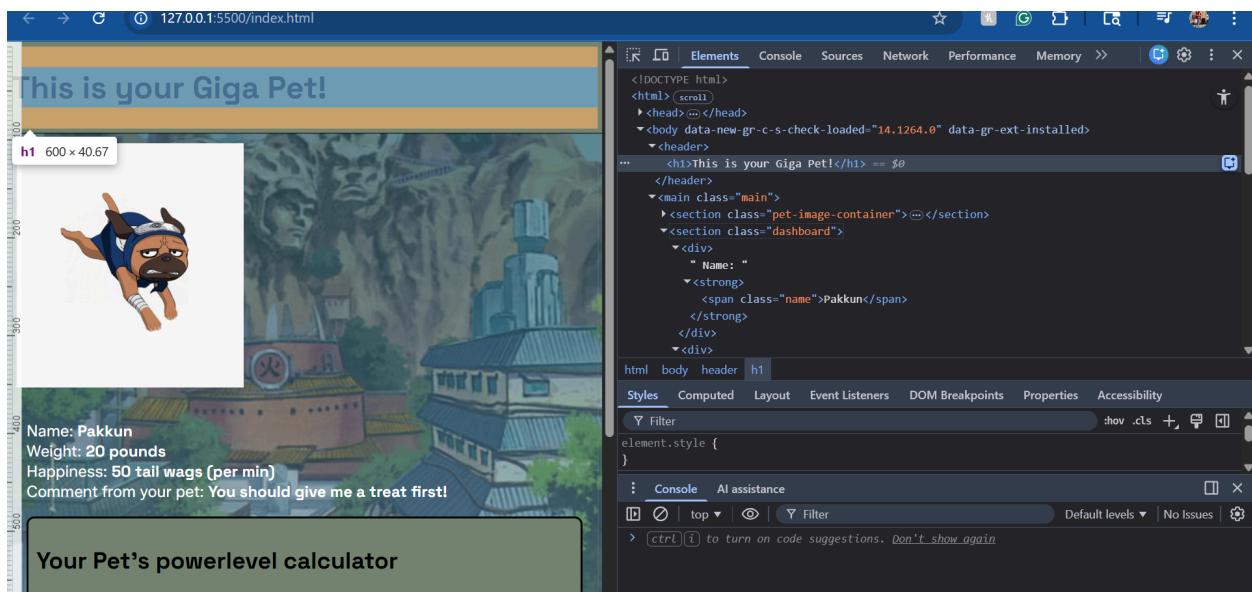
Treat Play Exercise Special Weight Loss Treat Special Food Menu Sleep

```
<!DOCTYPE html>
<html> (scroll)
  > <head> ... </head>
  > <body data-new-gr-c-s-check-loaded="14.1264.0" data-gr-ext-installed>
    > <header>
      ... <h1>this is your Giga Pet!</h1> == $0
    </header>
    > <main class="main">
      > <section class="pet-image-container"> @</section>
      > <section class="dashboard">
        > <div>
          " Name: "
          > <strong>
            <span class="name">Pakkun</span>
          </strong>
        </div>
      </div>
    </main>
  </body>
```

Now if I were interested in the h1 title again, I can right-click on the h1(title) node. And clicked Scroll into view to go back to that location.

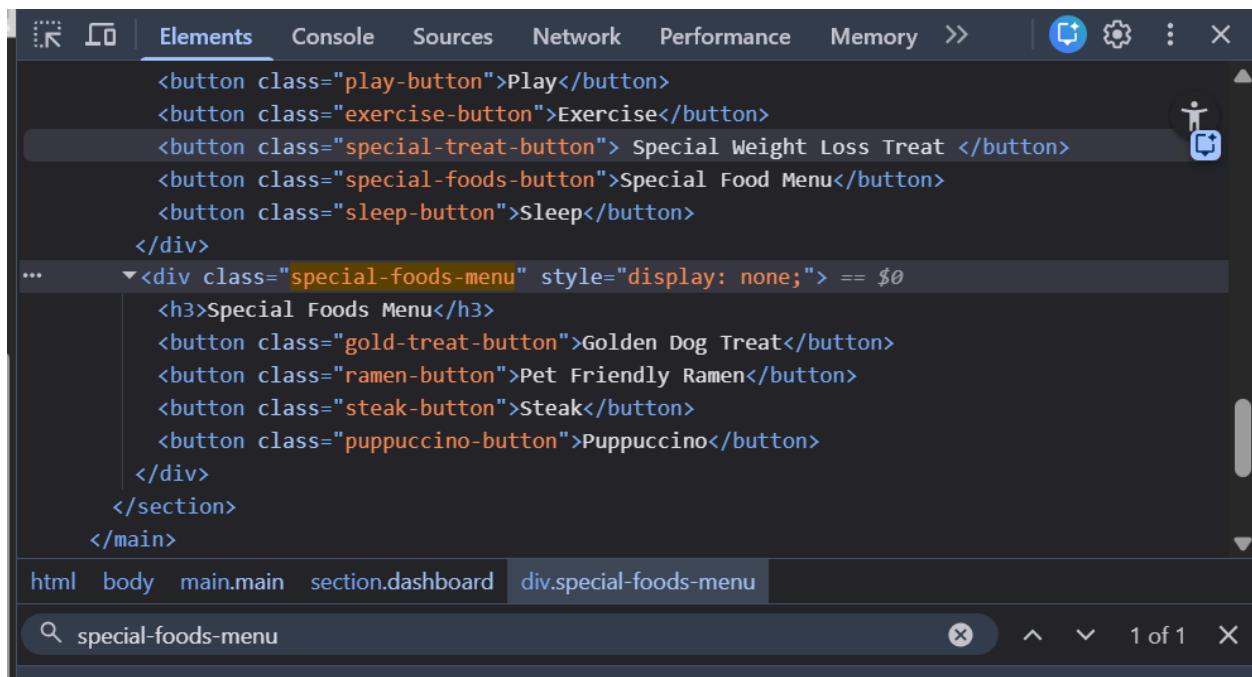


Next is **show rulers**.



Next is **Search for nodes**.

Using **ctrl+f** on the Elements section, I can find where I have the special-foods-menu.



The next section is **Edit the DOM**.

First is **Edit content**.

A screenshot of a web application interface. On the left, there is a placeholder image for a pet's photo. To its right, a large image of a village with a prominent rock formation featuring a face and a red circular emblem. Below these images, the following information is displayed:

Name: **Pakkun**  
Weight: **20 pounds**  
Happiness: **50 tail wags (per min)**  
Comment from your pet: **You should give me a treat first!**

By inspecting the name “Pakkun”, I can change it to “Kakashi”.

The same web application interface as before, but now the name has been changed to “Kakashi”. The rest of the information remains the same. To the right of the dashboard, a developer tools sidebar shows the DOM structure and the CSS styles applied to the elements.

**DOM Structure:**

```
<!-- Replace pet image with your own pet image -->

</section>
<section class="dashboard">
  <div>
    " Name: "
    <strong>
      <span class="name">Kakashi</span> == $0
    </strong>
  </div>
  " Weight: "
  <strong>...</strong>
</div>
<div>
  " Happiness: "
  <strong>50 tail wags (per min)</strong>
</div>
<div>
  " Comment from your pet: "
  You should give me a treat first!
</div>
```

**Styles:**

```
special-foods-menu
element.style { }
```

**Console:**

```
: Console AI assistance
```

Next is **Edit attributes**

If I inspect the name “Kakashi,” I can edit the attribute by setting the background color of that span to be gold.

This is your Giga Pet!

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator

```
<!-- Replace pet image with your own pet image -->

</section>
<section class="dashboard">
  <div>
    " Name: "
    <strong>
      <span class="name" style="background-color:gold">Kakashi</span> == $0
    </strong>
  </div>
  <div>
    " Weight: "
    <strong>20</strong>
  </div>
  <div>
    " Happiness: "
    <strong>50 tail wags (per min)</strong>
  </div>
</section>
```

This is your Giga Pet!

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator

```
element.style {
  background-color: #gold;
}
```

Next is **Edit node type**

I can change the h1 title “This is your Giga Pet” to a button.

This is your Giga Pet!

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

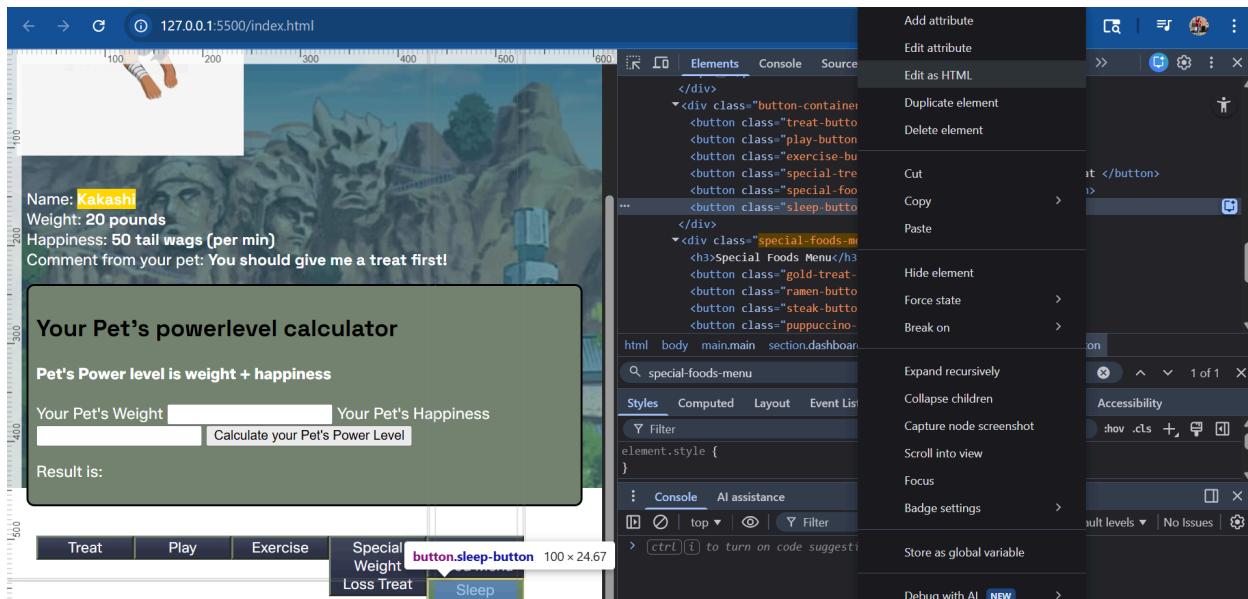
h1 600 x 40.67

```
</head>
<body data-new-gr-c-s-check-loaded="14.1264.0" data-gr-ext-installed>
  <header>
    <button>This is your Giga Pet!</button> == $0
  </header>
  <main class="main">
    <section class="pet-image-container">
      <!-- Replace pet image with your own pet image -->
      
    </section>
    <section class="dashboard">
      <div>
        " Name: "
        <strong>
          <span class="name" style="background-color: #000; color: #fff">Kakashi</span>
        </strong>
      </div>
    </section>
  </main>
</body>
```

This is your Giga Pet!

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

## Next is Edit as HTML

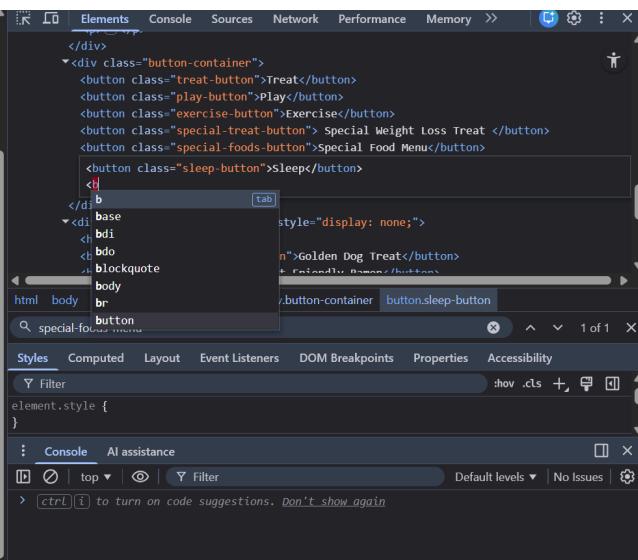


The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html. The page displays a pet's profile for "Kakashi" with the following details:

- Name: Kakashi
- Weight: 20 pounds
- Happiness: 50 tail wags (per min)
- Comment from your pet: You should give me a treat first!

A modal window titled "Your Pet's powerlevel calculator" is open, containing a form to calculate power level based on weight and happiness.

The developer tools' Elements tab is open, focusing on the "button.sleep-button" element. The element's style is highlighted with a yellow background and black border. The element itself is a blue button labeled "Sleep".



The screenshot shows the same browser window and developer tools interface, but now the "sleep" button has been removed from the DOM. The "special-foods-menu" div is styled with "display: none;" in its CSS. The developer tools status bar indicates "Default levels" and "No Issues".

```
<button class="exercise-button">Exercise</button>
<button class="special-treat-button"> Special Weight Loss Treat </button>
<button class="special-foods-button">Special Food Menu</button>
<button class="sleep-button">Sleep</button>
<button>Testing/random</button>

```

I added a new button by using edit by HTML.

Name: **Kakashi**  
Weight: **20 pounds**  
Happiness: **50 tail wags (per min)**  
Comment from your pet: **You should give me a treat first!**

## Your Pet's powerlevel calculator

Pet's Power level is weight + happiness

Your Pet's Weight  Your Pet's Happiness   
Calculate your Pet's Power Level

Result is:

Treat | Play | Exercise | Special Weight Loss Treat | Special Food Menu | Sleep | Testing/random

### Next is Duplicate a node

I will be duplicating the “Treat” button.

127.0.0.1:5500/index.html

Name: **Kakashi**  
Weight: **20 pounds**  
Happiness: **50 tail wags (per min)**  
Comment from your pet: **You should give me a treat first!**

## Your Pet's powerlevel calculator

Pet's Power level is weight + happiness

Your Pet's Weight  Your Pet's Happiness   
Calculate your Pet's Power Level

Result is:

button.treat-button 100 x 24.67

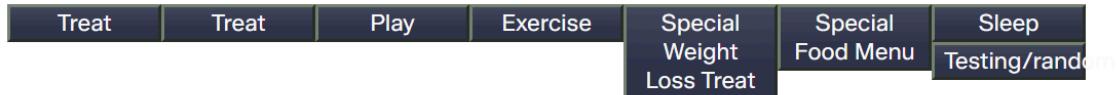
Treat | Play | Exercise | Special Weight Loss Treat | Special Food Menu | Sleep | Testing/random

The screenshot shows a browser window with a pet calculator application. The developer tools are open, specifically the context menu for a 'Treat' button. The 'Duplicate element' option is highlighted. The page content includes a pet profile for 'Kakashi' and a calculator interface. A tooltip for the 'treat' button indicates its dimensions as 100 x 24.67.

**Pet's Power level is weight + happiness**

Your Pet's Weight  Your Pet's Happiness

Result is:



### Next is Capture a node screenshot

I will be capturing a node screenshot of the “dashboard”.

Name: **Kakashi**  
 Weight: **20 pounds**  
 Happiness: **50 tail wags (per min)**  
 Comment from your pet: **You should give me a treat first!**

**Your Pet's powerlevel calculator**

**Pet's Power level is weight + happiness**

Your Pet's Weight  Your Pet's Happiness

Result is:



## Next is Reorder DOM nodes

I will be bringing the “special food menu” button to the front of the group of buttons.  
before:

Name: **Kakashi**  
Weight: **20 pounds**  
Happiness: **50 tail wags (per min)**  
Comment from your pet: You should give me a treat first!

### Your Pet's powerlevel calculator

Pet's Power level is weight + happiness

Your Pet's Weight  Your Pet's Happiness

Result is:

Treat   Treat   Play   Exercise   Special Weight Loss Treat   Special Food Menu   Sleep   Testing/random

```
<input type="number" id="input2">
<button id="calc-button">Calculate your Pet's Power Level</button>
<p>...</p>
</div>
<div class="button-container">
  <button class="treat-button">Treat</button>
  <button class="treat-button">Treat</button>
  <button class="play-button">Play</button>
  <button class="exercise-button">Exercise</button>
  <button class="special-treat-button"> Special Weight Loss Treat
    </button>
  ...
  <button class="special-foods-button">Special Food Menu
    </button> == $0
  <button class="sleep-button">Sleep</button>

```

After:

The screenshot shows a web application interface. On the left, there's a pet profile for 'Kakashi' with stats: Name: Kakashi, Weight: 20 pounds, Happiness: 50 tail wags (per min). A comment from the pet says: You should give me a treat first! Below this is a calculator titled 'Your Pet's powerlevel calculator' with fields for 'Your Pet's Weight' and 'Your Pet's Happiness', and a button to 'Calculate your Pet's Power Level'. To the right is a 'button-container' containing several buttons: 'Special Food Menu', 'Treat', 'Treat', 'Play', 'Exercise', 'Special Weight Loss Treat', 'Sleep', and 'Testing/random'. The 'Treat' button is highlighted with a green border. On the far right, the browser's developer tools are open, specifically the Elements tab, showing the HTML structure of the page, including the button definitions.

## Next is Force state

I will make the treat button have a force state of “hover”. When I hover over the button, it has a green highlight initially. After the force state, it will have a green highlight at all times.

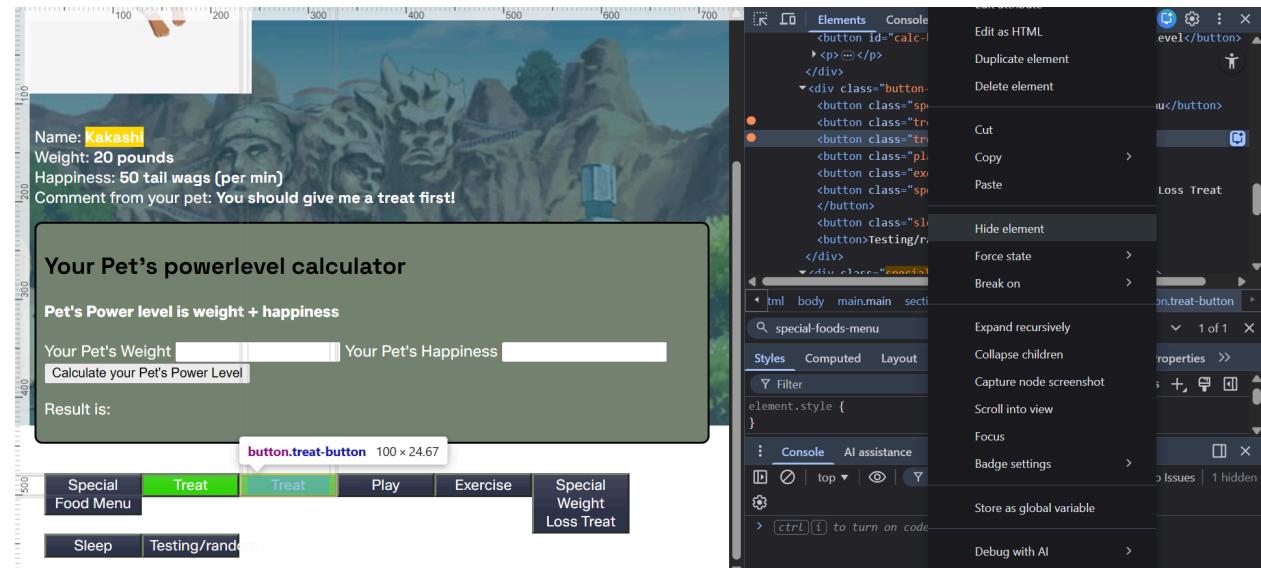
The screenshot shows the same web application after applying a CSS rule to the 'Treat' button. The 'Treat' button now has a green background color when the mouse hovers over it. The other buttons ('Special Food Menu', 'Play', 'Exercise', 'Special Weight Loss Treat', 'Sleep', 'Testing/random') remain dark grey. The developer tools on the right show the CSS rule applied to the 'Treat' button.

After:

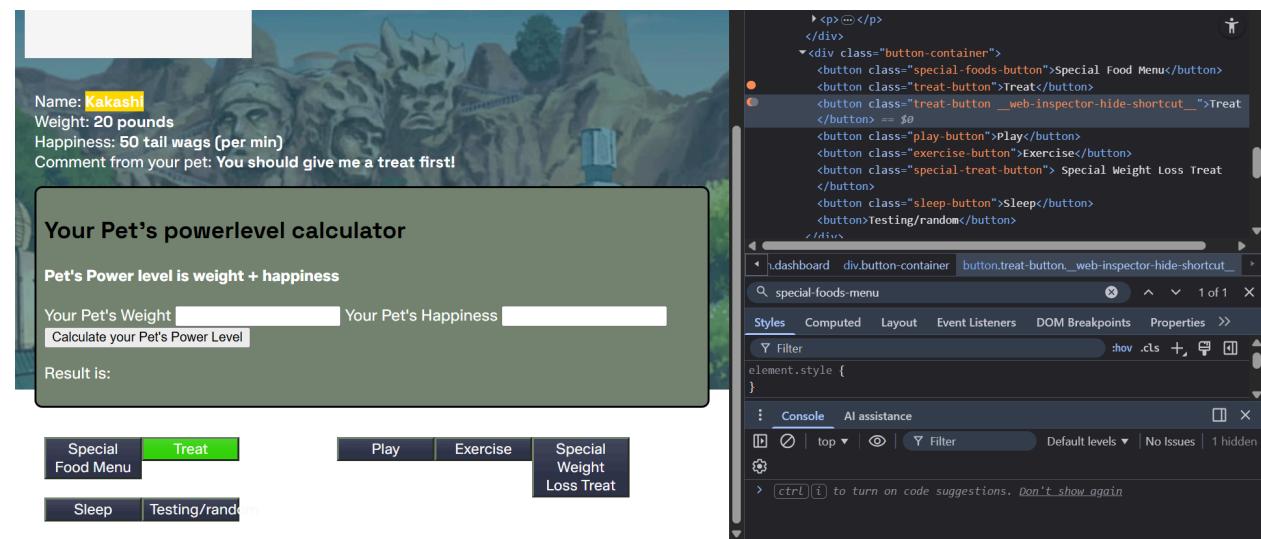
The screenshot shows the final state of the application. The 'Treat' button is now consistently green, indicating it has entered a permanent force state. The other buttons ('Special Food Menu', 'Play', 'Exercise', 'Special Weight Loss Treat', 'Sleep', 'Testing/random') are dark grey. The developer tools on the right show the CSS rule applied to the 'Treat' button.

## Next is Hide a node

I will now hide the extra Treat button.



After:



## Next is to Delete a node.

I will delete the treat button that isn't hidden.

## Before:

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator  
Pet's Power level is weight + happiness  
Your Pet's Weight  Your Pet's Happiness   
Calculate your Pet's Power Level  
Result is:

Special Food Menu    Treat    Play    Exercise    Special Weight Loss Treat  
Sleep    Testing/random

Code View (Web Inspector):

```
<button class="treat-button">Treat</button>
<button class="treat-button __web-inspector-hide-shortcut__">Treat</button>
<button class="play-button">play</button>
<button class="exercise-button">Exercise</button>
<button class="special-treat-button">Special Weight Loss Treat</button>
<button class="sleep-button">Sleep</button>
<button>Testing/random</button>
```

Styles Computed Layout Event Listeners DOM Breakpoints Properties >

Console AI assistance

ctrl ⌘ to turn on code suggestions. Don't show again

## After:

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator  
Pet's Power level is weight + happiness  
Your Pet's Weight  Your Pet's Happiness   
Calculate your Pet's Power Level  
Result is:

button.treat-button 100 x 24.67

Special Food Menu    Treat    Play    Exercise    Special Weight Loss Treat  
Sleep    Testing/random

Contextual menu (right-clicked on the Treat button):

- Add attribute
- Edit 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
- Debug with AI

Code View (Web Inspector):

```
<div class="button-container">
<button class="special-foods-button">Special Food Menus</button>
<button class="treat-button __web-inspector-hide-shortcut__">Treat</button>
</button> == $0
<button class="play-button">play</button>
<button class="exercise-button">Exercise</button>
<button class="special-treat-button">Special Weight Loss Treat</button>
<button class="sleep-button">Sleep</button>
<button>Testing/random</button>
```

Styles Computed Layout Event Listeners DOM Breakpoints Properties >

Console AI assistance

ctrl ⌘ to turn on code suggestions. Don't show again

Name: **Kakashi**  
Weight: 20 pounds  
Happiness: 50 tail wags (per min)  
Comment from your pet: You should give me a treat first!

Your Pet's powerlevel calculator  
Pet's Power level is weight + happiness  
Your Pet's Weight  Your Pet's Happiness   
Calculate your Pet's Power Level  
Result is:

Special Food Menu    Play    Exercise    Special Weight Loss Treat  
Sleep    Testing/random

Code View (Web Inspector):

```
<div class="button-container">
<button class="special-foods-menu">Special Food Menus</button>
<button class="treat-button __web-inspector-hide-shortcut__">Treat</button>
</button> == $0
<button class="play-button">play</button>
<button class="exercise-button">Exercise</button>
<button class="special-treat-button">Special Weight Loss Treat</button>
<button class="sleep-button">Sleep</button>
<button>Testing/random</button>
```

Styles Computed Layout Event Listeners DOM Breakpoints Properties >

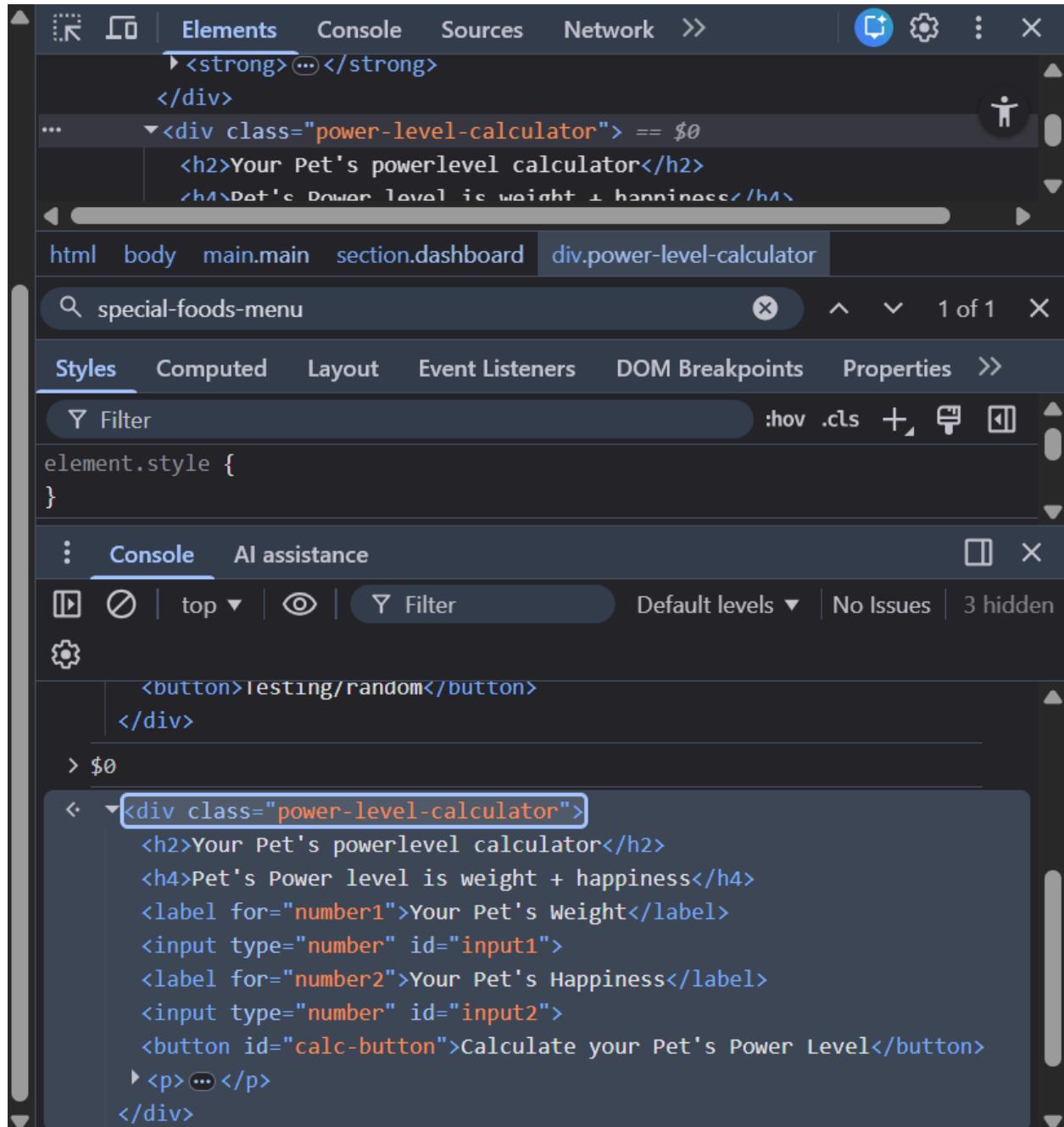
Console AI assistance

ctrl ⌘ to turn on code suggestions. Don't show again

## The next section is Access nodes in the Console

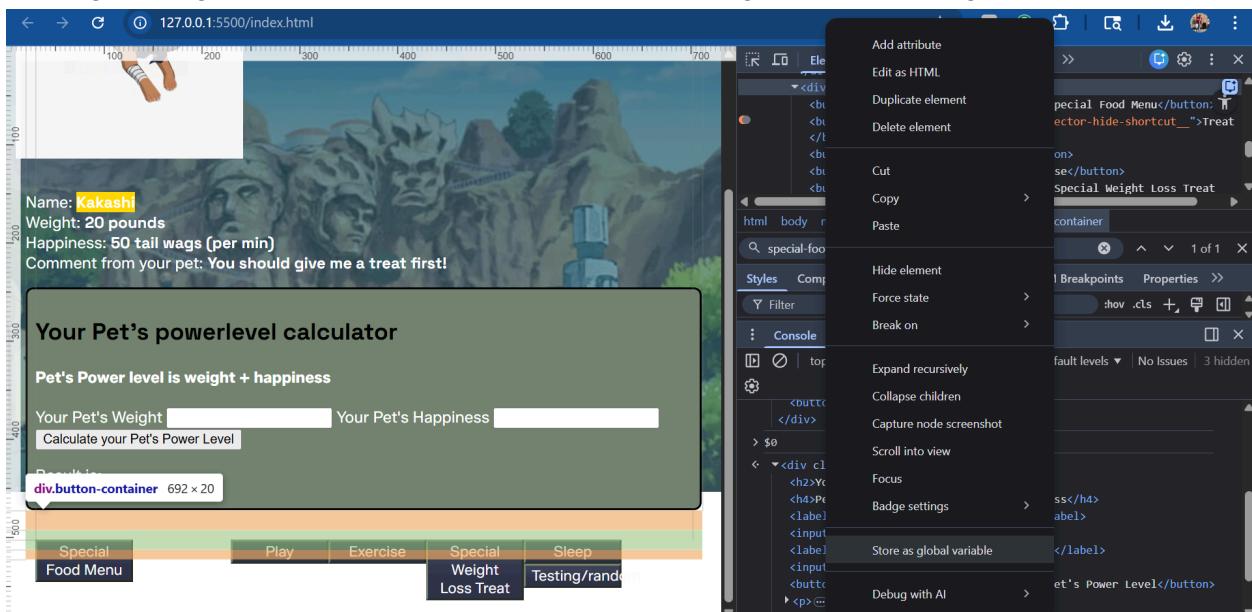
First is Reference the currently-selected node with \$0

Inspecting the power level calculator section and typing \$0 in the console, I get this:



## Next is **Store as global variable**

Looking at the group of buttons, I will make the div holding them all as the global variable.

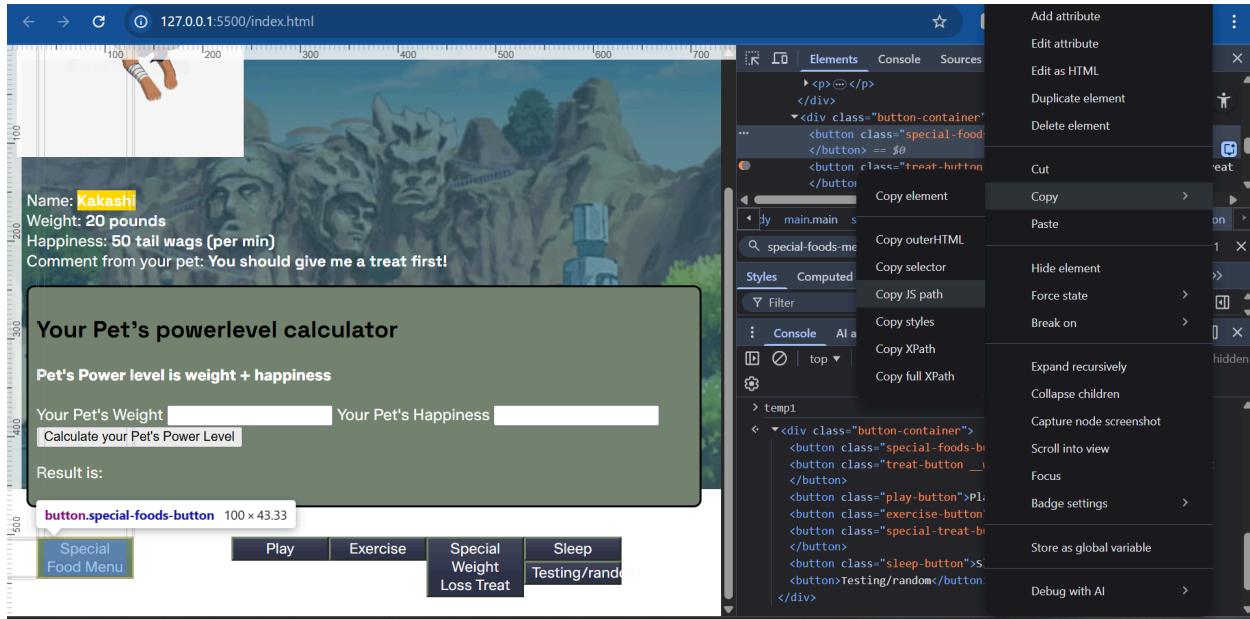


Then typing temp1

```
> temp1
<div class="button-container">
  <button class="special-foods-button">Special Food Menu</button>
  <button class="treat-button __web-inspector-hide-shortcut__">Treat
  </button>
  <button class="play-button">Play</button>
  <button class="exercise-button">Exercise</button>
  <button class="special-treat-button"> Special Weight Loss Treat
  </button>
  <button class="sleep-button">Sleep</button>
  <button>Testing/random</button>
</div>
```

## Next is **Copy JS path**

I will copy the JS path from the special foods button.



Putting the copied path in the console gives me this:

```

::: Console AI assistance
button.special-foods-button 100x43.33
Default levels ▾ | No Issues | 3 hidden
button.special-foods-button
<button class="play-button">Play</button>
<button class="exercise-button">Exercise</button>
<button class="special-treat-button"> Special Weight Loss Treat
</button>
<button class="sleep-button">Sleep</button>
<button>Testing/random</button>
</div>
> document.querySelector("body > main > section.dashboard > div.button-container > button.special-foods-button")
<span style="color: #0070C0;">button.special-foods-button >Special Food Menu</button>
>

```