[Open]

Debug Lab Skills Learning Center

Technical Assessment – Full Time Coding Coach

1. **Block Coding**

Summary

Code.org teaches students advanced problem-solving, loops, conditionals, functions, and debugging using block coding. Block coding is a visual programming by stacking blocks that represent code commands, helping learners to understand logic and structure without worrying about syntax. It builds foundational computational thinking and prepares students for real programming languages.

Submission

Complete by 29th October 2025 11:59PM

Login Instructions

Login Link: https://studio.code.org/join/MJMLGM

Username: hazma

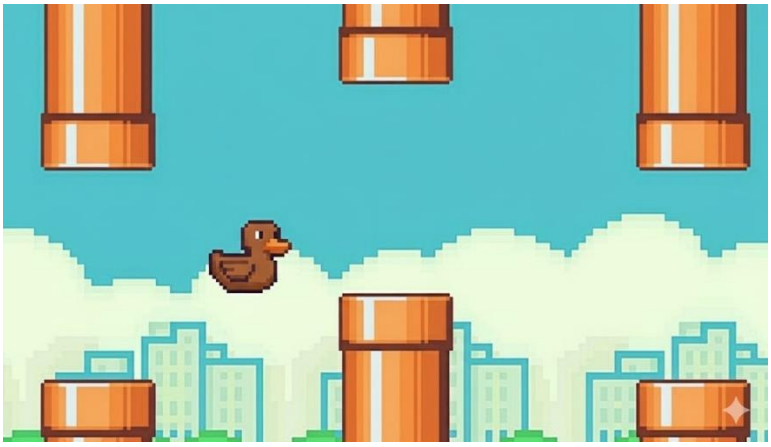
Password: hamster

Definition of Done (DOD)

Complete all the puzzles from lesson 1 to lesson 21 with a perfect completion (dark green)

[Open]

2. **Scratch**



Summary

Scratch programming is a visual, block-based coding platform developed by MIT to teach computational thinking and creativity. It allows learners to create interactive stories, games, and animations by dragging and connecting code blocks, promoting logical reasoning, problem-solving, and collaboration without syntax barriers and it's ideal for introducing programming concepts to beginners.

Submission

Submit the project link by 31st October 2025 11:59PM

Login Instructions

Create and login to your account in https://scratch.mit.edu/

Requirements

Replicate the game mechanism and logics from https://flappybird.io/ but change with Debug Lab theme

Features

A. **Title Display** – "Flappy Duck" or your chosen game title shown prominently at the center or top.
B. **Start Button** – Large, visible button labeled "Start Game" (or "Play Now").
   When clicked/tapped → navigates to the difficulty selection.
C. **Difficulty Selection** – Three buttons labeled **Easy**, **Medium**, and **Hard** allowing the player to choose difficulty before starting.
   a. **Easy**: Slow pipe speed, large gap.
   b. **Medium**: Normal pipe speed, medium gap.
   c. **Hard**: Fast pipe speed, small gap.
D. **Background Design** – Simple sky or pixel-art background consistent with the game theme.
E. **Sprite –** Simple pixel-art duck consistent with the game theme

F. **Gravity Effect –** The duck continuously falls due to gravity and rises when the player clicks or taps.
    a. Duck tilted downward as being pulled by the gravity
G. **Flap Mechanism** – Player presses the spacebar make the duck flap upward briefly.
    a. Gravity continuously pulls the duck downward when not flapping.
H. **Pipe Movement** – Pairs of pipes move from right to left; when off-screen, new pipes spawn with random vertical gaps
I. **Scoring System**
    a. +1 point when the duck successfully passes a pipe pair.
    b. Current score displayed on screen.
J. **High Score –** Display the high score from the previous session.
K. **Sound Effects**
    a. Flap sound when spacebar pressed.
    b. “Point” sound when scoring.
    c. “Hit” sound when colliding.
    d. Background music pauses when game over.s
L. **Collision Detection** – If the duck collides with a pipe or the ground, the game ends. Game Over Display appears with:
    - “Play Again” button to restart

[Open]

3. HTML/CSS/JavaScript

Summary

HTML defines the structure of a web page. CSS controls its visual presentation, including layout, colors, and fonts. JavaScript adds interactivity and dynamic behavior, enabling user interaction, animations, and real-time updates. Together, HTML, CSS, and JavaScript create fully functional, visually appealing, and interactive web experiences.

Submission

Submit the project link by 4th November 2025 11:59PM

Login Instructions

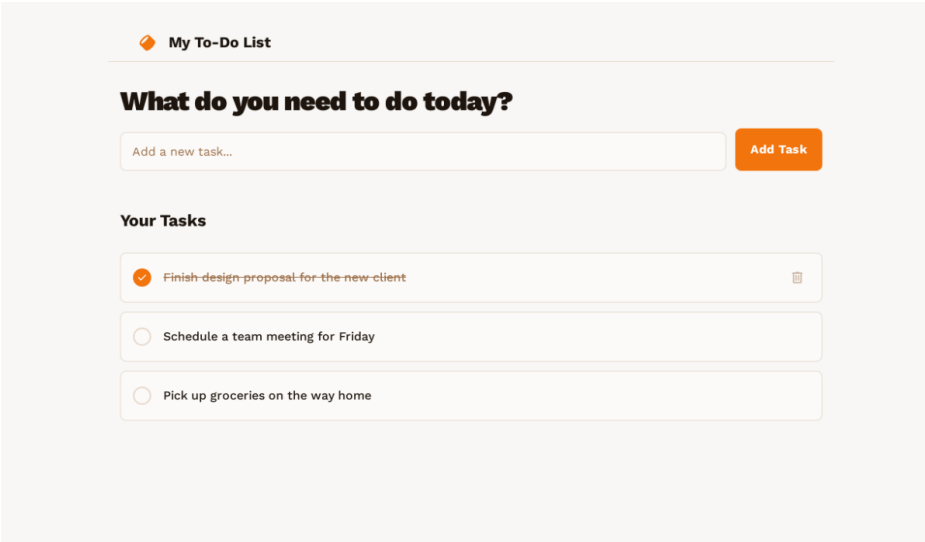
Create and login to your account in https://codepen.io/. You are required to use codepen.io compiler to develop the project below.

Learning Materials

Create account in FreeCodeCamp to continue with these courses

1. HTML & CSS
   https://www.freecodecamp.org/learn/2022/responsive-web-design/learn-html-forms-by-building-a-registration-form/step-1
2. Integrating Javascript with traditional website
   https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures-v8/learn-form-validation-by-building-a-calorie-counter/step-1
3. Creating Simple To Do List
   https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures-v8/learn-localstorage-by-building-a-todo-app/step-1

Project Requirements



1. Task Display
   a. Follow the theme and structure as above with accurate colour code
   b. Show a list of all added tasks on the page.

   c. Each task shows:
      i. Task name
      ii. Status (completed task displayed in crossed out)
2. Add Task
   a. Text input field to type the new task name.
   b. Add Button next to the input field.
   c. Clicking Add:
      i. Adds the new task to the task list.
      ii. Clears the input field.
      iii. Validates input (non-empty).
3. Mark as Completed
   a. Users can mark tasks as completed.
   b. Completed tasks visually differ with a strikethrough.
4. Delete Task
   a. Users can remove any task from the list.
   b. Deletion updates the task list immediately.