

# Chrome DevTools for GigaPet

## Part 1: Debugging JavaScript

### 1) Reproduce the Example Bug

A bug is shown in a commented out part of the function `clickedExerciseButton`. The incorrect line is:

```
Pet_info.hunger += 10
```

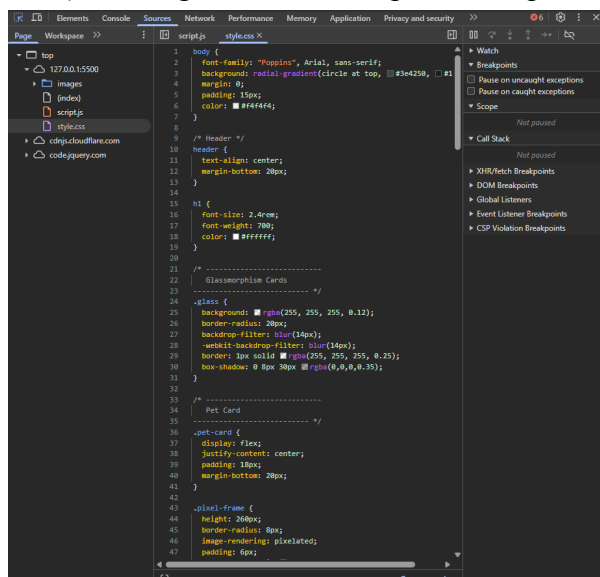
This causes hunger to increase instead of exercising

To reproduce

- 1) Open `index.html` in the browser
- 2) Open DevTools
- 3) Click Exercise Repeatedly
- 4) Observe the hunger bar going up instead of down.

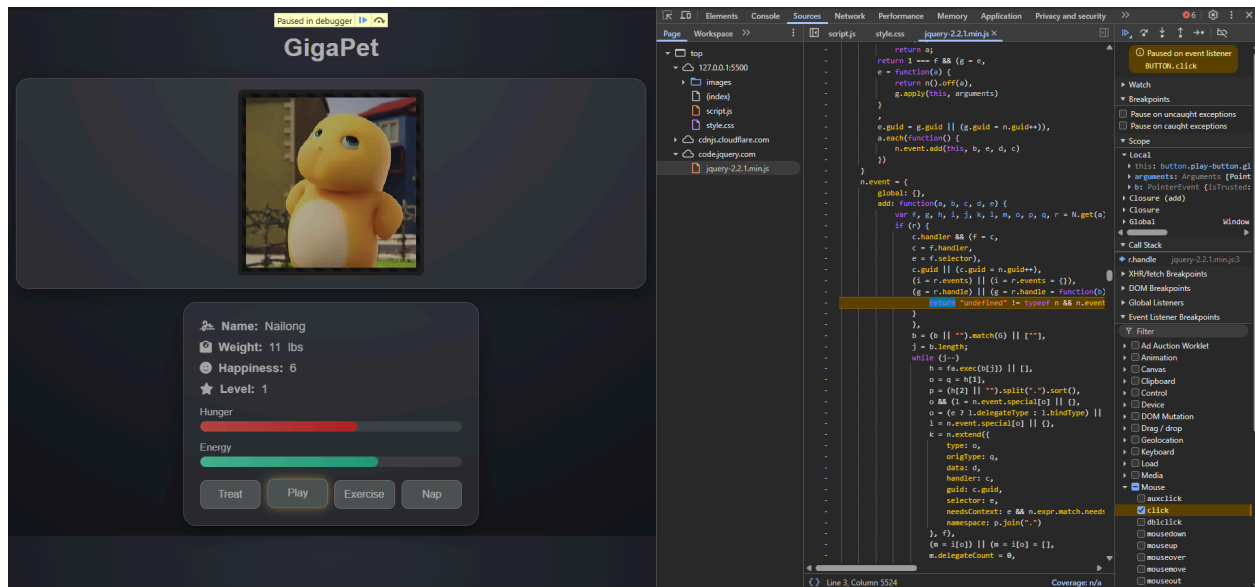
### 2) Sources Panel

- 1) Open DevTools and go to sources
- 2) The left is the file tree (open [script.js](#))
- 3) The middle is the code editor
- 4) The right are the breakpoints, scope, call stack, and watch



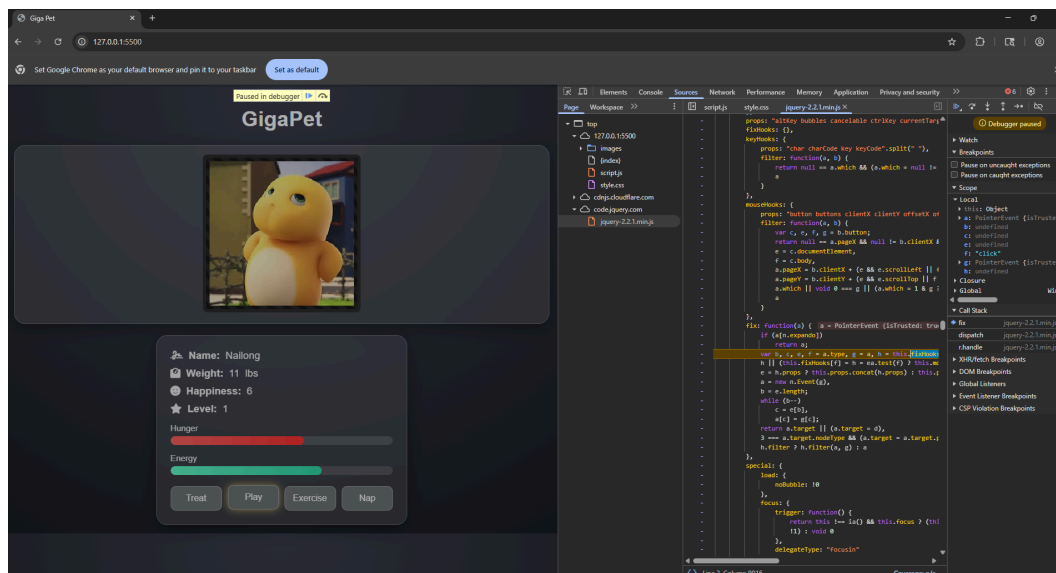
### 3) Pausing Code

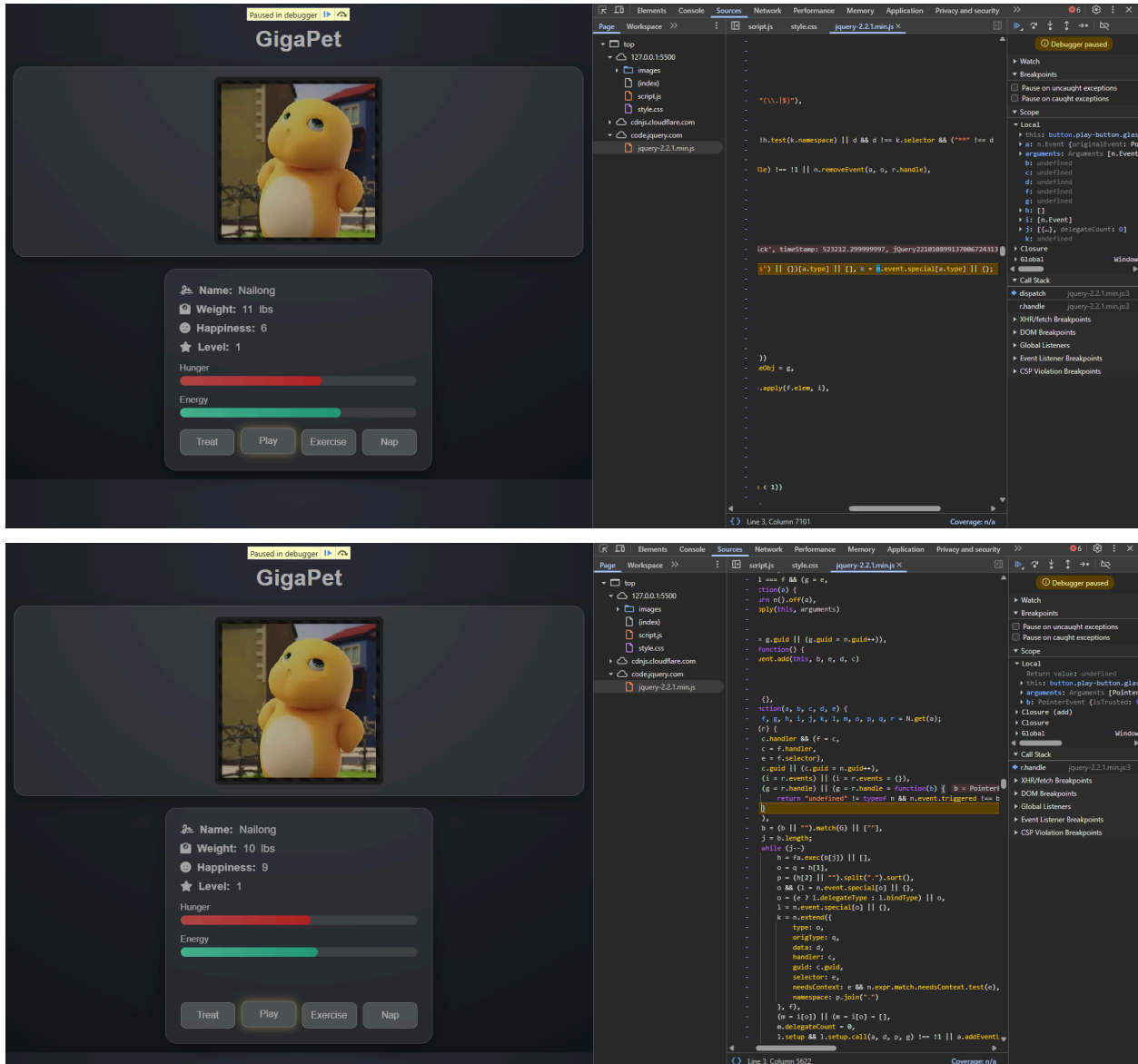
- 1) Go to the debugger
- 2) Then go to the event listener break points
- 3) Then go to mouse and check “click”
- 4) Clicking any button pauses inside its handler
- 5) For the line of code breakpoint click the left margin next to a line



### 4) Step Through the Code

- 1) While paused for stepping into, go inside the called function
- 2) For stepping over run the line without entering
- 3) For stepping out finish the function and return



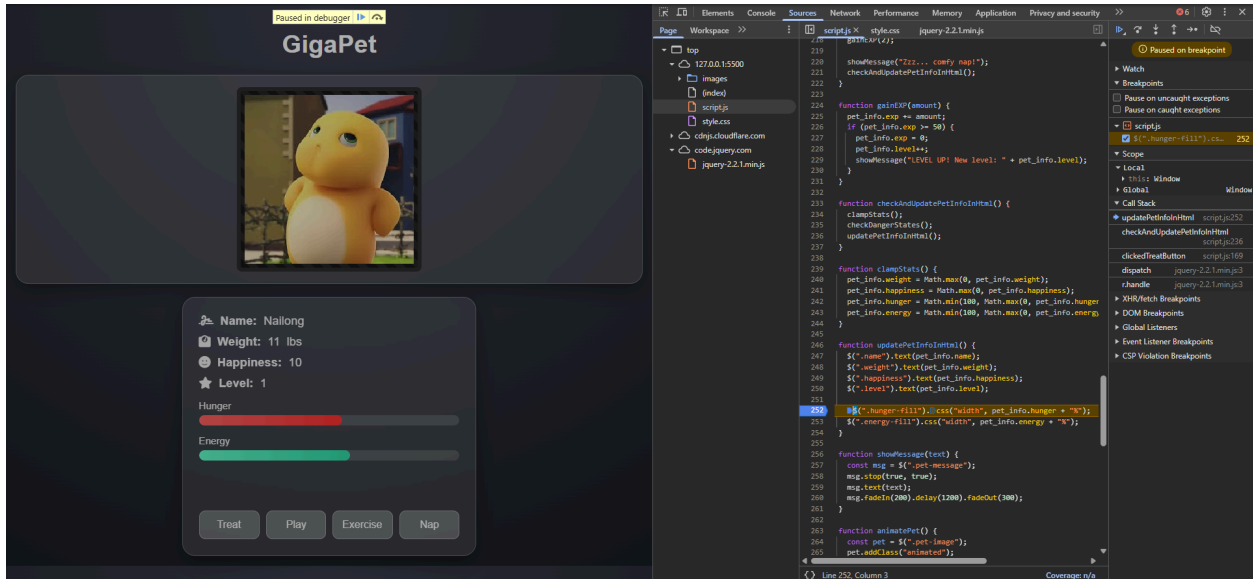


## 5) Breakpoint Example

- 1) Place a breakpoint on:

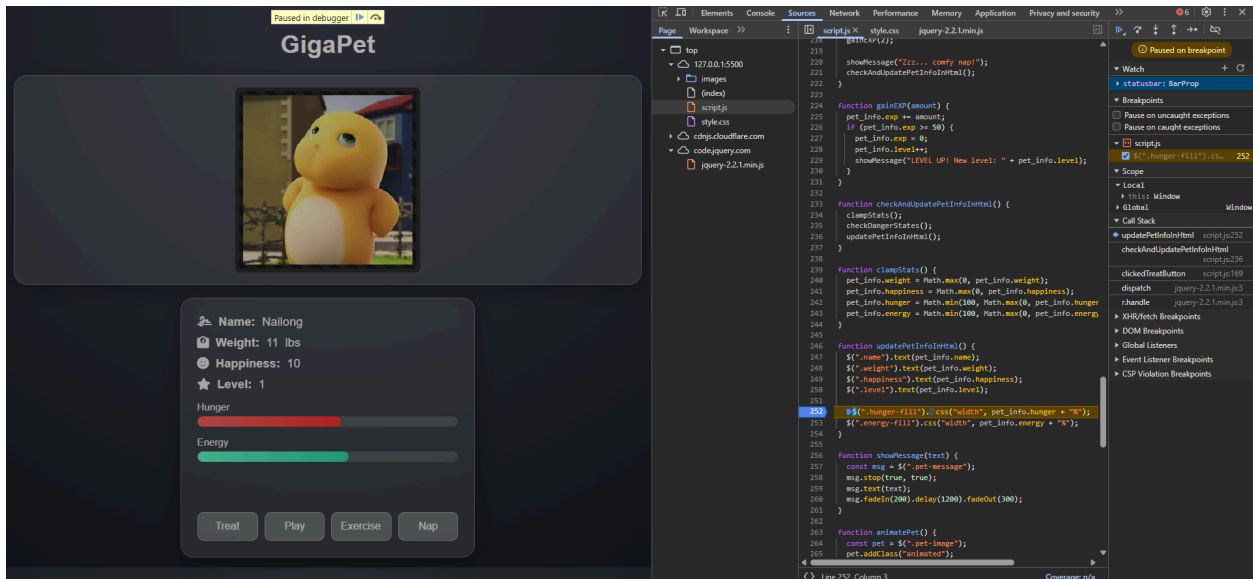
`$(“.hunger-fill”).css(“width, pet_info.hunger + “%”);`

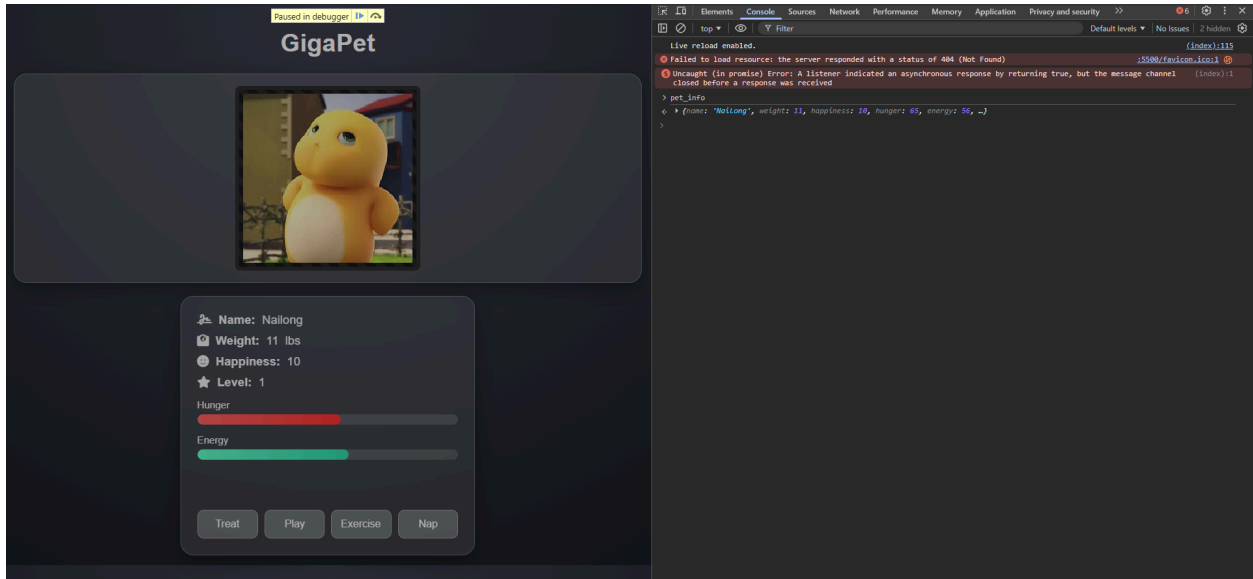
Which pauses before the bar updates so you can inspect the hunger and energy values.



## 6) Checking Variables

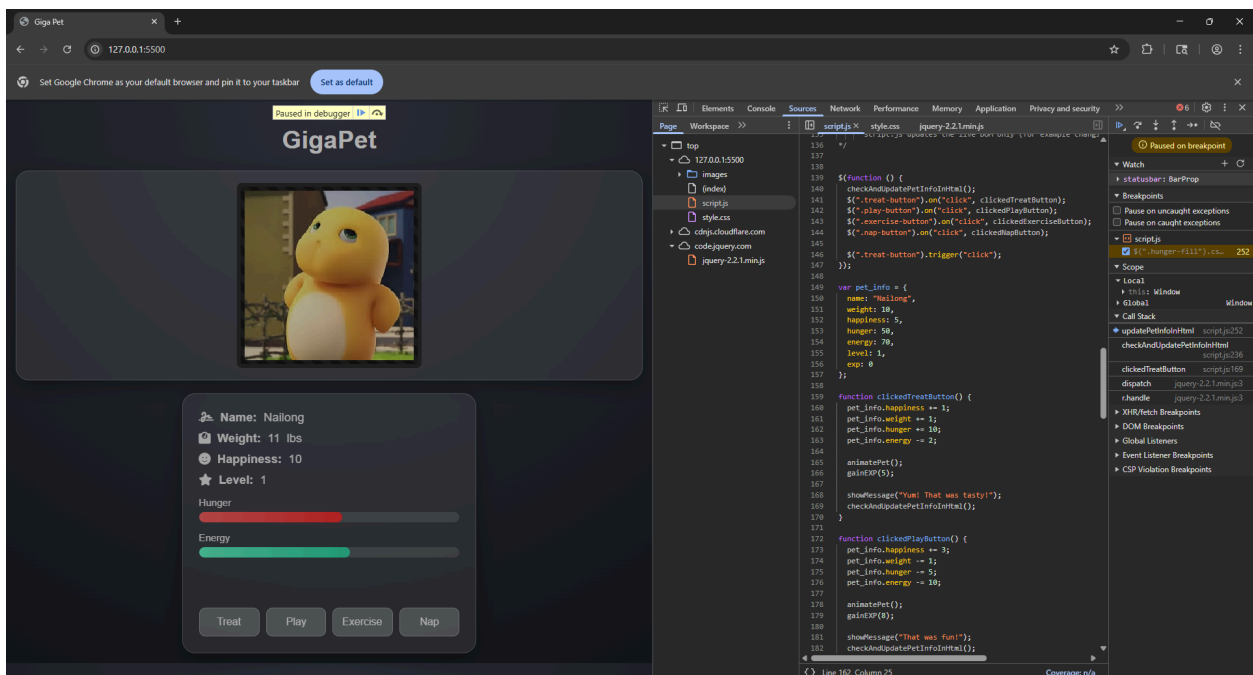
- 1) Scope panel shows all variables when paused
- 2) Watch expressions add values like pet\_info.hunger or pet\_info.level
- 3) In the console you can type pet\_info or pet\_info.hunger





## 7) Applying a Fix

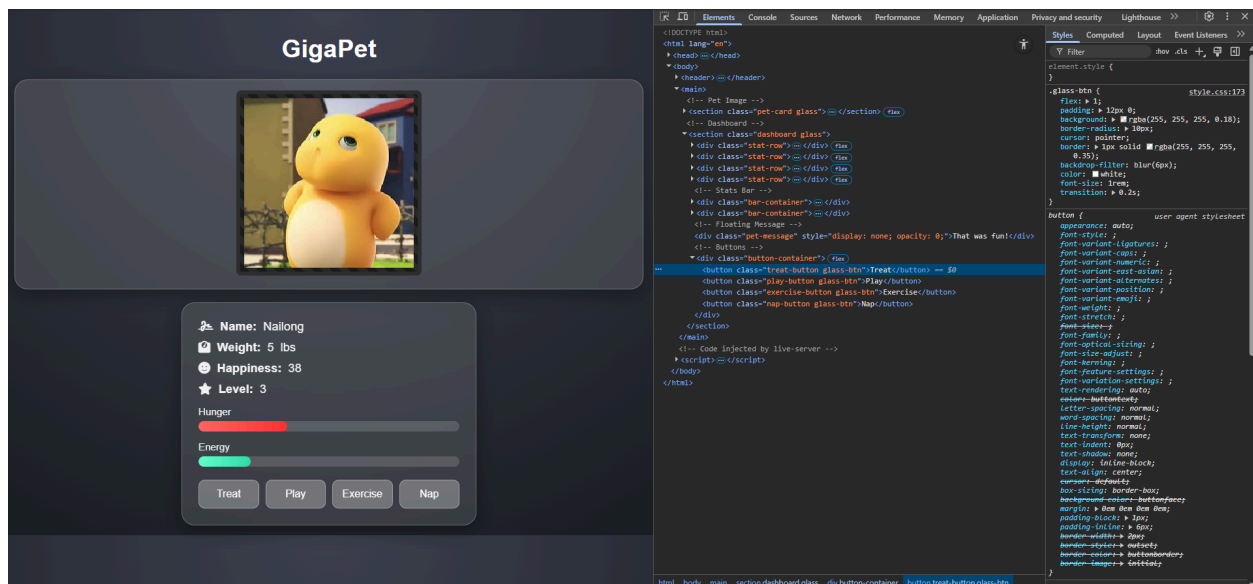
- 1) After identifying the wrong line ( $+=$  instead of  $-=$ ) edit it in the sources panel, save, execute, and test. Then apply actual fix to code.



# Part 2: DOM Tools

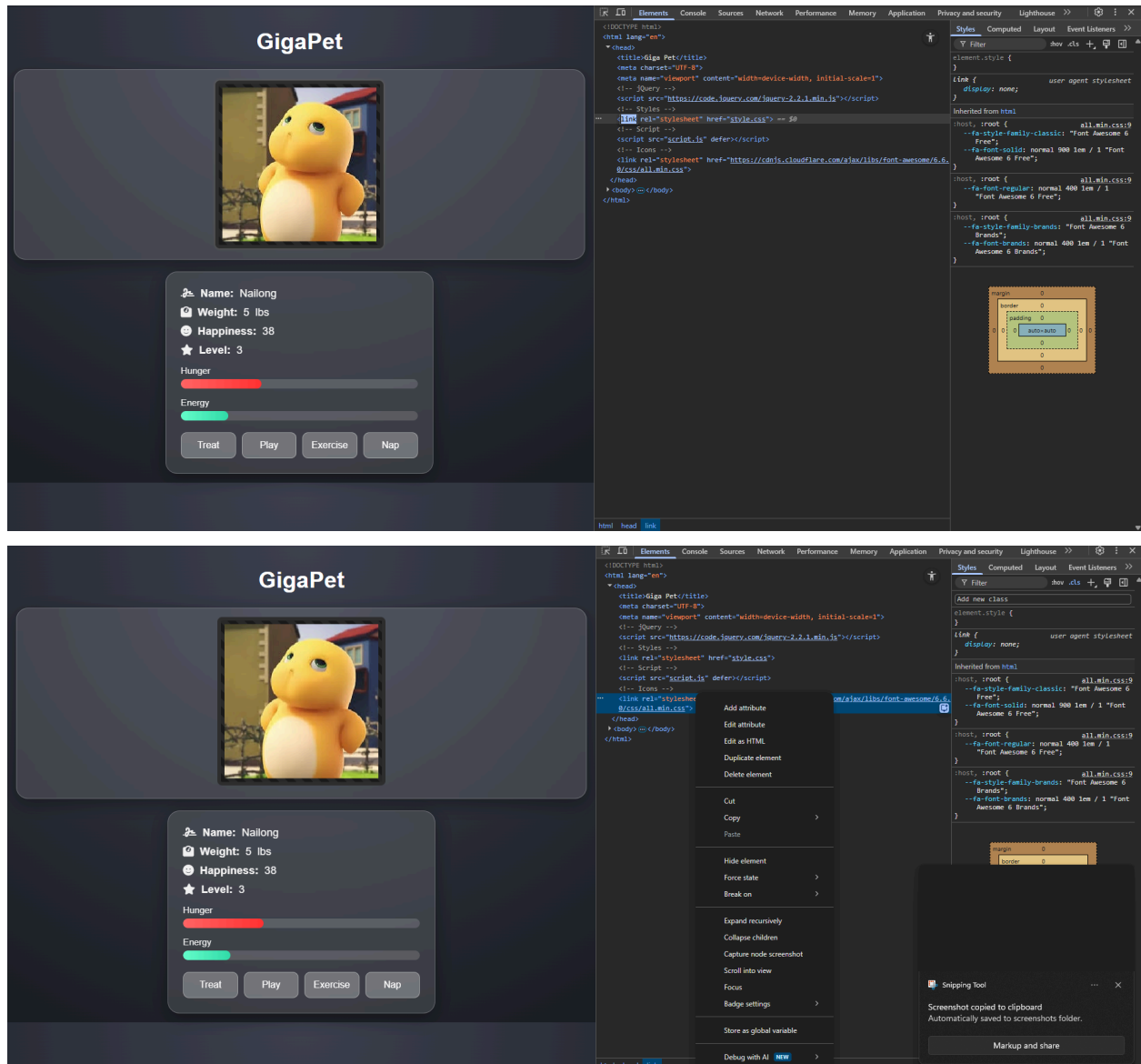
## 1) Viewing Elements

- 1) Inspect an element then use the arrow keys to navigate the DOM
- 2) Scroll into view jumps to the location
- 3) Rulers help visualize spacing
- 4) Ctrl + f to search for classes or text



## 2) Edit the DOM

- 1) Edit text by double clicking it
- 2) Edit attributes (like img or adding a title)
- 3) Change tag type by double clicking the name of the tag
- 4) Edit as HTML for structural changes
- 5) Duplicate Nodes to test layout ideas
- 6) Capture node screenshot
- 7) Drag nodes to reorder
- 8) Force states like :hover
- 9) Press H to hide nodes
- 10) Delete and undo with Delete with ctrl + z



### 3) Console Access to Elements

- 1) `$0` refers to the last inspected element in the DOM
- 2) Store elements as global variables (temp1 and temp2)
- 3) Copy JS path to generate a selector

### 4) Break on DOM Changes

- 1) Right click an element and go to break on
- 2) Then go to attribute modifications, subtree changes, or removal
- 3) It is useful for tracking where updates originate

## **5) HTML vs DOM**

- 1) Index.html never changes while the page runs.
- 2) JS updates the live DOM only
- 3) updatePetInfoInHtml modifies the DOM but not the actual HTML file