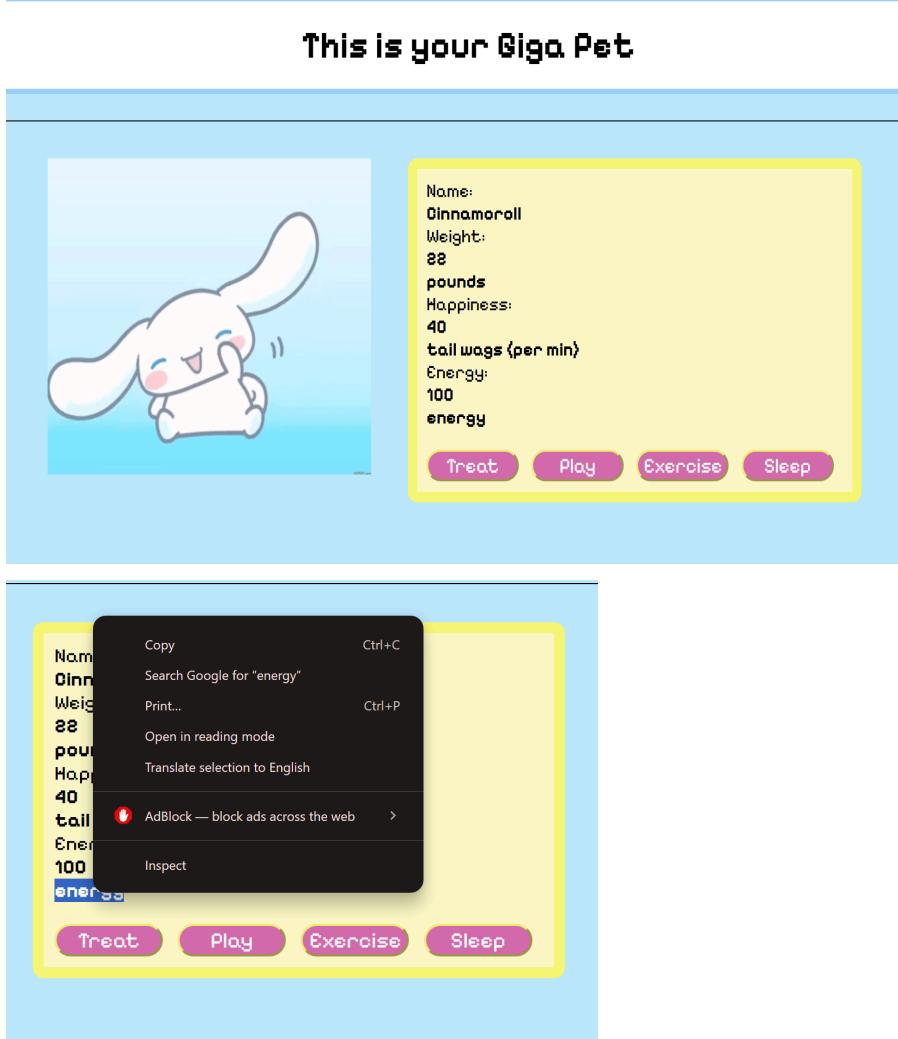


Editing a Project by Changing the DOM

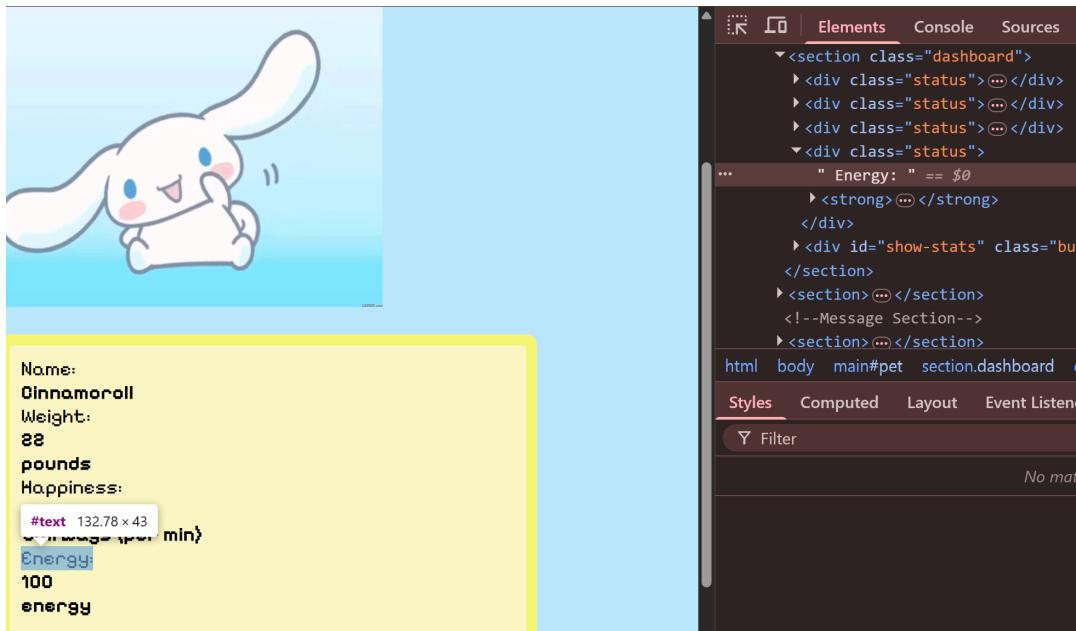
In Chrome, we can use DevTools to access the DOM of any webpage. This allows users to make temporary structural changes to a website until the page is refreshed. During this process, the browser parses the HTML and creates a tree of objects that we can access and edit through Chrome DevTools.

Access DevTools

We can access DevTools by right-clicking on the website and selecting **Inspect**:



This opens up Chrome's DevTools:



DevTools opens a panel that gives users access to several tools, including the **Elements** tab, which displays the DOM, and the **Console** tab, where users can run JavaScript commands for debugging. Note that changes made in the Console are not saved. The **Sources** tab is where programmers can view and edit files, and temporarily save changes during debugging.

Right now, I'll work on editing the nodes in the DOM to change the style of the webpage. I can edit the DOM directly in the **Elements** tab, so I am going to change the colors of the heading.

Search Tools

Currently, I am inspecting the node that contains the text *energy*, but I can't see the webpage heading in the viewport. There are a few easy shortcuts to find the header quickly, especially on long websites. For example, we can press **Ctrl + F** to search for the **h1** element.

Name:
Cinnamoroll
Weight:
22
pounds
Happiness:
40
tail wags (per min)
Energy:
100
energy

Treat Play Exercise Sleep

```

<div class="status">...</div>
<div class="status">
  "Energy: "
...
  <strong>...</strong> == $0
</div>
<div id="show-stats" class="button-container">...</div>
</section>
<section>...</section>
<!--Message Section-->
<section>...</section>

```

section.dashboard div.status strong

Find by string, selector, or

Styles Computed Layout >

element.style {

.dashboard div, span, style.css:74

strong {

display: block;
background-color: #fcf9c5;

* {

box-sizing: border-box;
background-color: #b2eaff;

This will bring **h1** into the viewport:

This is your Giga Pet

Name:
Cinnamoroll
Weight:
22
pounds
Happiness:
40
tail wags (per min)
Energy:
100
energy

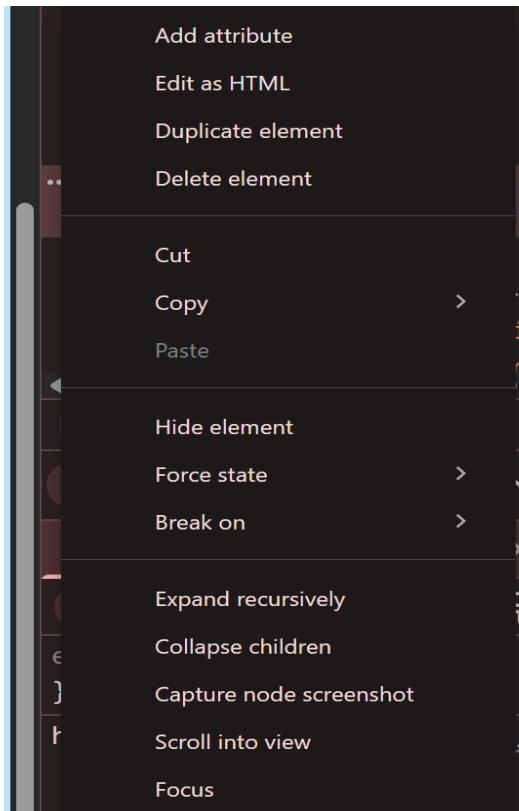
Treat Play Exercise Sleep

```

<h1>This is your Giga Pet
</h1> == $0
</header>
<!-- This is the pet -->
<main id="pet" role="main">
  <section class="pet-image-container">
    <!-- Pet image with your own pet image -->
    
</script>
</head>
<body>
  <header>
    ...   <h1>This is your Giga Pet
          </h1> == $0
    </header>
    <!-- This is the pet -->
    <main id="pet" role="main">
      <section class="net-image-con">
        ...
        <img alt="A white cartoon dog with long ears, pink cheeks, and a small bow tie, looking happy and slightly tilted to the side." data-bbox="120 110 476 308"/>
      </section>
      <div>
        <table border="1">
          <tr>
            <td>Name:</td>
            <td>Cinnamoroll</td>
          </tr>
          <tr>
            <td>Weight:</td>
            <td>22  
pounds</td>
          </tr>
          <tr>
            <td>Happiness:</td>
            <td>40</td>
          </tr>
        </table>
      </div>
    </main>
  </body>
</html>
```

Styles Computed Layout >>

element.style {
}
h1 { style.css:31
 font-weight: bold;



This is your Giga Pet

Name: Ginnamoroll
Weight: 22 pounds
Happiness: 40
tail wags (per min)
Energy: 100 energy

Treat Play Exercise Sleep

```
... <h1>This is your Giga Pet</h1> == $0
</header>
<!-- This is the pet --&gt;
&lt;main id="pet" role="main"&gt;
  &lt;section class="pet-image-container"&gt;
    &lt;!-- Pet image with your own pet image --&gt;
    &lt;img class="pet-image" src="ht...&gt;
&lt;/main&gt;
&lt;script&gt;...&lt;/script&gt;</pre>

html body header h1



Styles Computed Layout >>



element.style { } h1 { font-weight: bold; color: #000; }


```

Now that I can see the heading of the webpage, I have two options for editing it: I can edit it directly in the DOM, or I can modify it through the Console tab.

Editing DOM

I'll start by editing the style of the heading in the DOM. To do this, I click on the **h1** node and add a new background color, border color, and text color:

```
com/jquery-3.7.1.js" ></script>
<script src="script.js" def>
</script>
</head>
<body>
  <header>
    ... <h1>This is your Giga Pet
    </h1> == $0
  </header>
  <!-- This is the pet -->
  <main id="pet" role="main">

```

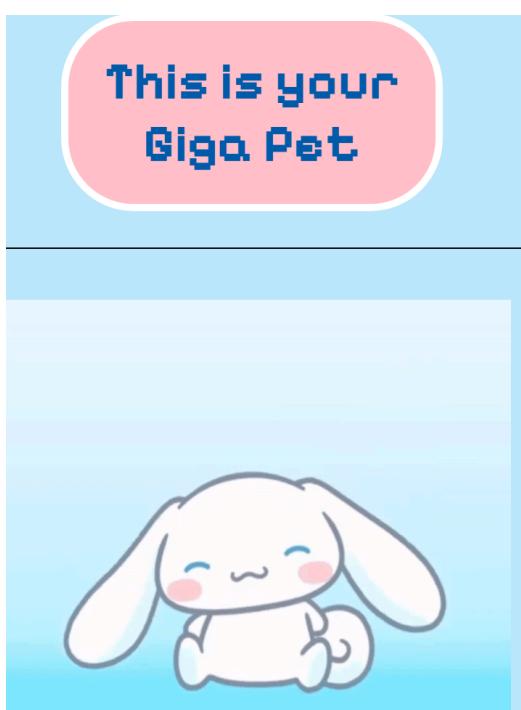
html body header h1

Search: this 1 of 2

Styles Computed Layout >

element.style {

h1 { style.css:31



```
<head>
<body>
  <header>
    <h1 style="background: pink; color: #005aab; border: 5px solid white;">This is your Giga Pet
    </h1> == $0
  </header>
  <!-- This is the pet -->
  <main id="pet" role="main">

```

html body header h1

Search: this 1 of 2

Styles Computed Layout >

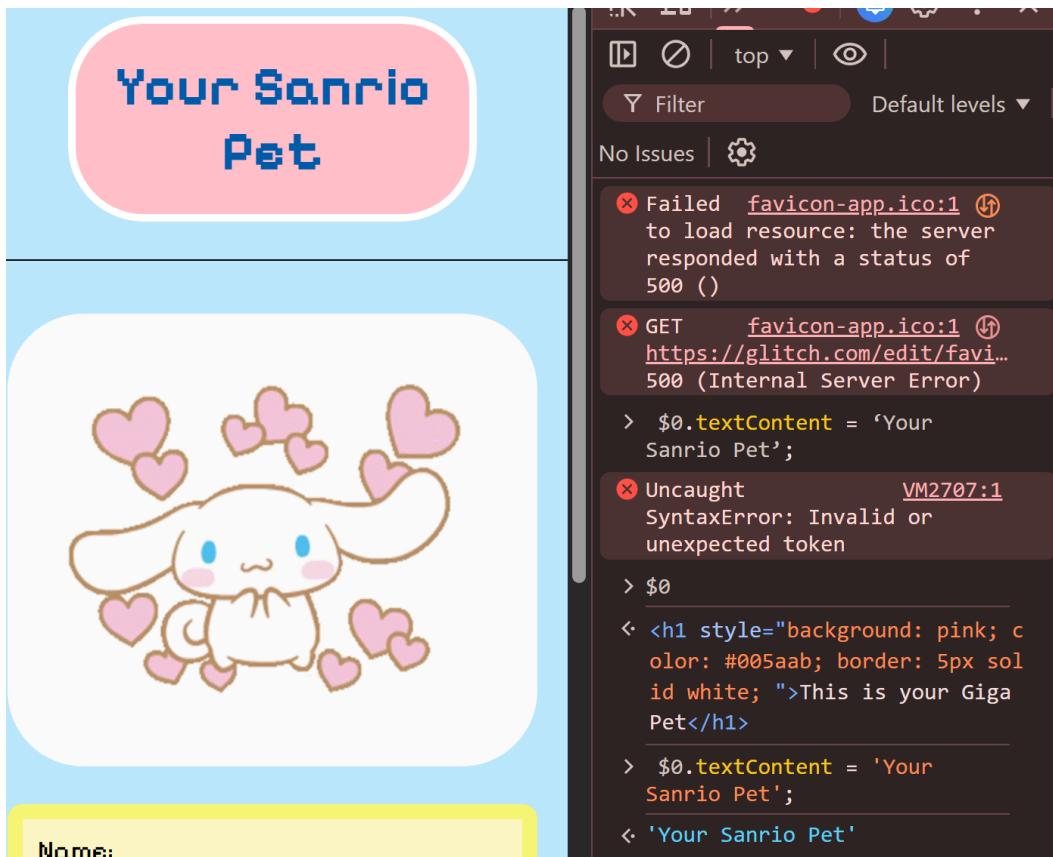
element.style {

background: ▶ pink;
color: □ #005aab;
border: ▶ 5px solid □ white;

I can also edit the website using the Console. For example, if I want to change the text in the **h1**, I can click on the Console tab and type:

```
$0.textContent = 'Your Sanrio Pet';
```

I used **\$0** because it refers to the currently selected node in the DOM, which is the **h1** at the moment. It allows the user to access and modify whatever that node contains.

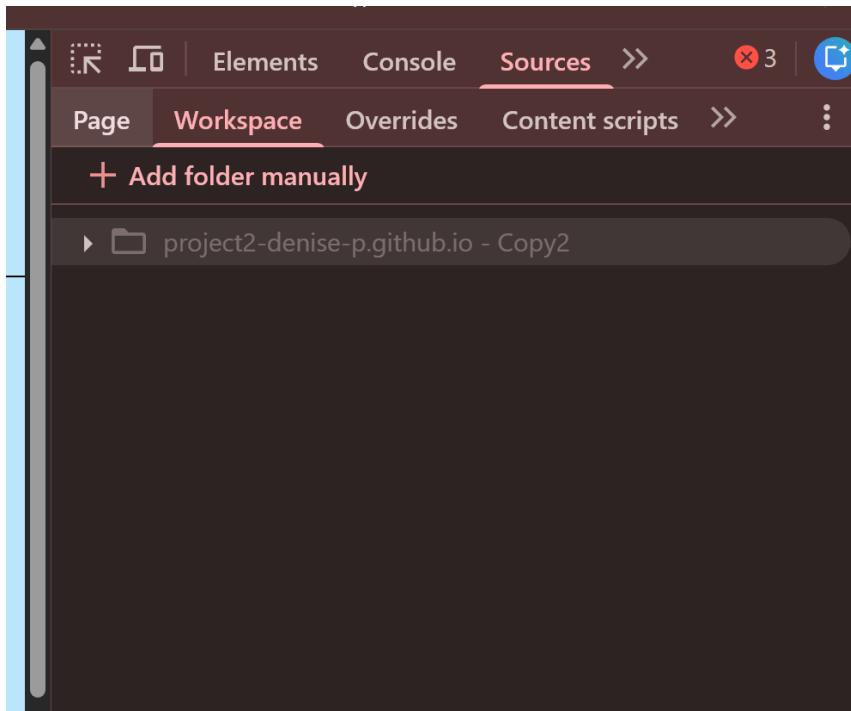


The changes to the website are coming along nicely, but once the webpage is refreshed, all of the modifications are lost.

Permit Changes

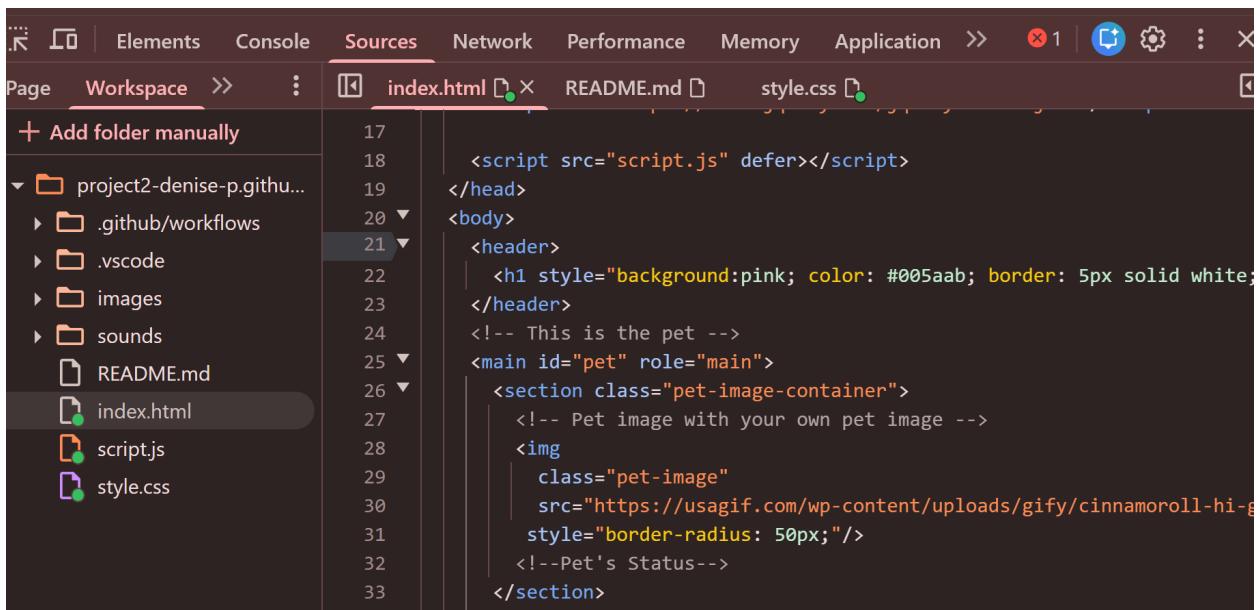
If a user wants to keep the edited code, the user needs to save changes in the workshop.

The first step to saving the changes is to click on the **Sources** tab.



In the Source tab there is a workshop section where you can save changes. If it is empty, I can add the folder containing my code through “Add folder manually”.

Next, I can open my folder and make changes directly to the DOM. (Note: Any changes made in the workspace will be permanently saved to the files in your folder.)



Before:

```
<script src="script.js" defer></script>
</head>
<body>
<header>
  <h1>This is your Giga Pet</h1>
</header>
<!-- This is the pet -->
<main id="pet" role="main">
  <section class="pet-image-container">
    <!-- Pet image with your own pet image -->
    
    <!--Pet's Status-->
  </section>
```

After:

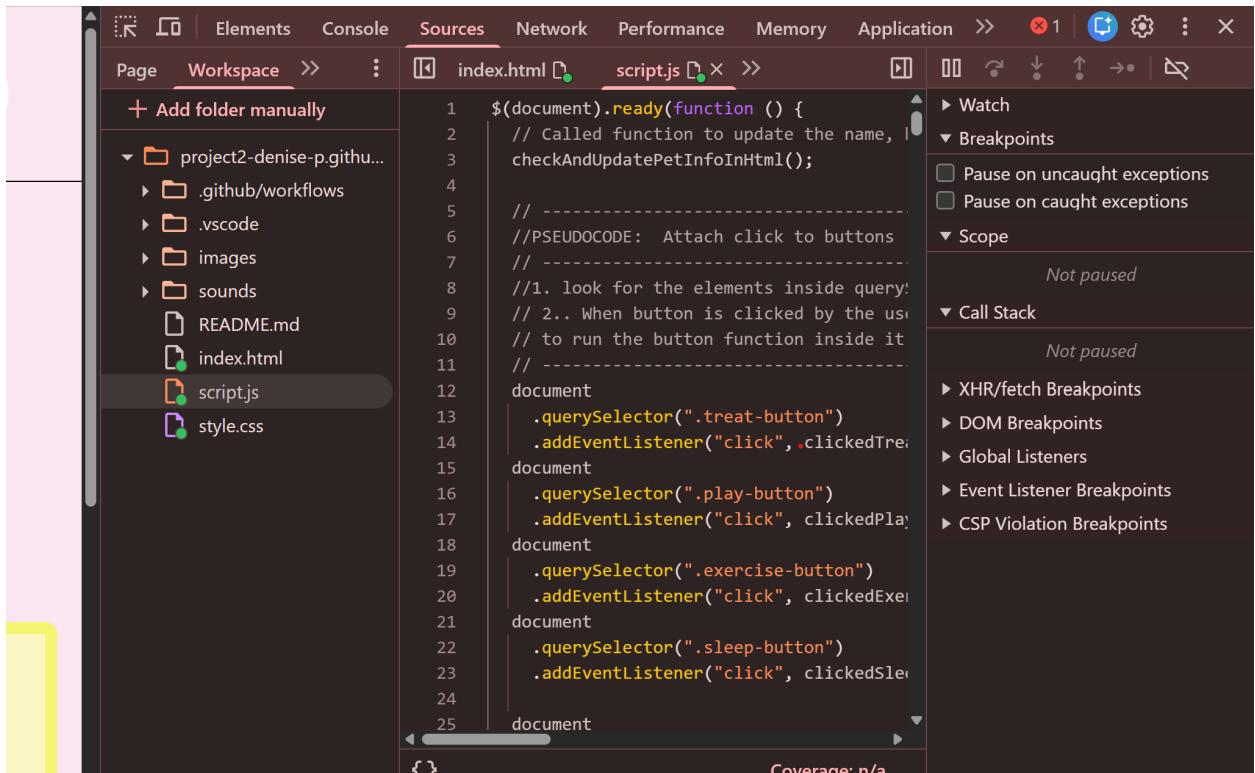
```
<script src="script.js" defer></script>
</head>
<body>
<header>
  <h1 style="background: #pink; color: #005aab; border: 5px solid white;">Your Sanrio Pet</h1>
</header>
<!-- This is the pet -->
<main id="pet" role="main">
  <section class="pet-image-container">
    <!-- Pet image with your own pet image -->
    
    <!--Pet's Status-->
  </section>
```

Debugging Code

In Chrome DevTools, we can not only edit the DOM but also debug any JavaScript issues.

The main feature I want to debug is the click function. It works, but I want the pet's energy to run out faster, so I will adjust the button's functions to consume more energy.

I'll start debugging by opening DevTools and navigating to the **Sources** tab.



Next, I go into the Workspace section to make sure I am in the **script.js** file. After that, I want to set a breakpoint on the click events to locate the function. To do this, I open **Event Listener Breakpoints**, expand the **Mouse** category, and select **click**:

The screenshot shows the Chrome DevTools Sources tab with the script.js file open. The right sidebar shows the Event Listener Breakpoints panel with the 'click' event selected under the Mouse category.

Anytime I click a button, the code will pause on the first line of the click event that executes. For example, here I clicked the exercise button:

The screenshot shows a browser developer tools window with the 'Sources' tab selected. A file named 'script.js' is open, showing a function 'clickedPlayButton()' with several lines of code. The line 'pet_info.happiness += 10;' is highlighted with a yellow box. The status bar at the top of the browser window says 'Paused in debugger'.

Now I am inside the clickedPlayButton function and can adjust the energy variable to remove more energy from pet_info.energy. By clicking the **Next** button, I can step to the line where the energy decreases by 10.

```
189  function clickedPlayButton() {  
190      // Increase pet happiness  
191      pet_info.happiness += 10;  
192      // Decrease pet weight  
193      pet_info.weight -= 1;  
194      //Decrease pet energy  
195      pet_info.energy -= 10;  
196      //change image
```

Since I found the energy variable, I will unpause the code, change the value to 15, and save the update by pressing **Ctrl + S**.

```
188 //  
189 function clickedPlayButton() {  
190     // Increase pet happiness  
191     pet_info.happiness += 10;  
192     // Decrease pet weight  
193     pet_info.weight -= 1;  
194     //Decrease pet energy  
195     pet_info.energy -= 15;  
196     //change image  
197     $(".pet-image").fadeOut(400, function ()  
198         | $(this).attr("src", "images/plays.gif")  
199     );  
200     //play sound  
201     playSound("sound-play", 0.2, 5000);  
202     //play message  
203     popUp("Play Time!");  
204     //check stats  
205     checkAndUpdatePetInfoInHtml();
```

Energy now decreases by 15 when clicking the play button.

By looking through the console tab or checking the viewport I can view the changes

The screenshot shows a browser window with developer tools open. On the left, there's a large, rounded rectangular icon featuring a white cartoon dog with blue eyes and a pink nose, set against a light blue gradient background. To the right of this icon is a yellow rectangular card containing pet information:

Name:	Cinnamoroll
Weight:	21 pounds
Happiness:	80 tail wags (per min)
Energy:	88 energy

On the far right, the browser's developer tools are visible, specifically the 'Console' tab. The console output shows the following:

```
✖ GET https://glitch.com/edit/favicon-app.ico 500 (Internal Server Error)  
✖ GET https://glitch.com/edit/favicon-app.ico 500 (Internal Server Error)  
> pet_info  
< {name: 'Cinnamoroll', weight: 21, happiness: 80, energy: 85}
```

The new code also saves in VScode since we saved in the **Workshop** section:



JS script.js > ...

```
185 // 8. After 5 seconds (when the animation finishes):
186 //     - Mark that no action animation is playing.
187 //     - Run the emotions() function to decide the pet's mood.
188 // -----
189 function clickedPlayButton() {
190     // Increase pet happiness
191     pet_info.happiness += 10;
192     // Decrease pet weight
193     pet_info.weight -= 1;
194     //Decrease pet energy
195     pet_info.energy -= 15;
196     //change image
197     $(".pet-image").fadeOut(400, function () {
```