

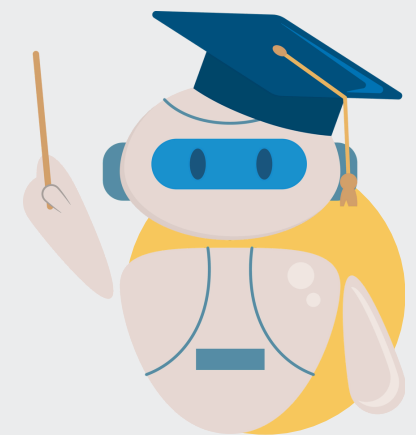


Problem Solving



What is Problem Solving in Coding?

Problem solving in coding means breaking down a challenge or bug in your code and finding a solution that works. It's not just about writing code—it's about thinking logically, testing ideas, and improving your approach.



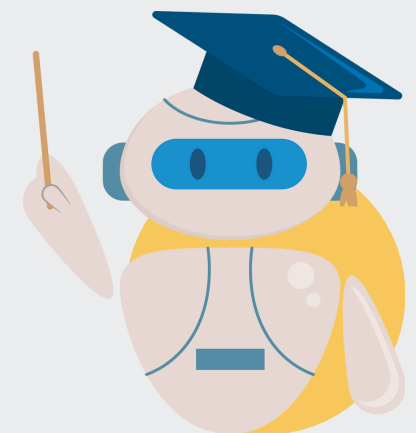


Problem Solving



What is Critical Thinking in Programming?

Critical thinking in programming helps you: - Understand the problem deeply - Question why something isn't working - Compare different coding solutions - Think ahead about the possible outcomes of your code



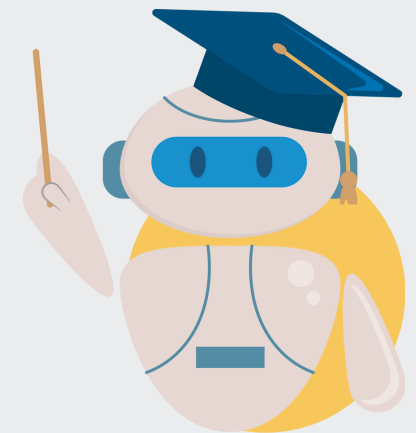


Problem Solving



Why Are These Skills Important for Coders?

- Debug code faster and more efficiently
- Build smarter and cleaner solutions
- Handle real-world software challenges
- Write code that other developers can understand
- Grow as a developer and thinker



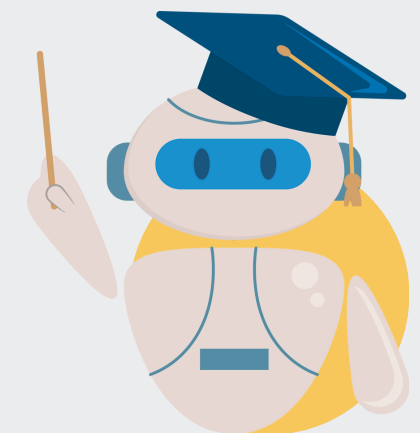


Problem Solving



Steps of Problem Solving (for Programmers)

1. Identify the Problem – What is the bug, error, or challenge?
2. Analyze the Problem – Read error messages, inspect code behavior.
3. Break It Down – Simplify the issue into small parts or steps.
4. Research – Look up similar problems, documentation, or tutorials.
5. Try a Solution – Write code to test an idea.





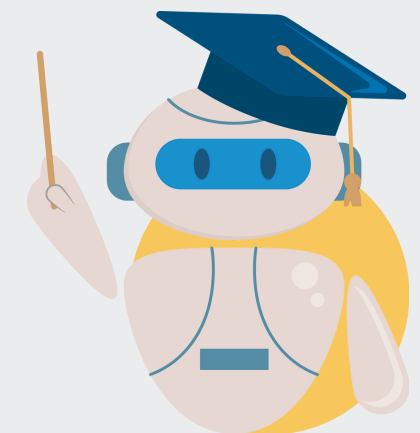
Problem Solving



Steps of Problem Solving (for Programmers)

6. Debug and Improve – Use tools like console.log or a debugger.

7. Reflect and Learn – What worked? What didn't? Document it.





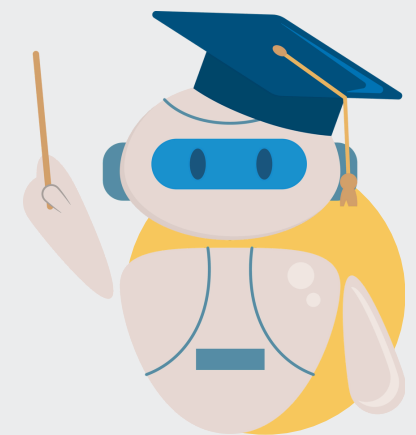
Problem Solving



Real Coding Exam

Problem: A button doesn't do anything when clicked.

- Identify: Button is unresponsive.
- Analyze: Check if the event listener is added.
- Break Down: Is the script loading correctly? Is the selector right?
- Research: Search “button click not working JavaScript”.
- Try: Add `console.log("Clicked")`
- Fix: Correct the selector and function. to see if it works.
- Learn: Next time, always check connections and browser console.



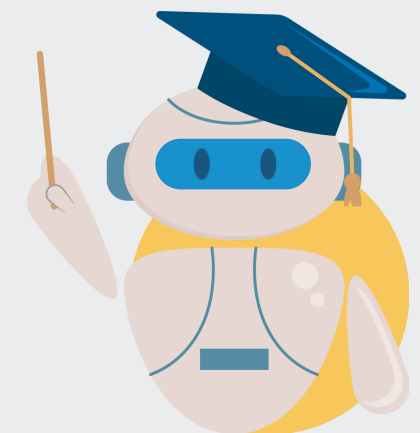


Problem Solving



Critical Thinking in Coding Tasks

- Compare methods: Should you use `.map()` or a for loop?
- Think ahead: What happens if the input is empty?
- Question defaults: Is the default behavior the best one?
- Test edge cases: What if someone enters unexpected data



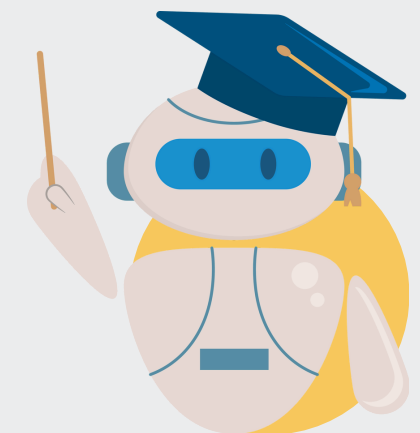


Problem Solving



Fun Developer Challenges to Practice

1. **Build a Calculator** – Break down input, operations, and output.
2. **Debug a Broken Game** – Fix bugs step-by-step using your logic.
3. **Code Golf** – Solve a problem using the least amount of code.
4. **Solve with Constraints** – Build a layout without using Flexbox.



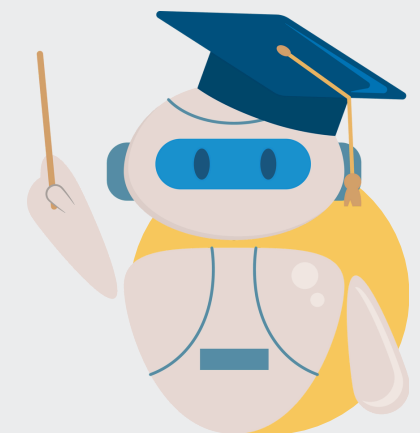


Problem Solving



Using Problem Solving in Web Development

- HTML/CSS: Why isn't this element styled right? Check selectors, cascade, or syntax.
- JavaScript: Why is this function not returning what I expect?
- Backend: Why is my API not sending the right data?
- Database: Why am I not seeing my saved data?



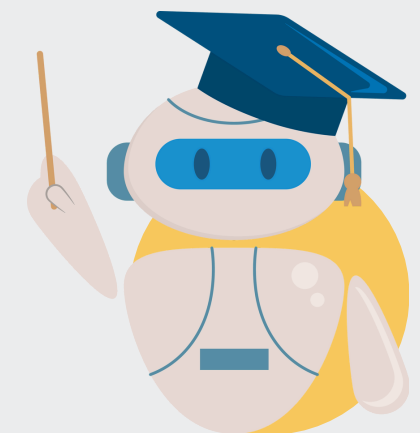


Problem Solving



Quick Tips for Coders

- Comment your logic—it helps you and others.
- Don't copy-paste without understanding.
- Practice small challenges daily (e.g., on Codewars, LeetCode).
- Ask “Why is this happening?” not just “How do I fix it?”
- Always test your code with weird or wrong inputs.



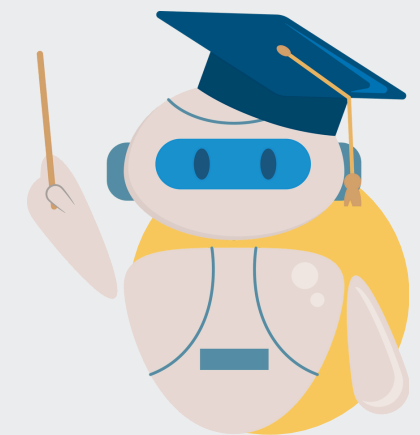


Problem Solving



Conclusion

Every great coder is a great thinker. Problem solving and critical thinking help you fix bugs, build smart apps, and grow your confidence. With time and practice, you'll go from writing code to creating powerful, creative solutions.





Problem Solving



Quote to Remember:

"Programs must be written for people to read, and only incidentally for machines to execute." — Harold Abelson

