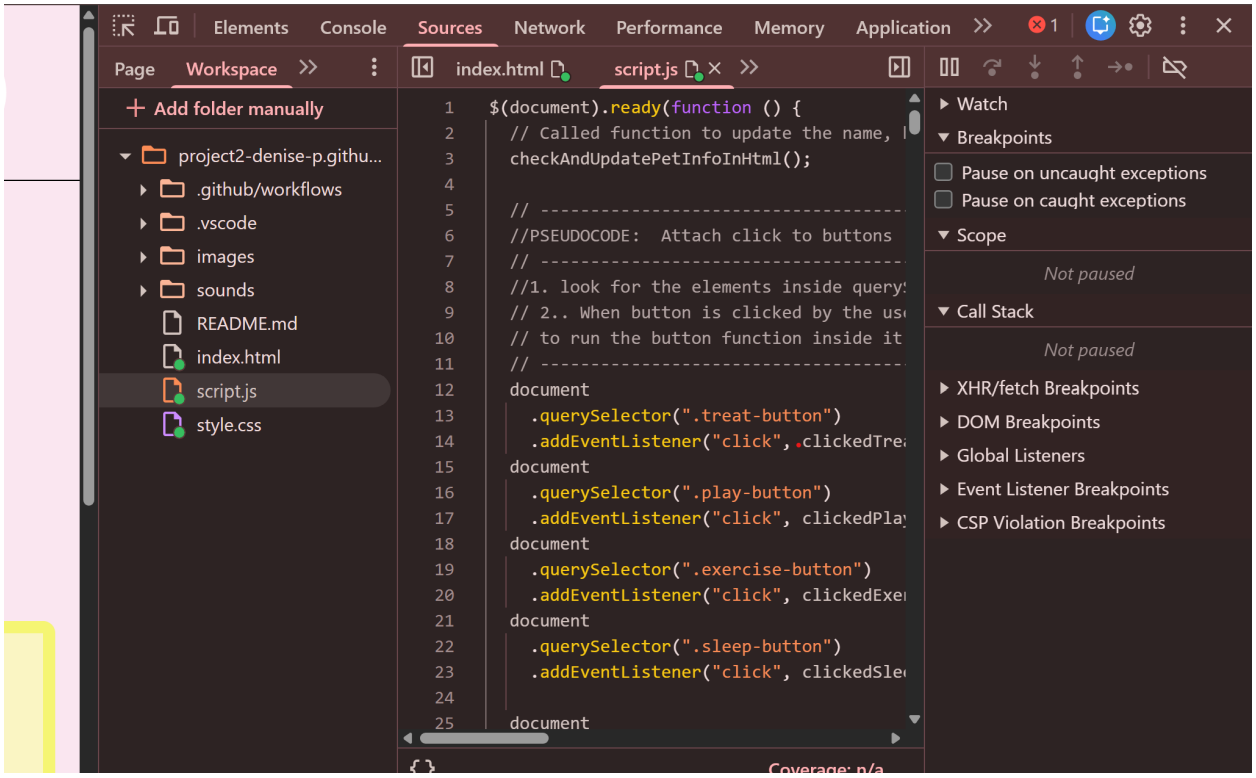
Debugging Code

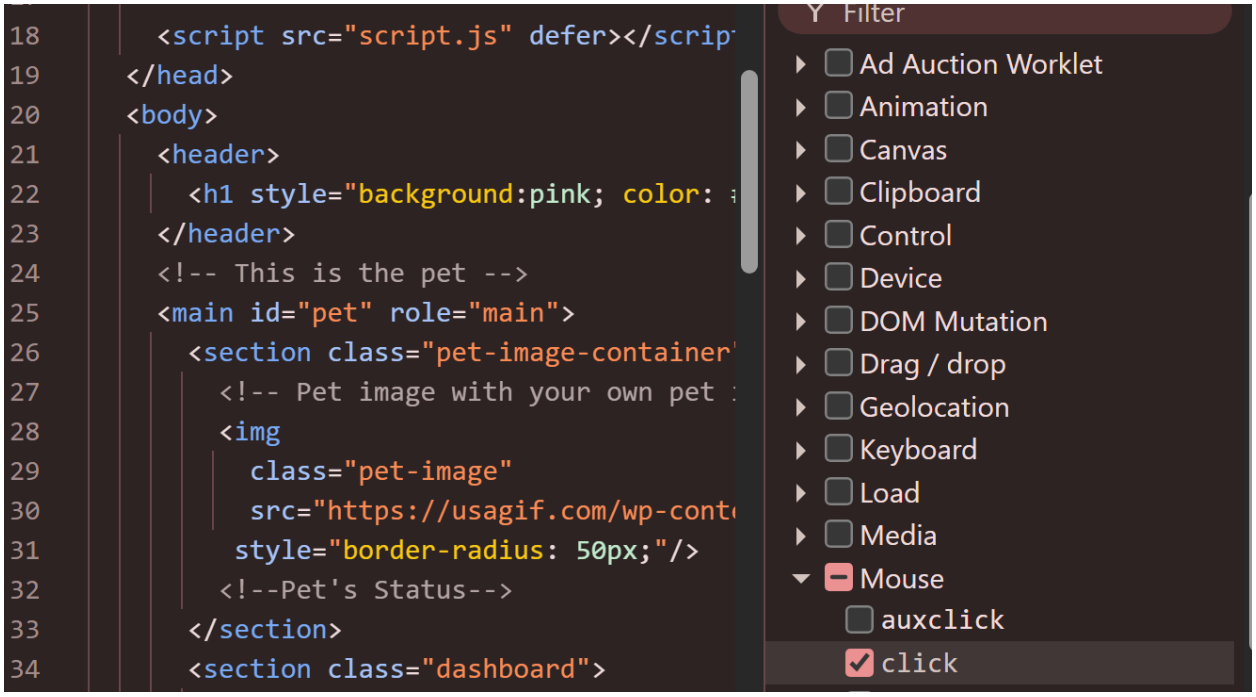
In Chrome DevTools, we can not only edit the DOM but also debug any JavaScript issues.

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.

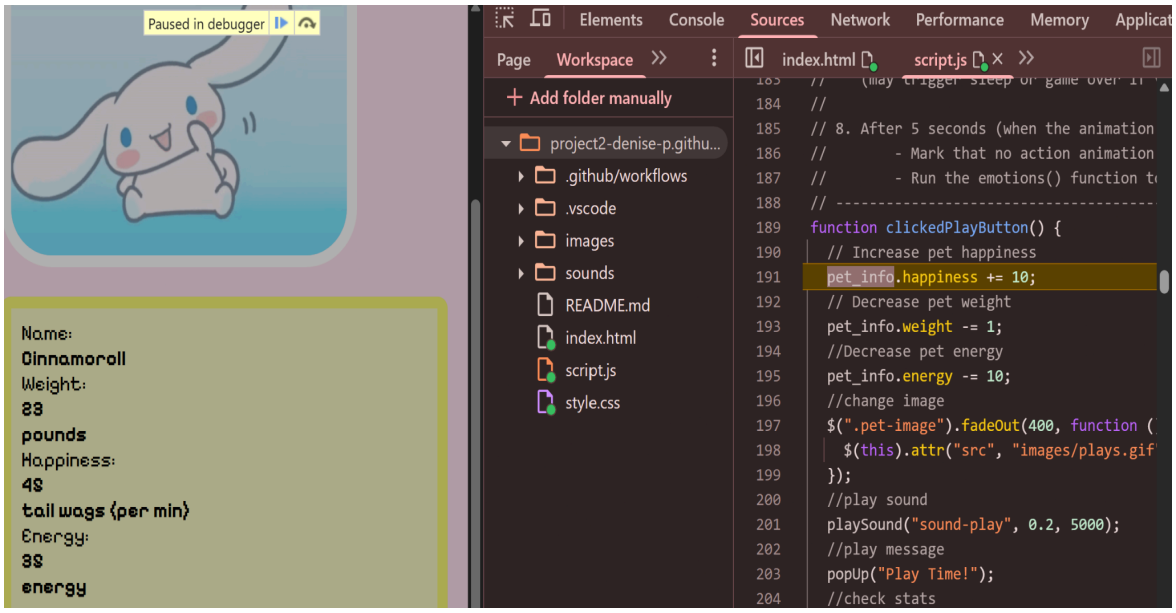
I'll start debugging by opening DevTools and navigating to the **Sources** tab.



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**:



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:



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.

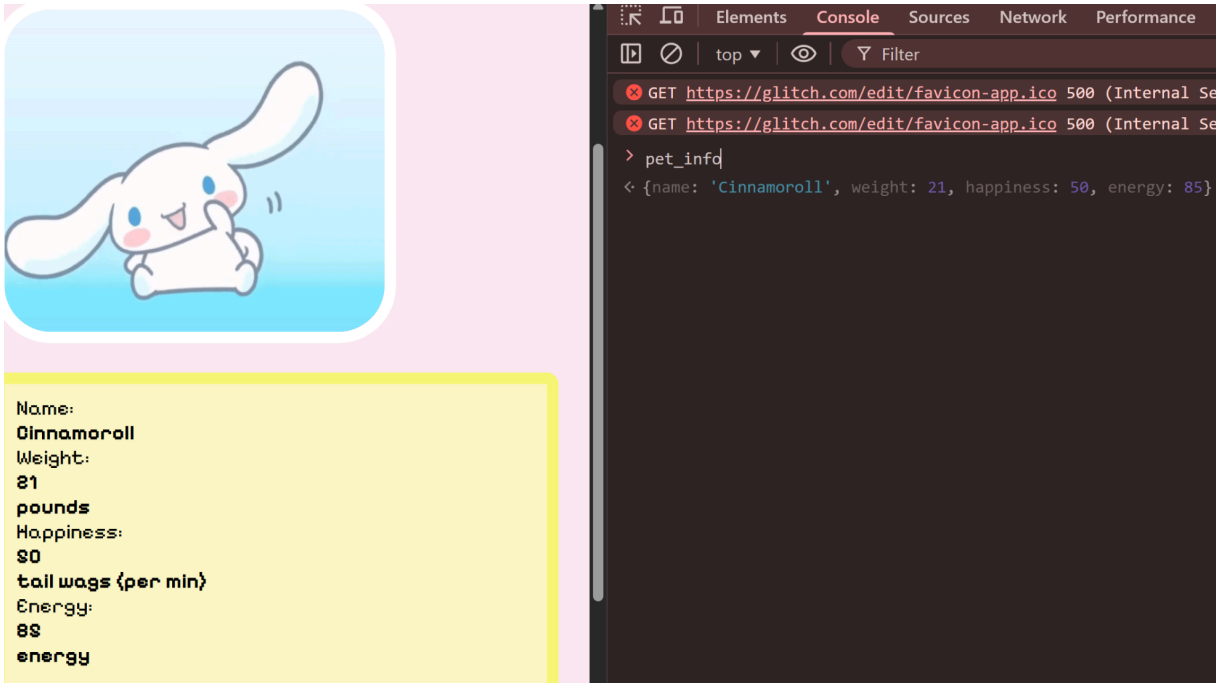
```
189  function clickedPlayButton() {
190    // Increase pet happiness
191    pet_info.happiness += 10;
192    // Decrease pet weight
193    pet_info.weight -= 1;
194    //Decrease pet energy
195    pet_info.energy -= 10;
196    //change image
```

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

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