Reusable code identified:
(1) Getting the selected deck size; (2) computing study time; (3) computing total cost with discount; (4) rendering the result block; (5) validating inputs.

Planned functions:

- `getSelectedDeckSize()` → no parameters, returns a number.
- `computeStudyMinutes(numCards, shuffleOn)` → two parameters, returns minutes (number).
- `computeCost(numCards, pricePerCard)` → two parameters, returns { `subtotal, discount, total` }.
- `renderEstimate(topic, numCards, minutes, totals)` → writes innerHTML, no return.
- `hasValidInputs(topic, numCards)` → returns boolean.
- Basic function: `showGreeting()` → no params/return, updates a greeting banner on load.

Why functions?
They improve readability, isolate logic for easier debugging, and avoid duplication. If requirements change (e.g., new discount tiers), I can update a single function without touching UI code.

New functions proposed:

- `validateTopic(topic)` to enforce a minimum topic length and trim whitespace (param: string; returns sanitized string or empty).
- `formatCurrency(value)` to standardize price display (param: number; returns string).
- `toggleLoading(state)` to show/hide a loader during future async features (param: boolean; no return). These enhance UX and keep concerns separated.