Title: Chrome DevTools Implementations – Project 2

Author: Eli Cubillo

Course: COMP 484 – Project 2 DevTools Extension

Section 1. JavaScript Debugging Implementation

A debugging demo was added to the bottom of the existing Project 2 interface. The calculator intentionally produced incorrect output because the values retrieved from the DOM were strings, causing concatenation rather than numeric addition.

Screenshot 1: Initial Bug Reproduction

The inputs “5” and “1” produced “5 + 1 = 51,” confirming the string-concatenation error.

Screenshot 2: Breakpoint Placement

A breakpoint was set on the line “const sum = n1 + n2;” inside script.js using the Chrome DevTools Sources panel. Execution paused upon pressing the Add Numbers button.

Screenshot 3: Scope Inspection

While paused on the breakpoint, the Scope panel reported:

n1 = "5"

n2 = "1"

indicating both values were string-typed.

Screenshot 4: Corrected Output After Fix

The code was corrected by converting inputs to numbers using the Number(...) constructor. The calculator then produced the correct output “5 + 1 = 6.”

Section 2. DOM Inspection and Manipulation Implementation

A DOM demonstration section was added to the page using a static list of players. All DOM manipulations were performed through the Chrome DevTools Elements panel.

Screenshot 5: Node Inspection

Right-click Inspect on “Connor Bedard” correctly highlighted the associated element in the DOM tree.

Screenshot 6: Live Text Modification

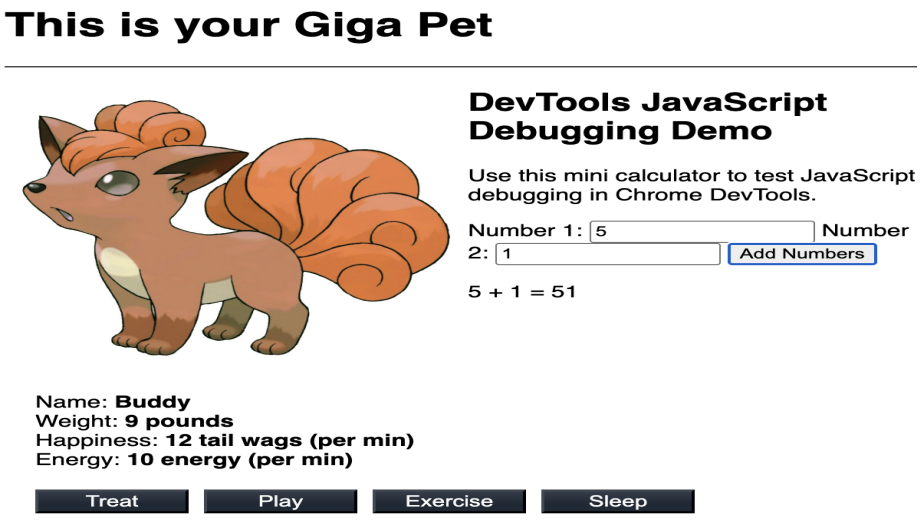
The text content inside the was edited directly inside Elements, changing “Connor Bedard” to “Eli Cubillo.” The live page reflected the modification immediately.

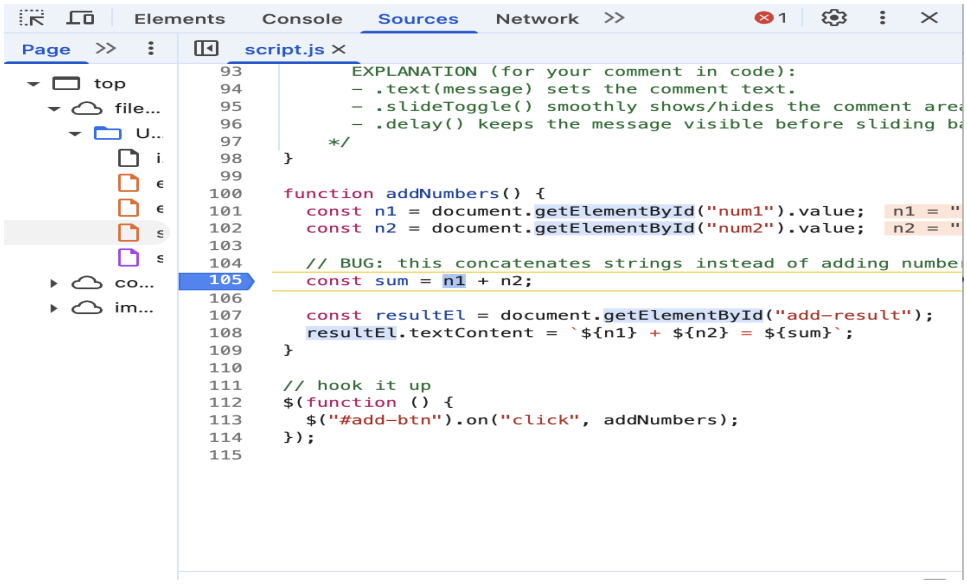
Screenshot 7: Node Insertion

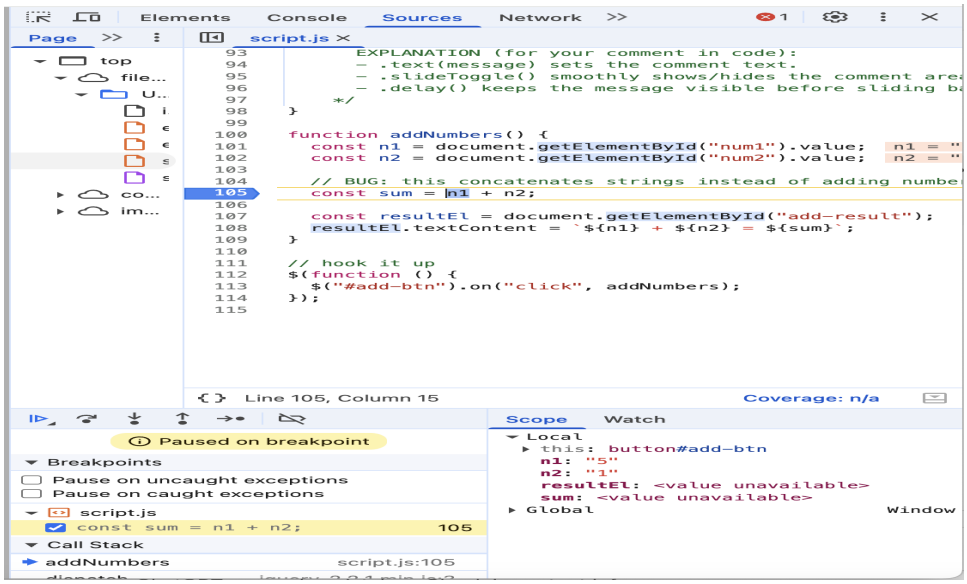
Using Edit as HTML on the parent , a new Wayne Gretzky element was inserted. The additional list item appeared instantly on the rendered page.

Summary

The JavaScript debugging workflow demonstrated the use of breakpoints, variable inspection, and correction of logic errors following the Chrome DevTools JavaScript tutorial. The DOM workflow demonstrated inspection, live editing, and structural modification of HTML nodes following the Chrome DevTools DOM documentation.




Elements
Console
Sources
Network
Page
script.js
top
file...
co...
im...
93 EXPLANATION (for your comment in code):
94 - .text(message) sets the comment text.
95 - .slideToggle() smoothly shows/hides the comment area
96 - .delay() keeps the message visible before sliding ba
97 */
98 }
99
100 function addNumbers() {
101 const n1 = document.getElementById("num1").value; n1 = "
102 const n2 = document.getElementById("num2").value; n2 = "
103
104 // BUG: this concatenates strings instead of adding numbe
105 const sum = n1 + n2;
106
107 const resultEl = document.getElementById("add-result");
108 resultEl.textContent = `${n1} + ${n2} = ${sum}`;
109 }
110
111 // hook it up
112 $(function () {
113 $("#add-btn").on("click", addNumbers);
114 });
115


Elements
Console
Sources
Network
Page
script.js
top
file...
co...
im...
93 EXPLANATION (for your comment in code):
94 - .text(message) sets the comment text.
95 - .slideToggle() smoothly shows/hides the comment area
96 - .delay() keeps the message visible before sliding ba
97 */
98 }
99
100 function addNumbers() {
101 const n1 = document.getElementById("num1").value; n1 = "
102 const n2 = document.getElementById("num2").value; n2 = "
103
104 // BUG: this concatenates strings instead of adding numbe
105 const sum = n1 + n2;
106
107 const resultEl = document.getElementById("add-result");
108 resultEl.textContent = `${n1} + ${n2} = ${sum}`;
109 }
110
111 // hook it up
112 $(function () {
113 $("#add-btn").on("click", addNumbers);
114 });
115
{ } Line 105, Column 15
Coverage: n/a
Paused on breakpoint
Breakpoints
Pause on uncaught exceptions
Pause on caught exceptions
script.js
const sum = n1 + n2; 105
Call Stack
addNumbers script.js:105
Scope
Watch
Local
this: button#add-btn
n1: "5"
n2: "1"
resultEl: <value unavailable>
sum: <value unavailable>
Global Window

This is your Giga Pet

DevTools JavaScript Debugging Demo

Use this mini calculator to test JavaScript debugging in Chrome DevTools.

Number 1: 5
Number 2: 1
Add Numbers

5 + 1 = 6

Name: **Buddy**
Weight: **10 pounds**
Happiness: **10 tail wags (per min)**
Energy: **10 energy (per min)**

Treat | Play | Exercise | Sleep

This is your Giga Pet

DevTools JavaScript Debugging Demo

Use this mini calculator to test JavaScript debugging in Chrome DevTools.

Number 1:
Number 2:
Add Numbers

Result will appear here.

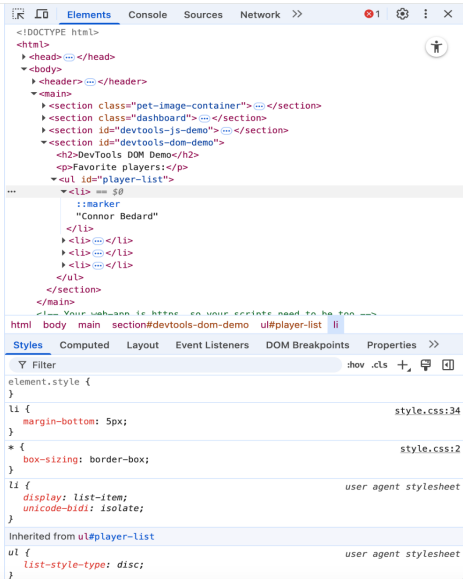
Name: **Buddy**
Weight: **10 pounds**
Happiness: **10 tail wags (per min)**
Energy: **10 energy (per min)**

Treat | Play | Exercise | Sleep

DevTools DOM Demo

Favorite players:

- Connor Bedard
- Leon Draisaitl
- Nathan MacKinnon
- Cale Makar



This is your Giga Pet

DevTools JavaScript Debugging Demo

Use this mini calculator to test JavaScript debugging in Chrome DevTools.

Number 1:
Number 2:
Add Numbers

Result will appear here.

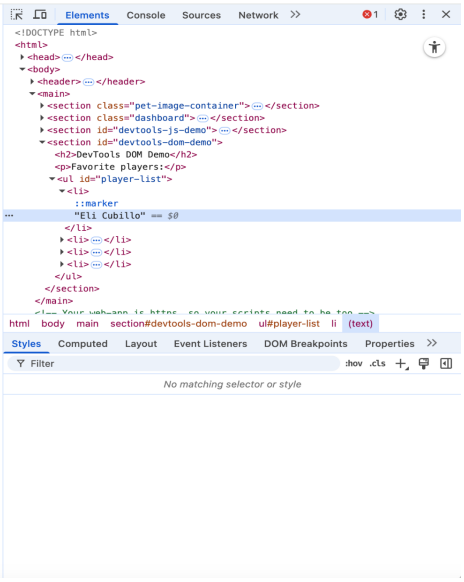
Name: **Buddy**
Weight: **10 pounds**
Happiness: **10 tail wags (per min)**
Energy: **10 energy (per min)**

Treat | Play | Exercise | Sleep

DevTools DOM Demo

Favorite players:

- Eli Cubillo
- Leon Draisaitl
- Nathan MacKinnon
- Cale Makar



This is your Giga Pet

DevTools JavaScript Debugging Demo

Use this mini calculator to test JavaScript debugging in Chrome DevTools.

Number 1:
Number 2:
Add Numbers

Result will appear here.

Name: **Buddy**
Weight: **10 pounds**
Happiness: **10 tail wags (per min)**
Energy: **10 energy (per min)**

Treat | Play | Exercise | Sleep

DevTools DOM Demo

Favorite players:

- Eli Cubillo
- Leon Draisaitl
- Nathan MacKinnon
- Cale Makar
- Wayne Gretzky