Week 3: Client-side scripting (JavaScript)

Slides: `slides/02_js.pdf`
Time: **50 minutes** (Tune In **10** / Activity **25** / Reflection **5** / Project Preparation **10**)

Get this instruction:
https://github.com/ruhendrawan/cs1520_recitation

How to work today

Work in pairs

Submit in the google form:
https://forms.gle/tYEtKjJunM1wb2we6

Part 1 — Tune In (10 minutes)

Pairs: talk in turns. Max **1 minute** per person before switching so everyone gets a chance to talk.

Pick **1–3 concepts** from this week's lecture and explain them in your own words (fast + messy is fine).

Concept ideas (pick any 1–3):

- JavaScript in HTML (`<script>...</script>` / external script)
- JavaScript vs ECMAScript (spec vs implementation)
- Dynamic typing + type coercion (== vs ===, `parseInt`, `NaN`)
- `var` vs `let` vs `const` (scope + reassignment)
- Template strings (backticks + `${...}`)
- First-class functions + arrow functions
- Arrays (methods like `push`/`pop`, `sort`, `join`)
- Objects / ES2015 classes
- Strict mode (`"use strict"`) and why it can break sloppy code

Part 1 — Starter questions

- What is it (in one sentence)?
- What's a super common bug this concept prevents?
- What did you *think* it meant before today vs what it actually means?
- If you had to teach this to a friend in 20 seconds, what would you say?
- Where would you use it in a website/app you care about?

Part 2 — Activity (30 minutes)

Break into pairs and run js02 and js03

https://github.com/nfarnan/cs1520_examples/blob/main/javascript/js02_vars_types.html https://github.com/nfarnan/cs1520_examples/blob/main/javascript/js03_more_vars_types.html https://github.com/nfarnan/cs1520_examples/blob/main/javascript/js03_more_vars_types.js

Each pair should make a guess filling out each comment, and then run the code to check the actual result.

- Open DevTools Console

References:

- MDN: JavaScript
- ECMA-262 spec
- MDN: Strict mode
- MDN Learn: JavaScript first steps
- MDN: JavaScript Guide

Part 3 — Reflection (5 minutes)

Pairs: talk in turns. Max **1 minute** per person before switching so everyone gets a chance to talk.

A) What's the programming concepts/ ideas that you see in the examples?

- Quick checklist is fine, no overcorrection needed

B) What's the difference in comparison to other prgramming language?

- Quick checklist is fine, no overcorrection needed

C) Connect it back (prior lecture)

Prior lecture idea to connect: **responsive design**

- Examples: What would you keep CSS-only vs use JS for?
  - menu toggles, show/hide panels, saving preferences, validation, content updates

Submit in the google form:
https://forms.gle/tYEtKjJunM1wb2we6

Project Preparation (10 minute)

1. Install NPM https://nodejs.org/en/download

JS = Interpreted (Just-in-Time/ JIT compiled) Language. Node.js is a JavaScript **Runtime Engine**. Long-term Support (LTS), even versions, is more stable.

```
node -v
npm -v
```

JS Application can be composed from others work, published with the NPM.

For example, https://www.npmjs.com/package/react

2. Create Empty React Project
   https://react.dev/learn/build-a-react-app-from-scratch

```
npm create vite@latest my-app -- --template react-ts
```

https://react.dev/learn/react-developer-tools

Do this part at home, spend 1-2 hour. Don't rush to finish, but learn by doing at your pace.

It's okay to not finish all parts.

- Quick Start: https://react.dev/learn
- Creating UI component:
  https://react.dev/learn/describing-the-ui