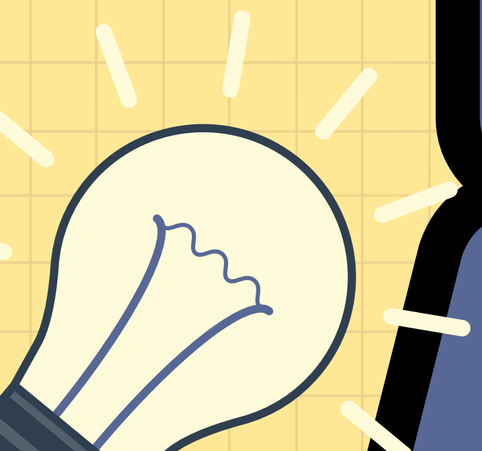
Nafisa's Section

CS106A - Code in Place 2025

Section 4



Week Four Overview

Ice Breaker & Check-in

Quick Concept Review

Understanding the "High-Low" Game

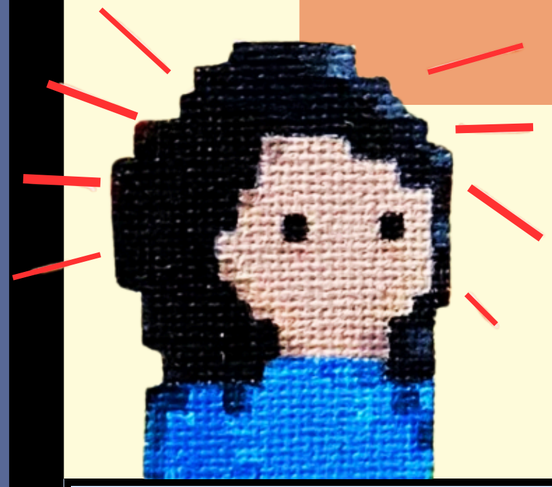
Building the game together

Testing and Refinement (Optional) Fun Extensions!

Wrap-up and Q&A

Let's Connect!

- What's your favorite game?



Quick Review: The Building Blocks of Decisions

- Boolean Values: True, False.
- Comparison Operators
  - > (greater than)
  - < (less than)
  - >= (greater than or equal to)
  - <= (less than or equal to)
  - == (equal to)
  - != (NOT equal to)

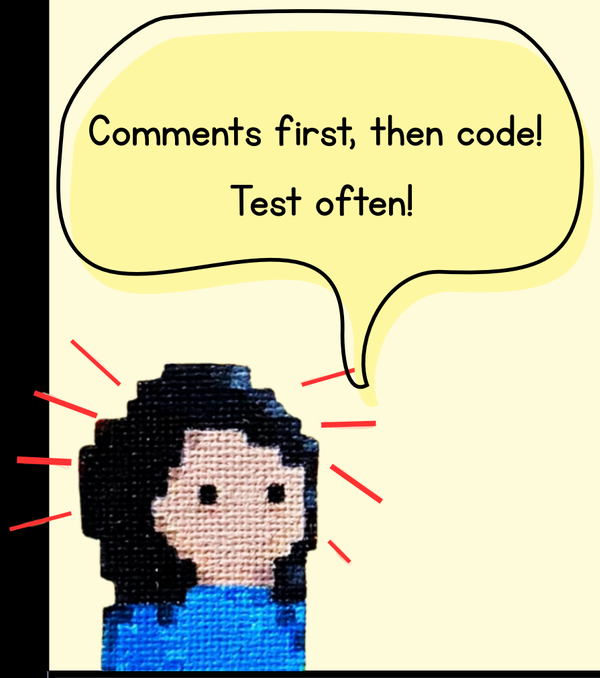
- Logical Operators:
  - and: (condition1 and condition2) - True if BOTH are true
  - or: (condition1 or condition2) - True if AT LEAST ONE is true
  - not: (not condition1) - Reverses the boolean value

How would we check if a score is greater than 10 AND a name is 'Alice'?

High-Low: How to Play

- Two numbers (1-100) are generated: one for you, one for the computer.
- You see YOUR number. The computer's number is hidden.
- You GUESS if your number is "higher" or "lower" than the computer's.
- If your guess is correct, you get 1 point!
- If your number IS THE SAME as the computer's, your guess ("higher" or "lower") will be incorrect (computer wins the tie).
- Play for a set number of rounds.

Breaking it Down: Our Milestones



- Generate Random Numbers: Get numbers for player and computer.
- Get User Choice: Ask player "higher" or "lower".
- Game Logic: Determine if the player's guess was correct.
- Play Multiple Rounds: Loop the game.
- Adding a Points System: Track and display the score.

Milestone 1: Generating Numbers

- Goal:
  - Generate a random number for the player (1-100).
  - Generate a random number for the computer (1-100).
- Question: Inside main(), (and for now, just for one round), how would we create these two numbers and print them?

Milestone 2: Player's Turn!

- Goal:
  - Show the player THEIR number.
  - Ask the player to guess "higher" or "lower".
  - Store their guess.
- Question: How do we display the player's number and get their input?

Milestone 3: Was the Guess Right?

- Goal: Compare the player's number, computer's number, and player's guess to see if they won the round.
- Conditions for Player to WIN:
  - Player guessed "higher" AND player's number IS GREATER than computer's.
  - Player guessed "lower" AND player's number IS LESS than computer's.
- Otherwise (including ties): Player is incorrect.
- Question: What kind of Python structure helps us check different conditions? And what about the user's input? They might type 'Higher', 'higher', or 'HIGHER'. How can we handle that easily?

Milestone 4: Again! Again!

- Goal:
  - Play the game for NUM_ROUNDS.
  - Print the current round number (e.g., "Round 1").
  - Add a blank line (print()) after each round for readability.
- Question: What parts of our current code need to be inside this loop?

Milestone 5: Who's Winning?

- Goal:
  - Keep track of the player's score.
  - Initialize score to 0 before the rounds start.
  - If the player wins a round, increment their score.
  - Print the player's score after each round.
  - Print a final "Thanks for playing!" message after all rounds.
- Question: Where should we initialize the score variable? Where should we update it? Where should we print it?

We Did It! (Base Game)

- Recap the main structure:
  - main() function.
  - Score initialization.
  - for loop for rounds.
    - Generate numbers.
    - Get user guess.
    - Determine win/loss & update score.
    - Print round results.
  - Final message.

Extensions!

- Option 1: Safeguard User Input
  - What if the user types "banana" instead of "higher" or "lower"?
  - Goal: Keep asking until they enter valid input.
  - Hint: A while loop can be useful here!

Extensions!

- Option 2: Conditional Ending Messages
  - Give different feedback based on the final score:
    - Perfect game?
    - Won at least half?
    - Won less than half?
  - Hint: if/elif/else after the game loop.

What We Practiced Today

- Control Flow is Power!
  - for loops for repetition.
  - if/elif/else for making decisions.
  - while loops for repeating until a condition is met.
- Booleans are the Gatekeepers: True/False values drive our logic.
- Problem Decomposition: Milestones help break big problems into small, manageable steps.
- Iterative Development: Build a little, test a little.
- The Joy of Debugging! Finding and fixing bugs is part of the process.

THANK YOU

See you in the next session!