



AttentionTag

Your AI Engagement Assistant

Session Report

# JavaScript Fundamentals

Browser Storage · DOM Strategies · OOP & Prototypes

---

<b>Class Size</b>	41 students
<b>Session Duration</b>	~100 minutes
<b>Report Generated by</b>	AttentionTag

---

# 1. ATTENDANCE & PARTICIPATION

This was a ~100 minute session with 41 students joining at some point across 39 unique participants. Attendance was spread widely — from students who stayed the full session to those who dropped in briefly.

<b>13</b> Full session (60m+)	<b>12</b> Partial (30–60 min)	<b>8</b> Brief (15–30 min)	<b>8</b> Very brief (<15 min)
----------------------------------	----------------------------------	-------------------------------	----------------------------------

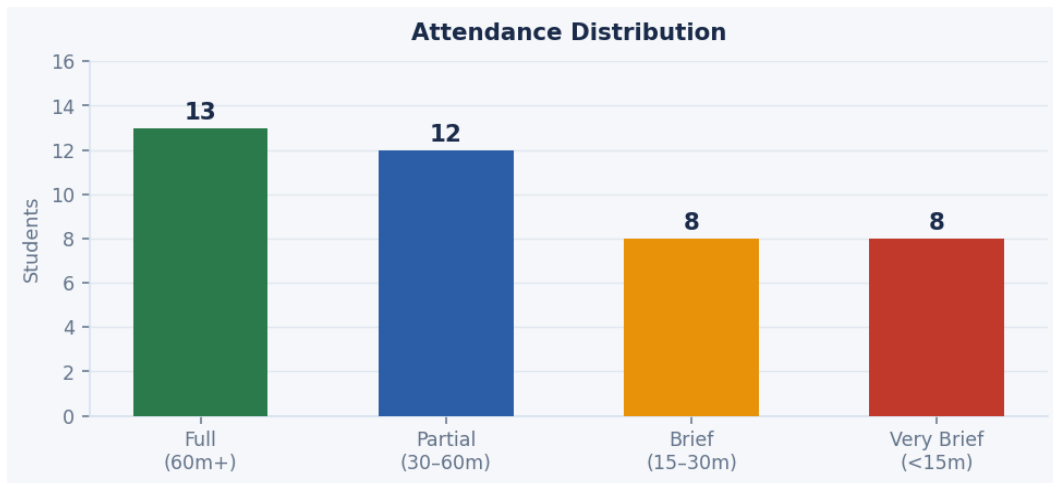
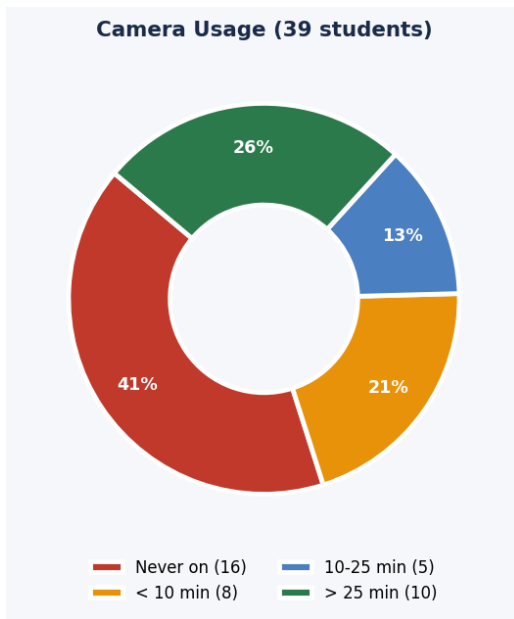


Fig 1 — Attendance distribution across duration buckets

14 out of 39 students were present for less than half the session. Very brief attendees: Student H (4m), Student I (7m), Student J (6m), Student K (5m), Student L (5m).

# 2. CAMERA USAGE & FACE-DETECTED PRESENCE

Camera data is the primary signal source for AttentionTag's engagement analysis. When a student's camera is off, **no engagement data can be collected**. This session had critically low camera participation.



**16** students with camera never on

**7** students with 20+ min detected presence

**26** students with <5 min detected presence

**Students with reliable presence data (20+ min detected):**

Student A (64m)	Student B (62m)	Student C (40m)	Student D (31m)
Student E (31m)	Student F (24m)	Student G (20m)	

The engagement analysis below is most reliable for these 7 students. For all others, presence data was too limited for meaningful emotion or focus analysis.

### 3. ENGAGEMENT INSIGHTS

The following insights are drawn from students with substantial camera-on time. AttentionTag tracks **Positivity** (emotional tone, 0–100), **Intensity** (activation level, 0–100), **Drowsiness**, and face-detected **Presence**. A Positivity score below 50 indicates negative emotions; above 50 indicates positive engagement.

#### 3.1 Key Disengagement Moments

**Min 6** Confusion during class start-up logistics. The teacher was managing attendance and camera requests, drawing attention away from content.

**Min 33–36** Students showed difficulty during the innerHTML vs DocumentFragment demo. Engaged but not confident — performance internals (browser reflows) were challenging.

**Min 56–60** Highest difficulty in the session. The prototype / \_\_proto\_\_ / constructor / super explanation caused the most confusion. Multiple students became non-visible during this window.

#### 3.2 What Worked Well

**Q&A** **Named direct questions (Min 2–12)**  
Calling students by name and using concrete recall questions reliably increased positivity. The highest positivity readings in the session coincided with this period.

**Live** Live coding in VS Code & browser console

Executing visible changes in the editor and console produced repeated bursts of student attention, especially around minutes 30–40.

**Link** Tying content to the capstone project (~Min 30)

Connecting DocumentFragment to the upcoming library-management project noticeably increased motivation and student interest.

### 3.3 Individual Student Signals

Student	Presence	Engagement Signal
Student B	62m	Consistently positive tone. Most engaged student in session.
Student A	64m	Steady neutral-to-positive. Highest visibility.
Student C	40m	Stable engagement across tracked minutes.
Student D	31m	Consistent neutral-to-positive tone.
Student E	31m	Consistent neutral-to-positive tone.
Student F	24m	Signs of difficulty during prototype/inheritance segment.
Student G	20m	Mixed engagement patterns across session.

## 4. STUDENTS FLAGGED FOR FOLLOW-UP

### Attendance Concerns — attended for very short durations:

Student	Duration	Note
Student H	4 min	Too brief for any engagement analysis.
Student I	7 min	Camera on 4m, no detected presence.
Student J	6 min	Camera on 5m, <1m presence.
Student K	5 min	Camera on 5m, <1m presence.
Student L	5 min	No camera data.

### Camera-Off Concerns — attended 50+ min but never turned camera on:

Student	Duration	Note
Student N	91 min	Full session. Zero camera time.
Student O	90 min	Full session. Zero camera time.
Student P	89 min	Full session. Zero camera time.
Student Q	64 min	No camera.
Student R	59 min	No camera.
Student S	54 min	No camera.



## 5. RECOMMENDATIONS FOR NEXT SESSION

- 01 Start with a warm-up question**  
Replace attendance logistics at the top with a 2-minute quick-fire Q&A. Don't wait for full attendance before starting content — early confusion (Min 6) was driven by this delay.
- 02 Side-by-side predict-then-reveal**  
For DOM demos (innerHTML vs DocumentFragment), ask students to predict the outcome before running the code. This anchors understanding and turns passive watching into active engagement.
- 03 Pause before prototype/inheritance**  
The data shows this was the hardest block of the session (Min 56–60). A 60-second readiness check — "who can explain the prototype chain in one sentence?" — before diving in significantly reduces the risk of mass confusion.
- 04 Consolidation gap between dense topics**  
Insert a 2-minute summary slide or student-recap moment between DOM performance internals and OOP/prototypes. Running them consecutively created compounding cognitive load.
- 05 Camera-on as a participation metric**  
16 students never turned cameras on — including 3 who stayed the full 90+ minutes. This severely limits AttentionTag's data coverage. Consider making camera-on a participation requirement.

## 6. CONCEPT STRENGTHENING EXERCISES

The following exercises target the three highest-difficulty topics identified in this session. They are designed for assignment or in-class review.

**DOM** **1000-item append benchmark**

Write two snippets: one that appends 1,000 list items using repeated `createElement/append`, and one using a `DocumentFragment`. Measure render time. Students explain why the second approach reduces browser reflow cost.

**OOP** **Person → Student inheritance chain**

Create class `Person(name)` with a method `species()`. Extend with `Student` that: (1) calls `super()` in constructor, (2) calls `super.species()` inside a child method, (3) adds a new property `grade` and a method `info()`.

**Storage** **localStorage array round-trip**

Store and retrieve an array of objects using `JSON.stringify / JSON.parse`. Deliberately omit `JSON.stringify` in one attempt and observe the result. Explain what happens to complex data types when stored as raw strings.

**OOP** **Prototype chain diagram**

Draw or describe the prototype chain for an instance created with `new Student()`. Mark where instance properties live, where shared methods live, and trace `__proto__` all the way to `Object.prototype`.



**MCQ Quick knowledge check**

Which storage persists across browser tabs AND browser restarts? A) sessionStorage B) localStorage C) In-memory variable D) HTTP cache Answer: B — localStorage. sessionStorage is tab-scoped and cleared on tab close.

---

Report generated by **AttentionTag** · Classroom Engagement Intelligence · [attentiontag.com](https://attentiontag.com)