
100 PROJECTS IN 100 DAYS
HTML, CSS & JAVASCRIPT
Day 50

HTML Concepts

Elements

- <p>
- <mark>
- <input type="number">
- <input type="search">

DOM Structure & Properties

- **textContent**
- **tagName**
- **classList**
- **nextElementSibling**
- **previousElementSibling**
- **offsetWidth**
- **activeElement**

DOM Methods

- **remove()**
- **cloneNode()**
- **insertBefore()**
- **replaceWith()**
- **closest()**
- **contains()**
- **focus()**

Attributes

- **inputmode**
- **pattern**
- **disabled**
- **draggable**
- **src** property
- **dataset** property (using data attributes)

Accessibility & Extras

- Understanding ARIA live regions
- Understanding Markdown
- HTML Drag and Drop API

CSS Concepts

Layout

- Basic concepts of **Flexbox**
- **flex-direction**
- **align-items**
- Basic concepts of **grid layout**
- **display**
- **inset**
- **z-index**
- Viewport-percentage lengths

Styling, Text & Selectors

- **font-weight**
- **text-decoration**
- **user-select**
- **setProperty()** method
- **hsl()** function
- HEX to HSL conversion
- **:hover** pseudo-class
- **::before** pseudo-element
- **::after** pseudo-element

Animations & Effects

- Using CSS animations
- **animation**
- **animation-play-state**
- Shake animation
- CSS Skeleton Loadings

JavaScript Concepts (I)

Language Fundamentals

- JavaScript functions
- Arrow function expressions
- For statement (**for**)
- Don't repeat yourself (DRY)
- Event bubbling/delegation

Objects

- **Object**
- **Array**
- **String**
- **RegExp**
- **Date.now()** method
- **JSON.stringify()** method
- Working with objects

Events

- **keydown**
- Key values for keyboard events
- **dblclick**
- **input**
- **paste**
- **mouseenter**
- **mouseleave**
- **resize**
- **scroll**
- **transitionend**
- **visibilitychange**
- **dragstart**
- **drop**

JavaScript Concepts (II)

String Methods

- **trim()**
- **replace()**
- **toLowerCase()**
- **toLocaleString()**
- **localeCompare()**
- **String.fromCharCode()**

Numbers & Math

- Remainder (%) operator
- **parseInt()** function
- **Math.min()** method
- **Math.floor()** method
- ASCII characters

Async & Timing

- Using promises
- **Promise.all()** method
- **await** operator
- **clearTimeout()** method
- Debounce
- **requestAnimationFrame()** method

Array Methods

- **push()**
- **unshift()**
- **splice()**
- **slice()**
- **sort()**
- **filter()**
- **find()**
- **includes()**
- **some()**
- **join()**
- **split()**
- **Array.from()**
- keys()
- How to randomize an array

DOM & Interaction

- **target** property
- Clipboard API (**getData()**)
- **confirm()** method
- **location.reload()** method

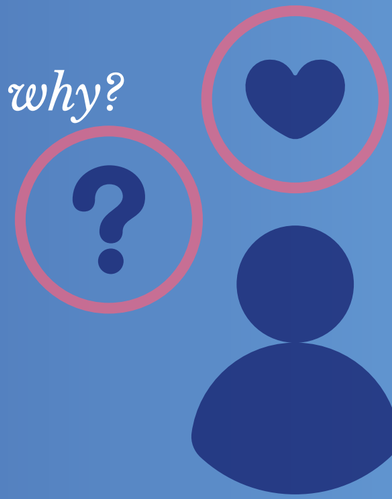
Identifying Strengths & Weaknesses

Where Do You Stand?

Which HTML, CSS, and JavaScript concepts can you confidently explain and apply?

What's your favorite topic so far, and why?

Do you feel stronger in CSS (design) or JavaScript (logic)?



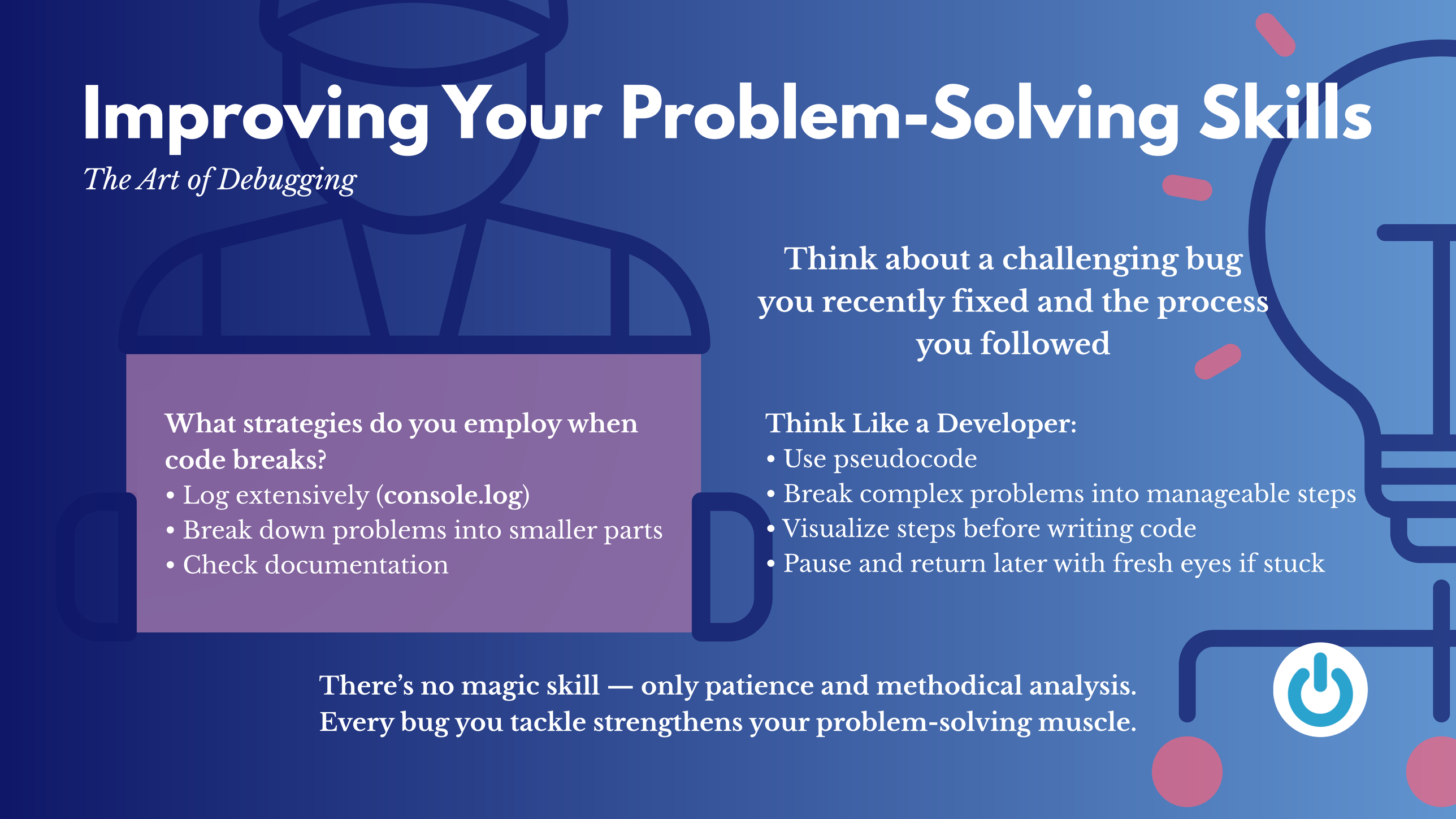
Deliberate Practice Still Works

1. **Pinpoint** areas that still require significant effort
2. **Plan** to practice those areas deliberately
3. **Repeat** until they click

Had an “aha” moment? Share it with the community!

Improving Your Problem-Solving Skills

The Art of Debugging



Think about a challenging bug you recently fixed and the process you followed

What strategies do you employ when code breaks?

- Log extensively (**console.log**)
- Break down problems into smaller parts
- Check documentation

Think Like a Developer:

- Use pseudocode
- Break complex problems into manageable steps
- Visualize steps before writing code
- Pause and return later with fresh eyes if stuck

There's no magic skill — only patience and methodical analysis.
Every bug you tackle strengthens your problem-solving muscle.