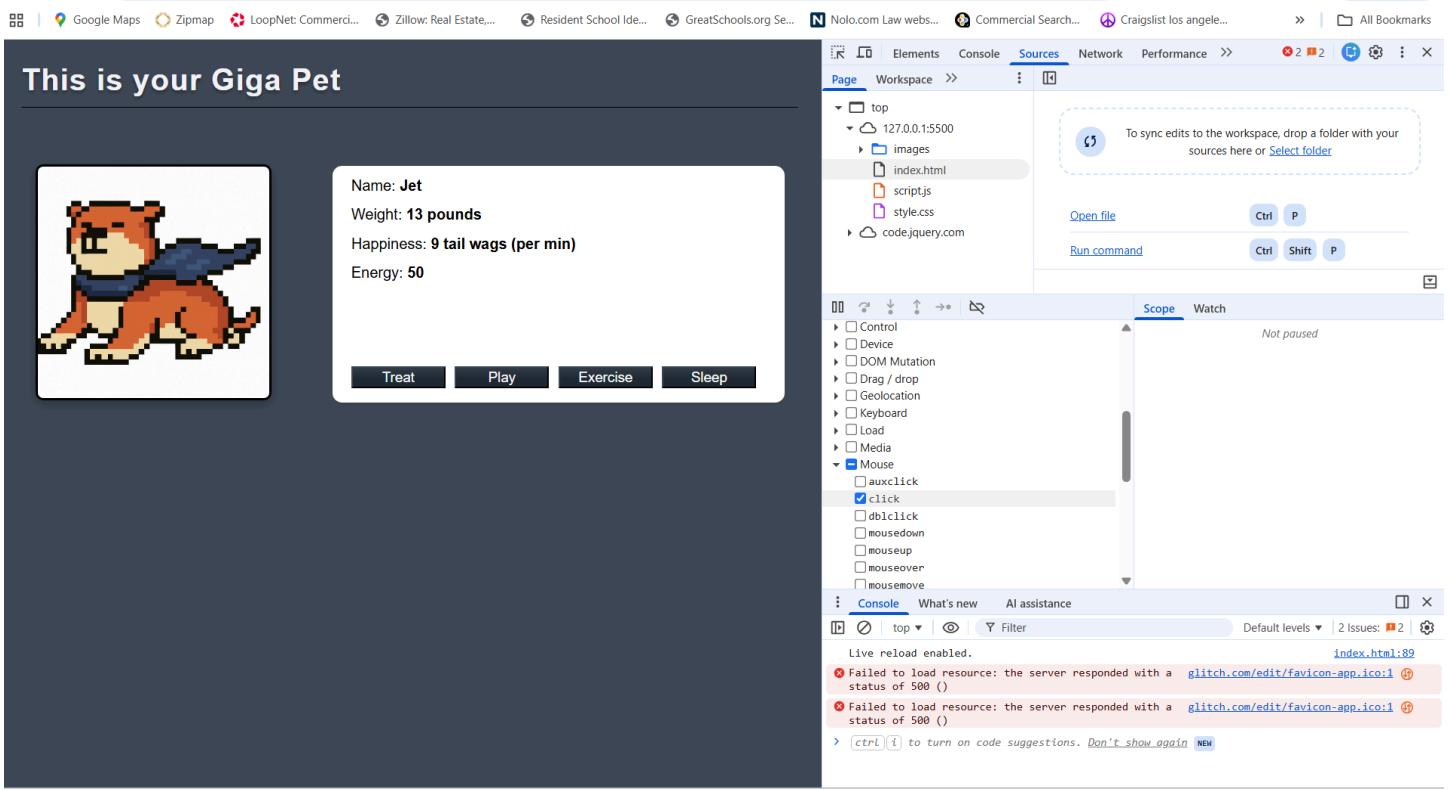


COMP 484 - HW10 Dev Tools

Spencer Levine

11/30/25



Event Listener Breakpoints > Mouse > Click

The screenshot shows a browser window with a pet simulation application titled "This is your Giga Pet". The application features a pixelated dog named "Jet" and displays its stats: Name: Jet, Weight: 13 pounds, Happiness: 9 tail wags (per min), and Energy: 50. Below the stats are four buttons: Treat, Play, Exercise, and Sleep. The browser's developer tools are open, specifically the Sources tab, which is paused on a line of code in the "jquery-2.2.1.min.js" file. The highlighted line of code is: `return "undefined" != typeof n && r`. The debugger also shows the call stack, local variables, and a console with two errors about failed resource loads.

Click and program pauses at line of code. Blue highlight indicates line of code paused on.

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. On the left, the file tree shows the current workspace with files like index.html, script.js, and style.css under '127.0.0.1:5550'. A file named 'jquery-2.2.1.min.js' is open in the main editor area. The code is heavily minified, with several lines highlighted in yellow. The bottom status bar indicates 'Line 3, Column 5524' and 'Coverage: n/a'. On the right, the 'Scope' tab is active, showing local variables: 'this: button.treat-button', 'arguments: Arguments [PointerEvent, callee: f]', 'b: PointerEvent {isTrusted: true, pointerId: 1}', and 'Closure (add)'. Below the scope, there's a toolbar with icons for resume, step, and breakpoints, and a dropdown menu with options like 'Resume script execution - F8 - Ctrl + \', 'XHR/fetch Breakpoints', 'DOM Breakpoints', and 'Global Listeners'.

```
c = t.handler,
e = f.selector),
c.guid || (c.guid = n.guid++),
(i = r.events) || (i = r.events = {}),
(g = r.handle) || (g = r.handle = function()
{
    return "undefined" != typeof n && r
}),
b = (b || "").match(G) || [],
j = b.length;
while (i--)
```

Resume script execution

atSchools.org Se... Nolo.com Law webs... Commercial Search... Craigslist los angele... > All Bookmarks

The screenshot shows the Chrome DevTools interface with the Sources tab selected. On the left, the file tree shows a local file structure under 'top' and a remote file 'jquery-2.2.1.min.js' from 'code.jquery.com'. The code editor displays the contents of 'jquery-2.2.1.min.js' with several lines highlighted in yellow. The status bar indicates 'Line 3, Column 9000' and 'Coverage: n/a'. Below the code editor is the Breakpoints sidebar, which lists breakpoints for 'fix', 'dispatch', and 'r.handle' in 'jquery-2.2.1.min.js:3'. The Scope panel on the right shows the current local variables: 'this', 'a', 'b', 'c', 'e', 'f', 'g', and 'h', all currently undefined.

```
a.which || void 0 === g || (a.which =  
a  
}  
},  
fix: function(a) { a = PointerEvent {isTrusted:  
if (a[n.expando])  
return a;  
var b, c, e, f = a.type, g = a, h = this.  
h || (this.fixHooks[f] = h = ea.test(f) ?  
e = h.props ? this.props.concat(h.props)  
a = new n.Event(a)  
}  
Line 3, Column 9000 Coverage: n/a
```

Scope Watch

- Local
 - this: Object
 - a: PointerEvent {isTrusted: true, pointerI
 - b: undefined
 - c: undefined
 - e: undefined
 - f: undefined
 - g: undefined
 - h: undefined

Step into next

Page Workspace >> : jquery-2.2.1.min.js X

top
127.0.0.1:5500
 images
 index.html
 script.js
 style.css
code.jquery.com
 jquery-2.2.1.min.js

```
n.event = {
    global: {},
    add: function(a, b, c, d, e) {
        var f, g, h, i, j, k, l, m, o, p, q, r = N;
        if (r) {
            c.handler && (f = c,
            c = f.handler,
            e = f.selector),
            c.guid || (c.guid = n.guid++),
            (i = r.events) || (i = r.events = {}),
            (g = r.handle) || (g = r.handle = funct
                return "undefined" != typeof n && r
            }
        },
        b = (b || "").match(G) || [""],
        j = b.length;
        while (j--)
            h = fa.exec(b[j]) || [],
            o = q = h[1],
            p = (h[2] || "").split(".").sort(),
            o && (l = n.event.special[o] || {}),
            o = (e ? l.delegateType : l.bindType),
            l = n.event.special[o] || {};
    }
},
```

Line 3, Column 5524 Coverage: n/a

Scope Watch

▶ r.handleClick Step over next function call - F10 - Ctrl + ' B

▶ XHR/fetch Breakpoints

▶ DOM Breakpoints

▼ Local

- ▶ this: button.treat-button
- ▶ arguments: Arguments [PointerEvent, callee: ,]
- ▶ b: PointerEvent {isTrusted: true, pointerId: , }

Step over next function call

The screenshot shows the Chrome DevTools Sources tab. The left sidebar lists resources: top, 127.0.0.1:5500 (images, index.html, script.js, style.css), and code.jquery.com (jquery-2.2.1.min.js). The main pane displays the code for jquery-2.2.1.min.js in VM379. A yellow highlight covers the following code block:

```
        rating: "good",
        delta: 0,
        entries: [],
        id: `v4-${Date.now()}-${Math.floor(8999999999999999)}`,
        navigationType: o
    }
}, o = (e, t, n) => {
try {
    if (PerformanceObserver.supportedEntryTypes.includes("performance"))
        const r = new PerformanceObserver(e =>
            Promise.resolve().then(() => {
                t(e.getEntries())
            })
        );
        return r.observe(Object.assign({
            type: e,
            buffered: !0
        }, n || {}));
    r
}
}
```

The Scope panel at the bottom shows variables:

- (anonymous) Step out of current function - Shift + F11 - Ctrl + Shift +;
- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints

Scope variables:

- this: undefined
- e: PerformanceObserverEntryList {}
- Closure (o)
- Closure
- Closure

Step out of current function

The screenshot shows the Chrome DevTools Sources tab with the file 'jquery-2.2.1.min.js' selected. The code editor displays the following snippet:

```
var c, e, f, g = b.button;
return null == a.pageX && null != b.clientX
e = c.documentElement,
f = c.body,
a.pageX = b.clientX + (e && e.scrollLeft || a.pageY = b.clientY + (e && e.scrollTop || f
a.which || void 0 === g || (a.which = 1 & g
a
}
},
fix: function(a) {
  a = PointerEvent {isTrusted: true}
  if (a[n.expando])
    return a;
  var b, c, e, f = a.type, g = a, h = this.fixHook
  h || (this.fixHooks[f] = h = ea.test(f) ? this.m
  e = h.props ? this.props.concat(h.props) : th
  a = new n.Event(g),
  b = e.length;
  while (b--)
    c = e[b],
    a[c] = g[c];
  return a.target || (a.target = d),
  3 === a.target.nodeType && (a.target = a.target.
}
```

The status bar at the bottom indicates "Line 3, Column 9000". The "Scope" tab in the bottom right shows local variables:

Scope	Watch
fix	jquery-2.2.1.min.js:3
dispatch	jquery-2.2.1.min.js:3
r.handle	jquery-2.2.1.min.js:3

Local variables:

- this: Object
- a: PointerEvent {isTrusted: true, pointerId: ...undefined}

Set a line breakpoint

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar lists file resources, including 'jquery-2.2.1.min.js' which is currently selected. The main pane displays the source code of this file, with a blue arrow pointing to a specific line. This line contains the code: 'e = h.props ? this.props.concat(h.props) : th'. A yellow highlight covers the entire line. The status bar at the bottom indicates 'Line 3, Column 9111'. The bottom right corner shows 'Coverage: n/a'. The bottom navigation bar includes icons for resume, step, and breakpoints, along with a dropdown menu.

Line 3, Column 9111

Coverage: n/a

Scope Watch

Local

- ▶ this: Object
- ▶ a: PointerEvent {isTrusted: true, pointerId: undefined}
- ▶ b: undefined
- ▶ c: undefined

Execution stops at breakpoint

The screenshot shows the 'Scope' tab selected in the DevTools sidebar. Under the 'Local' section, there are several variables listed with their types and values:

- ▶ this: Object
- ▶ a: PointerEvent {isTrusted: true, pointerId:
 b: undefined
 c: undefined
 e: undefined
 f: "click"}
- ▶ g: PointerEvent {isTrusted: true, pointerId:
 h: {props: Array(11), filter: f}}
- ▶ Closure
- ▶ Global

A 'Window' label is positioned to the right of the variable list. At the bottom of the panel, there are navigation arrows.

Scope section shows all local and global variables

Page Workspace >> :

jquery-2.2.1.min.js X VM379

```
var c, e, f, g = b.button;
return null == a.pageX && null != b.clientX
e = c.documentElement,
f = c.body,
a.pageX = b.clientX + (e && e.scrollLeft || 
a.pageY = b.clientY + (e && e.scrollTop || f
a.which || void 0 === g || (a.which = 1 & g
a
}
}, fix: function(a) { a = PointerEvent {isTrusted: true}
if (a[n.expando])
return a;
var b, c, e, f = a.type, g = a, h = this.fixHook
h || (this.fixHooks[f] = h = ea.test(f) ? this.m
e = h.Dprops ? this.props.Dconcat(h.props) : th
a = new n.Event(g),
b = e.length;
while (b--)
c = e[b],
a[c] = g[c];
return a.target || (a.target = d),
3 === a.target.nodeType && (a.target = a.target.

```

Line 3, Column 9111 Coverage: n/a

Scope Watch

+ C

typeof h: "object"

fix jquery-2.2.1.min.js:3

dispatch jquery-2.2.1.min.js:3

r.handle jquery-2.2.1.min.js:3

▶ XHR/fetch Breakpoints

▶ DOM Breakpoints

▶ Global Listeners

▼ Event Listener Breakpoints

Filter

Watch section I entered typeof(h) and returned “object”

This is your Giga Pet

Name: Jet
Weight: 19 pounds
Happiness: 19 tail wags (per min)
Energy: 90

Treat Play Exercise Sleep

```
h.props
(11) [ 'button', 'buttons', 'clientX', 'clientY', 'offsetX', 'offsetY', 'pageX', 'pageY', 'screenX', 'screenY', 'toElement' ]
```

In console window, h.props returned property values

```
h.props
(11) [ 'button', 'buttons', 'clientX', 'clientY', 'offsetX', 'offsetY', 'pageX', 'pageY', 'screenX', 'screenY', 'toElement' ]
h.props.push('testProperty')
(12) [ 'button', 'buttons', 'clientX', 'clientY', 'offsetX', 'offsetY', 'pageX', 'pageY', 'screenX', 'screenY', 'toElement', 'testProperty' ]
```

In the console I entered the code `h.props.push('testProperty')`. It added that string to the end of the array.

The screenshot shows a debugger interface with the following details:

- Code Area:** Displays a block of JavaScript code with several lines highlighted in blue, indicating they are currently being executed or have been recently run.
- Breakpoint Status:** A large blue arrow icon is positioned above the code, indicating a breakpoint has been triggered.
- Line and Column:** The cursor is at Line 3, Column 9111.
- Coverage:** Coverage information is listed as "n/a".
- Toolbar:** Includes standard debugger icons for step, resume, and stop.
- Breakpoints:** A sidebar on the left lists "XHR/fetch Breakpoints" and "DOM Breakpoints".
- Scope Tab:** The "Scope" tab is active, showing a tooltip for "Deactivate breakpoints - Ctrl + F8".
- Watch Tab:** The "Watch" tab is also present.

Activate/Deactivate breakpoints

This is your Giga Pet



Name: **Jet**
 Weight: **21 pounds**
 Happiness: **21 tail wags (per min)**
 Energy: **90**

button.treat-button 100 x 22.8

Treat Play Exercise Sleep

Elements Console Sources Network Performance Memory > Finish update

```
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body>
    <header> ... </header>
    <main>
      <div class="giga-wrapper"> <flex>
        <!-- Wraps or groups pet image and dashboard together for styling and layout positioning
        -->
        <section class="pet-image-container"> ... </section>
      <section class="dashboard"> <flex>
        <div> ... </div>
        <div> ... </div>
        <div> ... </div>
        <div> ... </div>
        <!-- 5.) Add a new behavior that correlates with the new button you added.
        Yo
        u can add it below the happiness text in the html -->
        <div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum! Thanks for the treat!
        </div>
        <!-- 6.) adds a message area to step 6 -->
        <div class="button-container">
          <button class="treat-button"> Treat </button> -- $0
          <button class="play-button"> Play </button>
        </div>
      </section>
    </main>
  </body>
</html>
```

Styles Computed Layout Event Listener DOM Breakpoints Properties Accessibility

element.style { style.css:141 }

.button-container button:hover { background-color: #2d3a4d; transition: > background-color 0.2s ease-in-out; } style.css:92

.button-container button { width: 100px; margin-right: 10px; text-align: center; display: inline-block; float: left; font-size: 15px; background-color: #1e2835; border-color: > #404853; cursor: pointer; }

Inspect Treat button

Elements Console Sources Network Performance Memory >  Yo

```

<!-- 5.) Add a new behavior that correlates with the new button you added.

u can add it below the happiness text in the html -->
<div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum! Thanks for the treat!
</div>
<!-- 6.) adds a message area to step 6 -->
<div class="button-container">
  <button class="treat-button"> Treat </button>
  <button class="play-button"> Play </button>
  <button class="exercise-button"> Exercise </button>
...
  <button class="sleep-button"> == $0
    <!-- 4.) added the Sleep button -->
    " Sleep "
  </button>
</div>
<!-- EXTRA FEATURE: Sound effect for your pet extra feature -->
<!-->
<audio id="pet-sound" src="sound/dog_bark.mp3" preload="auto"></audio>
</section>
</div>
</main>
<!-- Your web app is https so your scripts need to be too -->

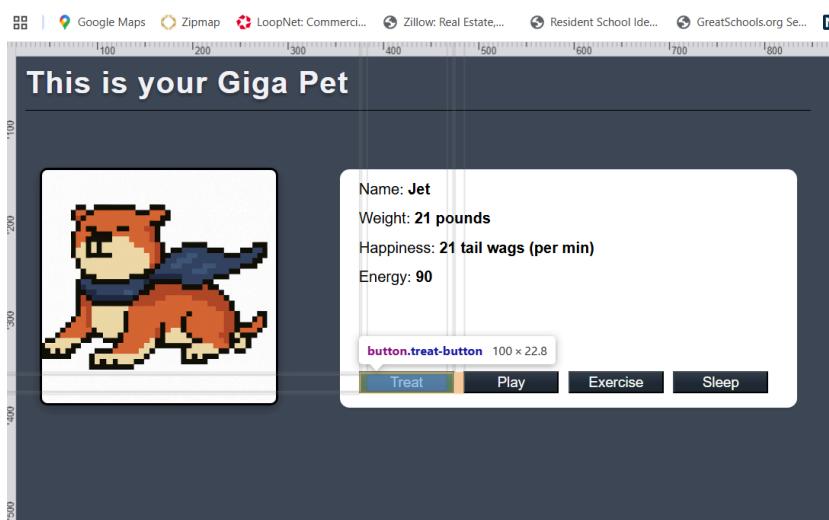
```

html body main div.giga-wrapper section.dashboard div.button-container button.sleep-button

Right arrow key on keyboard the button node expands.

Google Maps Zipmap LoopNet: Commercial... Zillow: Real Estate... Resident School Ide... GreatSchools.org Se... Nolo.com Law webs... Commercial Search... Craigslist los angele... All Bookmark

This is your Giga Pet



Name: Jet
Weight: 21 pounds
Happiness: 21 tail wags (per min)
Energy: 90

button.treat-button 100 x 22.8

Treat Play Exercise Sleep

Elements Console Sources Network Performance Memory >  Yo

```

<span class="happiness">21</span>
</div>
<!-- 5.) Add a new behavior that correlates with the new button you added.

u can add it below the happiness text in the html -->
<div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum! Thanks for the treat!
</div>
<!-- 6.) adds a message area to step 6 -->
<div class="button-container">
  <button class="treat-button"> Treat </button>
  <button class="play-button"> Play </button>
  <button class="exercise-button"> Exercise </button>
  <button class="sleep-button"> == $0
    <!-- 4.) added the Sleep button -->
    " Sleep "
  </button>
</div>
<!-- EXTRA FEATURE: Sound effect for your pet extra feature -->
<!-->
<audio id="pet-sound" src="sound/dog_bark.mp3" preload="auto"></audio>

```

html body main div.giga-wrapper section.dashboard div.button-container button.treat-button

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Show rulers on hover set.

This is your Giga Pet

Name: Jet

Weight: 21 pounds

Happiness: 21 tail wags (per min)

Energy: 90

Treat Play Exercise Sleep

```
<div class="button-container">
  <button class="treat-button"> Treat </button>
  <button class="play-button"> Play </button>
  <button class="exercise-button"> Exercise </button> == $0
  ...
  <button class="sleep-button">
    <!-- 4. --> added the Sleep button -->
    " Sleep "
  </button>
</div>
<!--
<!-- EXTRA FEATURE: Sound effect for your pet extra feature -->
<!--
<audio id="pet-sound" src="sound/dog_bark.mp3" preload="auto"></audio>
</section>
</div>
</main>
<!-- Your web-app is https, so your scripts need to be too -->
<script src="https://code.jquery.com/jquery-2.2.1.min.js" integrity="sha256-gVQAFzTH6trSrAwohHIp09Xc9G0x23fe6kem+000" crossorigin="anonymous"></script>
<!-- Code injected by live-server -->
<script>...</script>
</body>
</html>
```

exercise

Searched elements, ctrl-f find the string “exercise” in the HTML.

```
        " tail wags (per min)"  
    </strong>  
    </div>  
    > <div>...</div>  
    <!-- 5.) Add a new behavior that correlates with the new button you added.  
  
    u can add it below the happiness text in the html -->  
    <div id="pet-msg" aria-live="polite">  
    </div>  
    <!-- 6.) adds a message area -->  
    <div class="button-container">  
        ...  
        <button class="treat-button">  
        <button class="play-button">  
        <button class="exercise-button">  
        <button class="sleep-button">  
            <!-- 4.) added the Sleep -->  
            " Sleep "  
        </button>  
    </div>  
    <!-- EXTRAS -->  
    <!-- EXTRA FEATURE: Sound effects -->  
    <!-- EXTRAS -->
```

Add/Edit attribute

Elements Console Sources Network Performance Memory >

```

21
" tail wags (per min)"

```


</div>

► <div> ... </div>

<!-- 5.) Add a new behavior that correlates with the new button you added.

Yc

can add it below the happiness text in the html -->

```

<div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum! Thanks for the treat!
```

</div>

<!-- 6.) adds a message area to step 6 -->

▼ <div class="button-container">

```

<button class="treat-button">
    Treat
</button>

<button class="play-button"> Play </button>
<button class="exercise-button"> Exercise </button>
▼ <button class="sleep-button">
    <!-- 4.) added the Sleep button -->
    " Sleep "
</button>

```

Edit as HTML

← → C 127.0.0.1:5500/index.html

Finish update

Google Maps Zipmap LoopNet: Commercial... Zillow: Real Estate... Resident School Ide... GreatSchools.org Se... Nolo.com Law webs... Commercial Search... Craigslist los angele... All Bookmark

This is your Giga Pet



Name: Jet
Weight: 21 pounds
tail wags (per min)
Energy: 90

Treat Play Exercise Sleep

Elements Console Sources Network Performance Memory >

... <div> ... </div>

- Add attribute
- Edit as HTML
- Duplicate element
- Delete element
- Cut
- Copy
- Paste
- Hide element
- Force state
- Break on
- Find by string
- Expand recursively
- Collapse children
- Capture node screenshot

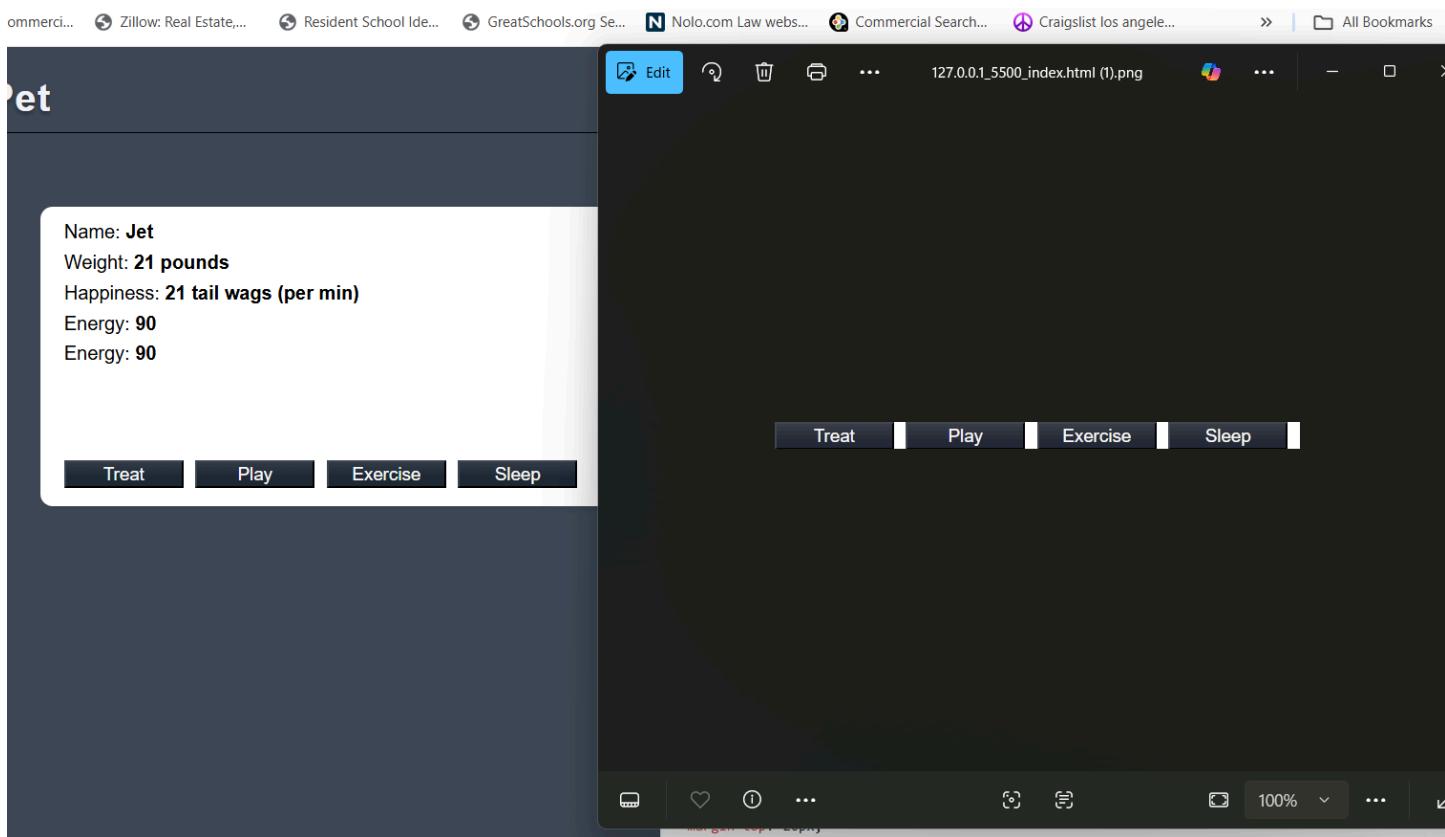
Breakpoints Properties Accessibility

Select Duplicate element

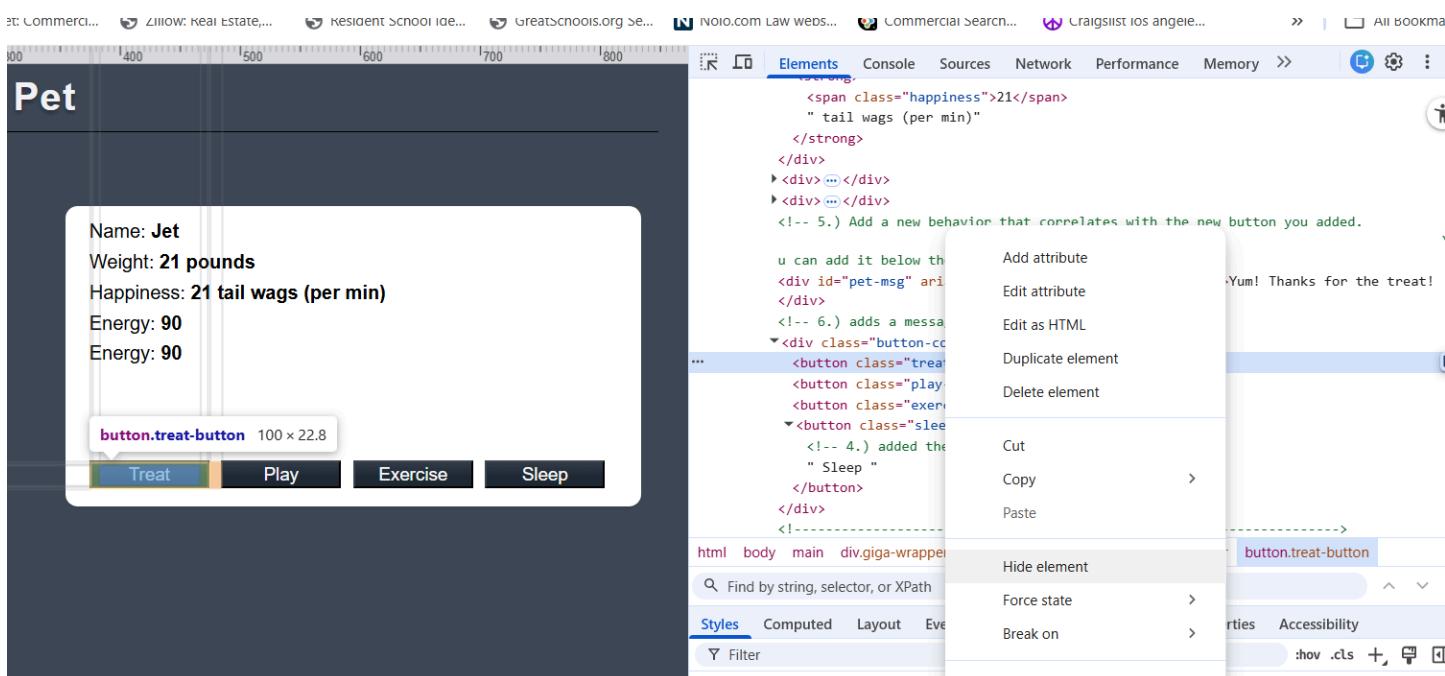
The screenshot shows a web application for managing a pet. The title bar says "a Pet". The main content area displays a pet's information: Name: Jet, Weight: 21 pounds, tail wags (per min), Energy: 90. Below this are four buttons: Treat, Play, Exercise, and Sleep. The "tail wags (per min)" text has a tooltip pointing to it. The "Energy: 90" text is highlighted with a blue background. The "Elements" tab in the developer tools shows the HTML structure, including a div for the pet's stats and a list of buttons.

```
<span c: " tail \n</strong>\n</div>\n...><div>...</d>\n><div>...</d>\n<!-- 5.) Ad\n\nu can add i\n<div id="pe\n</div>\n<!-- 6.) ad\n<div class=\n<button c\n<button c\n<button c\n<button c\n<!-- 4.)\n" Sleep\n</button>\n</div>\n<!-- ----->\nhtml body main div!
```

Element has been duplicated.



Capture node screenshot



Hide element

The screenshot shows a pet dashboard with the following details:

- Name: Jet
- Weight: 21 pounds
- Happiness: 21 tail wags (per min)
- Energy: 90
- Energy: 90

Below the details are three buttons: Play, Exercise, and Sleep.

In the browser's developer tools (Elements tab), the DOM structure is visible. A button with the class "play-button" is present in the DOM, but it is visually hidden or disabled. The "Exercise" button is highlighted with a blue border in the browser's UI.

```

<span class="happiness">21</span>
    " tail wags (per min)"
</strong>
</div>
> <div>...</div>
> <div>...</div>
<!-- 5.) Add a new behavior that correlates with the new button you add -->
    u can add it below the happiness text in the html -->
<div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum!
</div>
<!-- 6.) adds a message area to step 6 -->
<div class="button-container">
    <button class="treat-button __web-inspector-hide-shortcut__">
        <button class="play-button"> Play </button>
        <button class="exercise-button"> Exercise </button>
        <button class="sleep-button">
            <!-- 4.) added the Sleep button -->
            " Sleep "
        </button>
    </div>
<!-->
<!-- EXTRA FEATURE: Sound effect for your pet extra feature -->

```

Element is hidden in DOM

The screenshot shows a pet dashboard with the following details:

- Name: Jet
- Weight: 21 pounds
- Happiness: 21 tail wags (per min)
- Energy: 90
- Energy: 90

Below the details are three buttons: Play, Exercise, and Sleep. The "Exercise" button is highlighted with a blue border in the browser's UI.

In the browser's developer tools (Elements tab), the DOM structure is visible. The "play-button" button is now clearly visible and functional. The "Exercise" button is highlighted with a blue border in the browser's UI.

```

<span class="happiness">21</span>
    " tail wags (per min)"
</strong>
</div>
> <div>...</div>
> <div>...</div>
<!-- 5.) Add a new behavior that correlates with the new button you add -->
    u can add it below the happiness text in the html -->
<div id="pet-msg" aria-live="polite" style="opacity: 0;">Yum! Thanks for
</div>
<!-- 6.) adds a message area to step 6 -->
<div class="button-container">
    <button class="play-button"> Play </button> == $0
    <button class="exercise-button"> Exercise </button>
    <button class="sleep-button">
        <!-- 4.) added the Sleep button -->
        " Sleep "
    </button>
</div>
<!-->
<!-- EXTRA FEATURE: Sound effect for your pet extra feature -->

```

Delete element

Elements Console Sources Network Performance Memory > 1 Default levels | No Issues

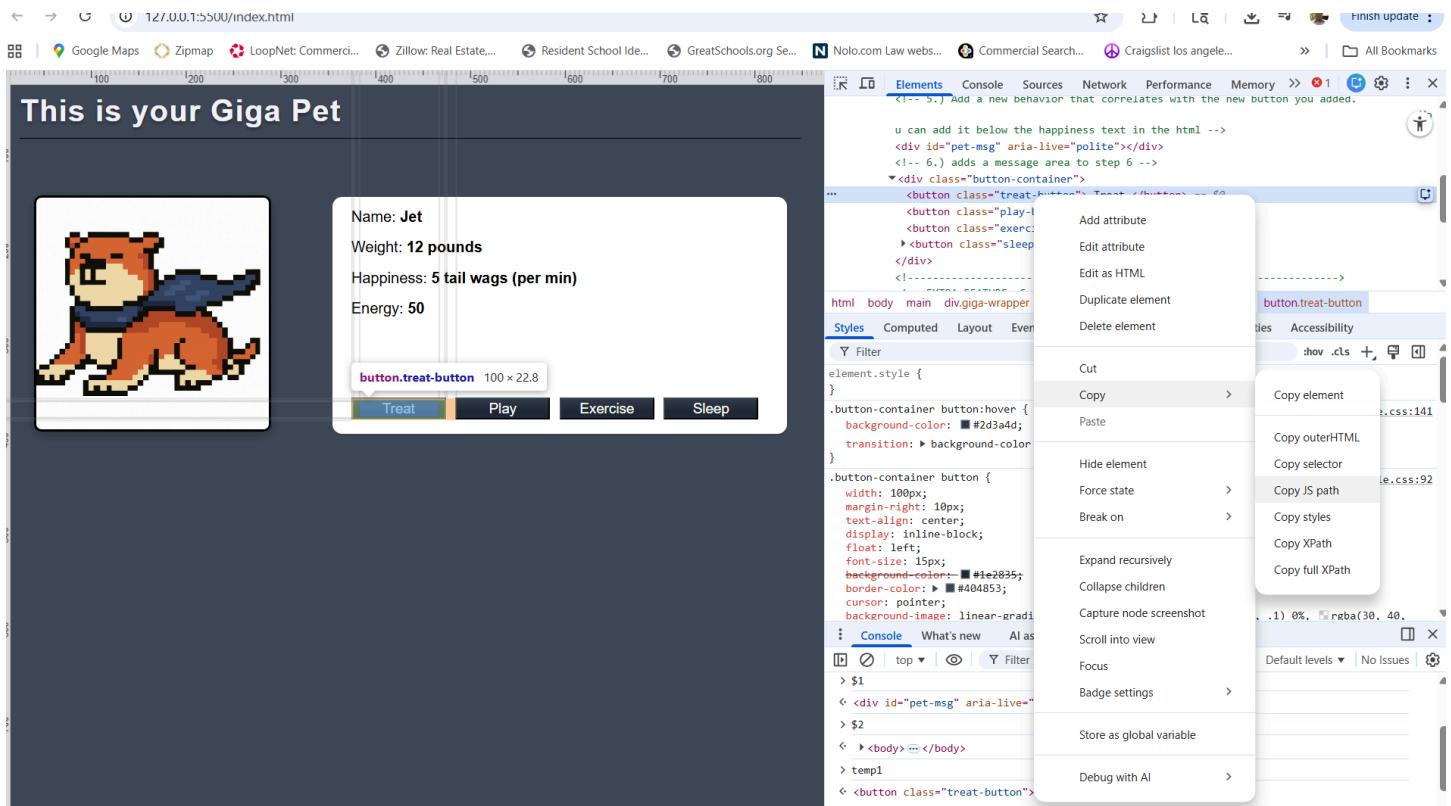
```
GET https://glitch.com/edit/favicon-app.ico 500 (Internal Server Error) favicon-app.ico:1
```

```
> $0
< button class="treat-button"> Treat </button>
> $1
< div id="pet-msg" aria-live="polite"></div>
> $2
< > <body> ...</body>
```

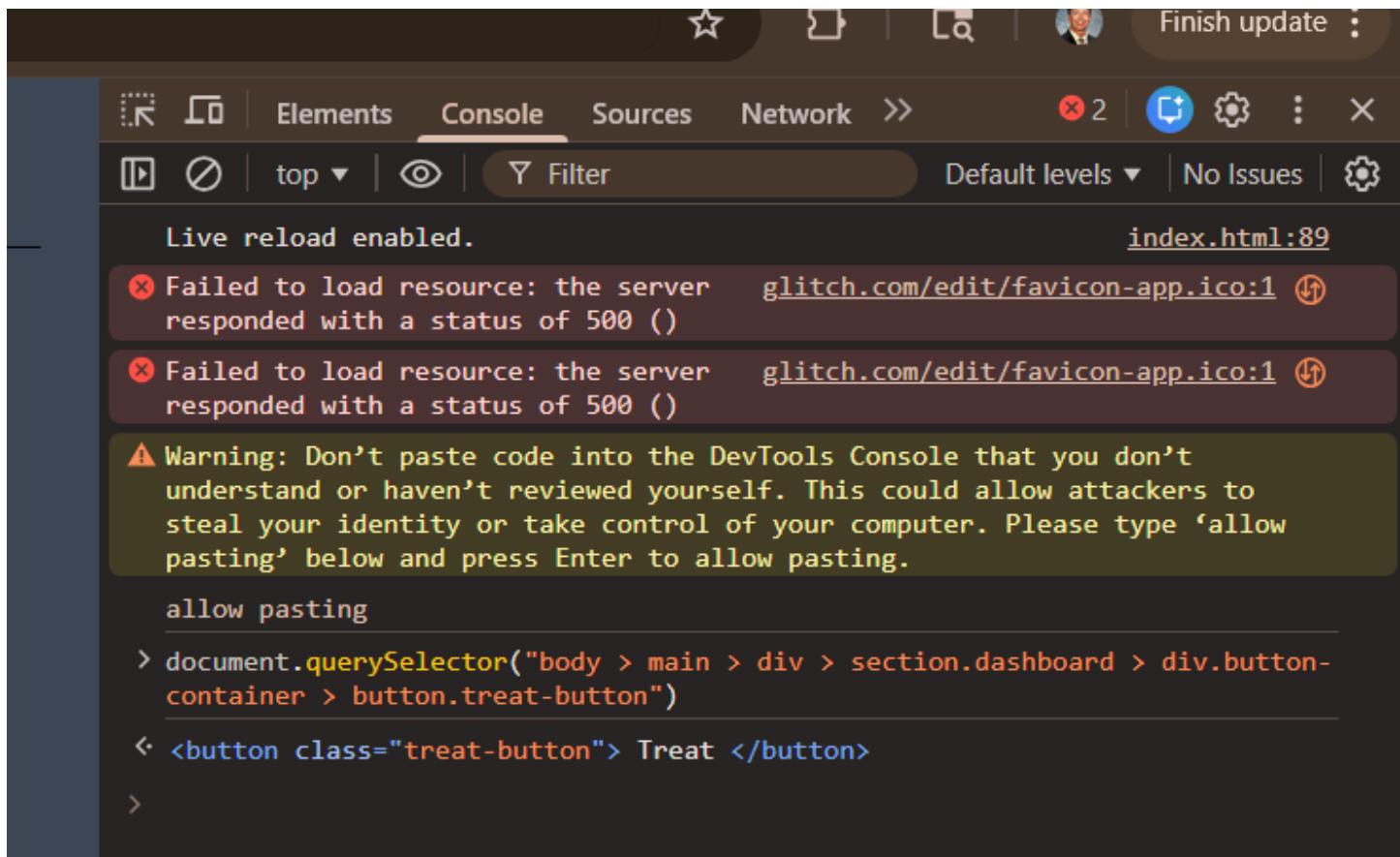
In console \$0 shows currently selected node

Elements Console Sources Network Performance Memory > 1

Store as global variable



Copy > Copy JS path



Paste JS path into console.

HTML versus the DOM

This section quickly explains the difference between HTML and the DOM.

When you use a web browser to request a page like `https://example.com`, the server returns an HTML document:

```
<!doctype html>
<html>
  <head>
    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <p>This is a hypertext document on the World Wide Web.</p>
    <script src="/script.js" async></script>
  </body>
</html>
```

The browser parses the HTML and creates a tree of objects:

```
html
  head
    title
  body
    h1
    p
    script
```