



This is the Learn to Code Companion Doc for:
<http://doomscrollquest.neocities.org/learning.html>

Let's get right into it! Building a web game means understanding your tools. Here is the tag-by-tag and command-by-command breakdown of the HTML architecture and JavaScript engine powering Dragon Survival.

The HTML Structure

These tags form the skeleton of the page that interface.

- **<!DOCTYPE html>**: Declares the file as a modern HTML5 document.
- **<html>**: The root element wrapping the entire web page.
- **<head>**: Contains unseen background data like character encoding, the page title, and styling rules.
- **<meta>**: Sets the character encoding to UTF-8 and handles responsive viewport scaling for mobile devices.
- **<title>**: Defines the name appearing on the browser tab.
- **<style>**: Holds the CSS rules giving the page its retro black and neon green aesthetic.
- **<body>**: Contains all visible content. This includes text, buttons, and the game board.

- **<div>**: A general container used to group the control buttons and format the layout.
- **<h1>, <h2>, <h3>**: Heading tags that organize the tutorial text into distinct sections.
- **<p>**: Paragraph tags holding the lesson text.
- ****: Bolds text for visual emphasis.
- ** & **: Create the bulleted list detailing the bug-hunting methodology.
- **<canvas>**: The HTML5 element acting as the drawing board for the 2D game graphics.
- **<button>**: Clickable elements serving as the directional pad and action keys.
- **<script>**: Houses the JavaScript code running the game logic.

Core JavaScript Syntax

These structures form the foundation of the game's logic.

- **const**: Declares a variable that cannot be reassigned later, like the canvas element.
- **let**: Declares a variable that changes over time, such as player coordinates or the game loop ID.
- **function**: Defines a block of reusable code. You use this for repeatable actions like checking collisions or resetting the board.
- **return**: Exits a function immediately. It can also send a value back, like a true or false result from a collision check.
- **if / else if / else**: Conditional statements dictating logic branches. These decide what happens when objects overlap.
- **=> (Arrow Function)**: A shorter syntax for writing functions. The engine uses this heavily in the event listener assignments.

The JavaScript Engine & Events

These commands handle math, timing, and user inputs.

- **document.getElementById()**: Grabs a specific HTML element by its ID so the code can interact with it.
- **getContext('2d')**: Tells the canvas to operate in two dimensions and unlocks the drawing API.
- **addEventListener()**: Listens for user input like key presses or pointer clicks.
- **e.preventDefault()**: Stops default browser behaviors. This prevents the arrow keys from scrolling the web page during gameplay.
- **hasOwnProperty()**: Checks if a specific key pressed exists within the predefined object of mapped controls.
- **Date.now()**: Grabs the current timestamp. The game uses this to calculate exactly when the dragon should respawn.
- **Math.random()**: Generates a random number. This adds jitter to the NPC movement for unpredictable wandering.

- **forEach():** Loops through an array. This lets the game update every NPC's position in a single pass.
- **requestAnimationFrame():** Instructs the browser to run the update function right before the next screen repaint. This creates a smooth game loop.
- **cancelAnimationFrame():** Halts the game loop entirely when a win or loss state triggers.
- **confirm():** Prompts a browser popup asking the player a question with OK and Cancel choices.

Canvas Drawing Commands (ctx)

These commands paint the actual pixels onto the screen frame by frame.

- **ctx.clearRect():** Wipes the canvas clean at the start of every frame to prevent graphics from smearing.
- **ctx.fillStyle:** Sets the paint color for the next shape or text drawn.
- **ctx.font:** Defines the typography style and size for canvas text.
- **ctx.textAlign:** Centers or left-aligns the text on the screen.
- **ctx.fillText():** Paints the actual text directly onto the canvas.
- **ctx.fillRect():** Draws the colored rectangles representing the player, dragon, walls, chest, and NPCs.