Week 1 Full-Stack Developer Curriculum: Complete Implementation Guide

Environment Setup, JavaScript Fundamentals, HTML5/CSS, and Git Basics

---

📋 WEEK 1 SUCCESS CRITERIA

- ✅ **VS Code fully configured** with ESLint, Prettier, GitLens, Live Server
- ✅ **GitHub account + first repository** pushed with proper commit history
- ✅ **JavaScript ES6+ proficiency**: let/const, arrow functions, template literals, destructuring
- ✅ **3 complete HTML/CSS pages** demonstrating semantic structure and responsive design
- ✅ **5+ LeetCode Easy problems solved** with understanding of multiple approaches
- ✅ **Consistent daily routine** established with depression management protocols

---

🎯 DAILY SCHEDULE STRUCTURE (10-12 hours/day)

6:00-7:30 AM - Morning Routine (exercise, breakfast, mental prep)
8:00-9:50 AM - BLOCK 1: JavaScript Fundamentals (110 min)
9:50-10:10 AM - Break (20 min)
10:10-12:00 PM - BLOCK 2: HTML/CSS Fundamentals (110 min)
12:00-12:30 PM - Lunch + Active Break
12:30-2:20 PM - BLOCK 3: Git Practice (110 min)
2:20-2:40 PM - Break (20 min)
2:40-4:30 PM - BLOCK 4: Project Work (110 min)
4:30-5:00 PM - Extended Break
5:00-6:50 PM - BLOCK 5: LeetCode Problems (110 min)
6:50-7:10 PM - Break (20 min)
7:10-8:30 PM - BLOCK 6: Review, Documentation, Planning (80 min)
8:30-10:00 PM - Wind-Down Routine

DAY 1: FOUNDATIONS

Morning Routine (6:00-7:30 AM)

Physical Activity (20 min): Brisk walk, yoga, or bodyweight exercises
Mindfulness (15 min): 10 min meditation, 5 min journaling
Breakfast (30 min): High-protein meal, complex carbs, hydrate (16oz water)
Setup (25 min): Review goals, open workspace, mental state check (rate mood 1-10) HelpGuide.org↗

BLOCK 1: JavaScript - Values, Types & Operators (8:00-9:50)

Reading (60 min)

- Eloquent JavaScript Ch1: https://eloquentjavascript.net/01_values.html

- JavaScript.info Data Types: https://javascript.info/types
- **Code along in browser console (F12)**

Key Concepts: Numbers, Strings, Booleans, null, undefined, typeof, operators, type coercion, == vs === Eloquent JavaScript +3↗

Practice (40 min)


javascript

```javascript
// Exercise 1: Data Types
let myName = "John Doe";
let myAge = 25;
let isLearning = true;
console.log(typeof myName);    // "string"
console.log(typeof myAge);     // "number"
console.log(typeof isLearning); // "boolean"

// Exercise 2: Type Coercion
console.log("5" + 3);    // "53" (concatenation)
console.log("5" - 3);    // 2 (numeric subtraction)
console.log(5 == "5");   // true (type coercion)
console.log(5 === "5");  // false (strict equality)
```

Checkpoint (10 min): 1) Difference between == and ===? 2) Why does `typeof null` return "object"? 3) Explain `"5" + 3` vs `"5" - 3` GeeksforGeeks↗

BLOCK 2: HTML - Document Structure (10:10-12:00)

Reading (50 min)

- MDN Getting Started with HTML: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started
- MDN HTML Document Structure: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML

Key Concepts: DOCTYPE, html/head/body, meta tags, title, semantic elements (h1-h6, p, strong, em)

Practice (50 min)


html

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Day 1: My Coding Journey</title>
</head>
<body>
  <h1>Welcome to My Coding Journey</h1>
  <h2>Day 1: The Beginning</h2>
  <p>Today I learned about <strong>HTML structure</strong> and <em>JavaScript types</em>.</p>

  <h3>What I Learned:</h3>
  <ul>
    <li>HTML5 document structure</li>
    <li>JavaScript data types</li>
    <li>Git initialization</li>
  </ul>

  <footer>
    <p>Created by [Your Name] | Week 1 Day 1</p>
  </footer>
</body>
</html>
```

Checkpoint: 1) Purpose of DOCTYPE? 2) Semantic elements vs div/span? 3) Difference between `<strong>` and `<b>`?

BLOCK 3: Git - Repository Initialization (12:30-2:20)

Setup (40 min)

- Install Git: https://git-scm.com/downloads Visual Studio Code↗
- Configure: `git config --global user.name "Your Name"`
- Configure: `git config --global user.email "your@email.com"` Visual Studio Code↗
- Read: MIT Missing Semester Version Control (20 min): https://missing.csail.mit.edu/2020/version-control/

Practice (70 min)

bash

```
# Exercise 1: First Repository
mkdir day1-learning
cd day1-learning
git init
git status

# Exercise 2: First Commit
echo "# Day 1 Learning Log" > README.md
git add README.md
git commit -m "Initial commit: Add README"
git log

# Exercise 3: Multiple Files
# Copy your HTML file
git add index.html
git commit -m "Add first HTML page"
git log --oneline
```

Checkpoint: 1) What does `git init` do? 2) Three states of files in Git? 3) What's the staging area? Stack Overflow↗

BLOCK 4: Project - Personal Bio Page (2:40-4:30)

Requirements:

- [ ] HTML5 structure (DOCTYPE, html, head, body)
- [ ] Title tag with your name
- [ ] H1 heading with your name
- [ ] About section (2-3 paragraphs)
- [ ] Skills list (unordered, 5+ items)
- [ ] Goals list (ordered, 3-5 items)
- [ ] Footer with date

Build (90 min)

1. Planning (10 min): Sketch layout
2. Structure (20 min): HTML skeleton
3. Content (30 min): Add all text
4. Testing (10 min): Validate at validator.w3.org
5. Git (20 min): Commit after each section

BLOCK 5: LeetCode (5:00-6:50)

Problem 1: Two Sum (#1) - 60 min

javascript

```javascript
// Brute force approach
function twoSum(nums, target) {
    for (let i = 0; i < nums.length; i++) {
        for (let j = i + 1; j < nums.length; j++) {
            if (nums[i] + nums[j] === target) {
                return [i, j];
            }
        }
    }
}

// Hash map approach (optimal)
function twoSum(nums, target) {
    const map = {};
    for (let i = 0; i < nums.length; i++) {
        const complement = target - nums[i];
        if (complement in map) {
            return [map[complement], i];
        }
        map[nums[i]] = i;
    }
}
```

Problem 2: Contains Duplicate (#217) - 30 min

javascript

```javascript
function containsDuplicate(nums) {
    const seen = new Set();
    for (let num of nums) {
        if (seen.has(num)) return true;
        seen.add(num);
    }
    return false;
}
```

Reflection (20 min): Write notes explaining both approaches, time complexity

---

BLOCK 6: Review & Planning (7:10-8:30)

Review (40 min)

- Recite data types from memory
- Explain == vs === in own words
- Draw HTML structure from memory
- Explain Git staging area

Daily Log (20 min)

Study Time: ___ hours | Energy: ___/10 | Mood: ___/10
Completed: ☐ JS Ch1 ☐ HTML page ☐ Git init ☐ Bio page ☐ 2 LeetCode
Struggles: _____
Wins: _____
Tomorrow: Variables, text formatting, commit workflow

Planning (20 min): Preview Day 2 topics, set specific goals

DAY 2: VARIABLES & CONTROL FLOW

BLOCK 1: JavaScript - Variables (8:00-9:50)

Reading (60 min)

- JavaScript.info Variables: https://javascript.info/variables
- Eloquent JavaScript Ch2 Part 1: https://eloquentjavascript.net/02_program_structure.html

Key Focus: let vs const vs var, block scope, if/else W3Schools +2↗

5 Critical Scenarios:

javascript

```javascript
// 1. Constants for config
const API_KEY = "abc123";
const MAX_USERS = 100;

// 2. Loop counters
for (let i = 0; i < 5; i++) {
    console.log(i);
}

// 3. Function vs block scope
function test() {
    if (true) {
        var x = 5;  // Function-scoped
        let y = 10; // Block-scoped
    }
    console.log(x); // 5
    // console.log(y); // Error
}

// 4. Objects with const
const person = { name: "John", age: 30 };
person.age = 31;    // ✅ Allowed
person.city = "NYC"; // ✅ Allowed
// person = {};     // ❌ Error

// 5. Hoisting
console.log(a); // undefined
// console.log(b); // Error
var a = 5;
let b = 10;
```

Practice (40 min):

- Calculator with if/else
- Age category program
- Grade calculator (A-F)

Checkpoint: When to use const vs let? Can you modify const object properties? What's hoisting? GeeksforGeeks↗ freeCodeCamp↗

BLOCK 2: HTML - Text & Links (10:10-12:00)

Reading (30 min)

- MDN HTML Text Fundamentals: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals
- MDN Creating Hyperlinks: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks

Key Elements: `<strong>` vs `<b>`, `<em>` vs `<i>`, `<ul>`, `<ol>`, `<a>`

Practice (60 min):

html

```html
<nav>
  <ul>
    <li><a href="#about">About</a></li>
    <li><a href="#skills">Skills</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>
</nav>

<section id="about">
  <h2>About Me</h2>
  <p>I'm learning <strong>full-stack development</strong>.</p>
  <p>Check out my <a href="https://github.com/username" target="_blank">GitHub</a>.</p>
</section>
```

BLOCK 3: Git - Commit Workflow (12:30-2:20)

Reading (20 min): Good Commit Messages: https://chris.beams.io/posts/git-commit/

Commands:

bash

```bash
git add <file>         # Stage specific
git add .              # Stage all
git commit -m "msg"    # Commit
git log                # View history
git log --oneline      # Compact
git diff               # Unstaged changes
git diff --staged      # Staged changes
```

Practice (90 min): Make 10 commits on bio page, practice different log formats Visual Studio Code↗

BLOCK 4: Project - Recipe Page (2:40-4:30)

Requirements:

- Recipe name (h1)
- Prep/cook time list
- Ingredients (unordered list)
- Instructions (ordered list)
- Tips section
- External link to source
- At least 5 Git commits

BLOCK 5: LeetCode (5:00-6:50)

Problem 3: Valid Anagram (#242) - 40 min **Problem 4: Reverse String (#344)** - 30 min **Reflection**: 20 min

---

DAY 3: LOOPS & CSS INTRODUCTION

BLOCK 1: JavaScript - Loops (8:00-9:50)

Reading (60 min)

- Eloquent JavaScript Ch2 Part 2: https://eloquentjavascript.net/02_program_structure.html

Key Concepts: while, do-while, for loops, break, continue, ++/+= Eloquent JavaScript +2↗

Practice (40 min):


javascript

```
// FizzBuzz
for (let i = 1; i <= 100; i++) {
  if (i % 15 === 0) console.log("FizzBuzz");
  else if (i % 3 === 0) console.log("Fizz");
  else if (i % 5 === 0) console.log("Buzz");
  else console.log(i);
}

// Triangle pattern
for (let i = 1; i <= 7; i++) {
  console.log("#".repeat(i));
}
```

Checkpoint: while vs do-while? Explain for loop parts? What does continue do?

BLOCK 2: CSS Introduction (10:10-12:00)

Reading (30 min)

- MDN CSS Basics: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics mozilla↗
- MDN Getting Started: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Getting_started mozilla↗

Key Concepts: CSS syntax, selectors (element, class, ID), linking stylesheets, basic properties mozilla↗

Practice (70 min):

CSS

```css
/* style.css */
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background-color: #f4f4f4;
}

h1 {
    color: #333;
    border-bottom: 3px solid #0066cc;
}

.highlight {
    background-color: yellow;
    padding: 5px;
}

#main-content {
    background: white;
    padding: 20px;
    border-radius: 5px;
}
```

Link to HTML: `<link href="style.css" rel="stylesheet">`

BLOCK 3: Git - History & Branching (12:30-2:20)

Commands:

bash

```
git log --graph --all --decorate --oneline
git show <commit-hash>
git diff HEAD~1 HEAD
git branch <name>
git checkout <name>
git checkout -b <name>  # Create and switch
```

Practice: Create feature branch, make changes, view history

BLOCK 4: Project - Style Previous Pages (2:40-4:30)

Add CSS to bio page and recipe page:

- Color scheme (3-4 colors)
- Typography (font-family, sizes)
- Spacing (margins, padding)
- Layout improvements

BLOCK 5: LeetCode (5:00-6:50)

Problem 5: Palindrome Number (#9) - 30 min **Problem 6: Fizz Buzz (#412)** - 20 min **Review both approaches**: 60 min

---

DAY 4: FUNCTIONS & CSS BOX MODEL

BLOCK 1: JavaScript - Functions (8:00-9:50)

Reading (90 min)

- Eloquent JavaScript Ch3: https://eloquentjavascript.net/03_functions.html

Key Concepts: Function declarations, expressions, parameters, return, scope, closure, arrow functions Eloquent JavaScript +3↗

Practice (50 min):

javascript

```javascript
// Function declaration
function greet(name) {
  return `Hello, ${name}!`;
}

// Function expression
const square = function(n) {
  return n * n;
};

// Arrow function
const double = n => n * 2;

// Closure example
function makeCounter() {
  let count = 0;
  return function() {
    count++;
    return count;
  };
}
```

Checkpoint: Declaration vs expression? What's closure? Write arrow function?

BLOCK 2: CSS Box Model (10:10-12:00)

Reading (40 min)

- MDN Box Model: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Box_model mozilla ↗
- MDN Basic Selectors: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Basic_selectors mozilla ↗

Key Concepts: content, padding, border, margin, box-sizing, block vs inline mozilla ↗

Practice (70 min):

CSS

```css
.card {
    width: 300px;
    padding: 20px;
    margin: 20px;
    border: 2px solid #ddd;
    box-sizing: border-box;
}

.highlight:hover {
    background-color: #ffeb3b;
    cursor: pointer;
}
```

BLOCK 3: Git - Merging (12:30-2:20)

Commands:

bash

```
git checkout main
git merge <branch>
git merge --no-ff <branch>
git branch -d <branch>
```

Practice: Create feature branch, merge back, handle conflicts

---

BLOCK 4: Project - Landing Page Start (2:40-4:30)

Begin multi-section landing page:

- Header with navigation
- Hero section
- Features section (3 cards)
- Use box model for spacing

BLOCK 5: LeetCode (5:00-6:50)

Problem 7: Running Sum (#1480) - 30 min **Problem 8: Richest Customer Wealth (#1672)** - 30 min **2D arrays practice**: 50 min

---

DAY 5: ARROW FUNCTIONS & FLEXBOX

BLOCK 1: JavaScript - Arrow Functions & Templates (8:00-9:50)

Reading (60 min)

- JavaScript.info Arrow Functions: https://javascript.info/arrow-functions-basics
- Template Literals tutorial

Practice (50 min):


javascript

```javascript
// Arrow function variations
const greet = name => `Hello, ${name}!`;
const add = (a, b) => a + b;
const calculate = (a, b) => {
  const sum = a + b;
  const product = a * b;
  return { sum, product };
};

// Template literals
const name = "Alice";
const age = 25;
const message = `My name is ${name} and I'm ${age} years old.`;

// Multi-line strings
const html = `
<div>
  <h1>${name}</h1>
  <p>Age: ${age}</p>
</div>
`;
```

Checkpoint: When omit parentheses? Template literal expressions? Convert regular to arrow function? JavaScript.info +4↗

BLOCK 2: CSS Flexbox (10:10-12:00)

Reading (40 min)

- MDN Flexbox: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

- CSS-Tricks Flexbox Guide: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Key Concepts: display: flex, flex-direction, justify-content, align-items, flex-wrap, gap

Practice (70 min):

CSS

```css
.nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem;
  background: #333;
}

.cards {
  display: flex;
  gap: 20px;
  flex-wrap: wrap;
}

.card {
  flex: 1;
  min-width: 250px;
  padding: 20px;
}

.center {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
```

BLOCK 3: Git - Remote Repositories (12:30-2:20)

Setup GitHub (30 min): Create account, SSH keys Visual Studio Code↗ Adam the Automator↗

Commands:

bash

```
git remote add origin <url>
git push -u origin main
git pull origin main
git fetch
git clone <url>
```

Practice (90 min): Push project to GitHub, clone another repo Visual Studio Code↗Stack Overflow↗

BLOCK 4: Project - Landing Page Completion (2:40-4:30)

Complete landing page with Flexbox:

- Navigation bar (horizontal flex)
- Hero section (centered content)
- Feature cards (flex row, wrap)
- Footer
- Push to GitHub

BLOCK 5: LeetCode (5:00-6:50)

Problem 9: Roman to Integer (#13) - 40 min **Problem 10: Shuffle Array (#1470)** - 30 min **Hash map practice**: 40 min

---

DAY 6: ARRAYS, OBJECTS & CALCULATOR

BLOCK 1: JavaScript - Arrays & Objects (8:00-9:50)

Reading (90 min)

- Eloquent JavaScript Ch4 Part 1: https://eloquentjavascript.net/04_data.html

Key Concepts: Arrays (push, pop, shift, unshift, slice, indexOf), objects (properties, methods), dot vs bracket notation MDN Web Docs +3↗

Practice (50 min):

javascript

```
// Array methods
const numbers = [1, 2, 3, 4, 5];
numbers.push(6);
numbers.unshift(0);
const sliced = numbers.slice(2, 4);

// Objects
const person = {
    name: "John",
    age: 30,
    greet: function() {
        return `Hello, I'm ${this.name}`;
    }
};

// Array of objects
const students = [
    { name: "Alice", grade: 95 },
    { name: "Bob", grade: 87 }
];
```

BLOCK 2: CSS Typography & Colors (10:10-12:00)

Reading (30 min)

- MDN Backgrounds and Borders: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Backgrounds_and_borders mozilla ↗

Key Concepts: font-family, font-size, font-weight, text-align, color, background-color, hex/rgb/rgba

Practice (80 min): Style landing page with cohesive typography and color scheme

---

BLOCK 3: Git - Complete Workflow Practice (12:30-2:20)

Practice complete workflow:

bash

```
# Feature branch workflow
git checkout -b add-feature
# Make changes
git add .
git commit -m "Add new feature"
git checkout main
git merge add-feature
git push origin main
```

Exercise: Make 5 feature branches, merge all

BLOCK 4: Project - Calculator (2:40-4:30)

Requirements:

- Display screen
- Digit buttons (0-9)
- Operation buttons (+, -, ×, ÷)
- Equals, clear buttons
- Event listeners
- Chain calculations
- Error handling

HTML Structure:


html

```
<div class="calculator">
  <div class="display" id="display">0</div>
  <div class="buttons">
    <button class="btn" data-num="7">7</button>
    <!-- ... more buttons ... -->
    <button class="btn operator" data-op="+">+</button>
    <button class="btn equals">=</button>
  </div>
</div>
```

JavaScript:


javascript

```
let currentValue = '';
let previousValue = '';
let operator = '';

document.querySelectorAll('.btn').forEach(btn => {
    btn.addEventListener('click', function() {
        // Handle button click
    });
});
```

BLOCK 5: LeetCode (5:00-6:50)

Problem 11: Jewels and Stones (#771) - 25 min **Problem 12: Valid Palindrome (#125)** - 35 min **String processing review**: 50 min

---

DAY 7: DESTRUCTURING & TO-DO LIST

BLOCK 1: JavaScript - Destructuring (8:00-9:50)

Reading (60 min)

- JavaScript.info Destructuring: https://javascript.info/destructuring-assignment
- Eloquent JavaScript Ch4 Part 2

Practice (50 min):

javascript

```javascript
// Array destructuring
const [first, second, third] = [1, 2, 3];
const [a, , c] = [1, 2, 3]; // Skip element
const [x, ...rest] = [1, 2, 3, 4, 5];

// Object destructuring
const { name, age } = { name: "John", age: 30 };
const { name: userName, age: userAge } = person; // Rename
const { country = "USA" } = person; // Default

// Function parameters
function display({ name, age }) {
  console.log(name, age);
}
display({ name: "Alice", age: 25 });
```

Checkpoint: Swap values with destructuring? Default values? Rename in destructuring? JavaScript.info +2↗

BLOCK 2: Final HTML/CSS Review (10:10-12:00)

Portfolio Page: Create comprehensive portfolio combining all learned concepts:

- Semantic HTML5
- Flexbox layouts
- Typography and colors
- Responsive design
- All previous projects linked

BLOCK 3: Git - Advanced Workflow (12:30-2:20)

Topics: Tagging, stashing, undoing changes

bash

```bash
git stash
git stash pop
git reset HEAD <file>
git checkout -- <file>
git tag -a v1.0 -m "Week 1 complete"
git push origin v1.0
```

BLOCK 4: Project - To-Do List (2:40-4:30)

Requirements:

- Input field for new tasks
- Add button (+ Enter key)
- Task list with checkboxes
- Delete buttons
- localStorage persistence
- Empty state message
- Validation (no empty tasks)

JavaScript Structure:


javascript

```
let todos = [];

function addTodo(text) {
    const todo = {
        id: Date.now(),
        text: text,
        completed: false
    };
    todos.push(todo);
    saveTodos();
    renderTodos();
}

function deleteTodo(id) {
    todos = todos.filter(todo => todo.id !== id);
    saveTodos();
    renderTodos();
}

function toggleComplete(id) {
    const todo = todos.find(t => t.id === id);
    if (todo) todo.completed = !todo.completed;
    saveTodos();
    renderTodos();
}

function saveTodos() {
    localStorage.setItem('todos', JSON.stringify(todos));
}

function loadTodos() {
    const stored = localStorage.getItem('todos');
    if (stored) todos = JSON.parse(stored);
    renderTodos();
}

function renderTodos() {
    // Update DOM
}
```

```
// Initialize
document.addEventListener('DOMContentLoaded', loadTodos);
```

BLOCK 5: LeetCode (5:00-6:50)

Problem 13: Best Time to Buy/Sell Stock (#121) - 45 min **Problem 14: Remove Duplicates (#26)** - 35 min **Week review**: 30 min

---

BLOCK 6: Week 1 Comprehensive Review (7:10-8:30)

Complete Weekly Review Checklist:

- [ ] Rate all concepts (1-5)
- [ ] Calculate total study hours
- [ ] Review all projects
- [ ] Test knowledge without docs
- [ ] Plan Week 2
- [ ] Celebrate wins!

📚 COMPLETE RESOURCE LIBRARY

JavaScript

- **Eloquent JavaScript**: https://eloquentjavascript.net/
- **JavaScript.info**: https://javascript.info/
- **MDN JavaScript**: https://developer.mozilla.org/en-US/docs/Web/JavaScript

HTML/CSS

- **MDN HTML**: https://developer.mozilla.org/en-US/docs/Web/HTML
- **MDN CSS**: https://developer.mozilla.org/en-US/docs/Web/CSS
- **CSS-Tricks Flexbox**: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Git

- **MIT Missing Semester**: https://missing.csail.mit.edu/2020/version-control/
- **Pro Git Book**: https://git-scm.com/book/en/v2
- **Learn Git Branching**: https://learngitbranching.js.org/

LeetCode

- **LeetCode**: https://leetcode.com/problemset/
- Start with Easy problems
- Focus on understanding, not speed

🛠 VS CODE COMPLETE SETUP

Essential Extensions (Install Day 1)

1. **ESLint** - JavaScript error detection visualstudio↗
2. **Prettier** - Auto-formatting visualstudio↗
3. **Live Server** - Local dev server with auto-reload
4. **GitLens** - Git integration
5. **JavaScript (ES6) code snippets** - Code shortcuts Visual Studio Code +4↗

Recommended Extensions (Day 2-3)

6. **Path Intellisense** - File path autocomplete
7. **Auto Rename Tag** - Update paired HTML tags
8. **IntelliSense for CSS** - CSS class autocomplete
9. **Better Comments** - Color-coded comments
10. **REST Client** - API testing GitKraken↗CodeForGeek↗

Settings Configuration (settings.json)

json

```
{
  "editor.fontSize": 14,
  "editor.tabSize": 2,
  "editor.wordWrap": "on",
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.bracketPairColorization.enabled": true,

  "[javascript]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  },

  "prettier.semi": true,
  "prettier.singleQuote": true,
  "prettier.tabWidth": 2,
  "prettier.trailingComma": "es5",

  "eslint.enable": true,
  "files.autoSave": "afterDelay",
  "files.autoSaveDelay": 1000,

  "git.enableSmartCommit": true,
  "git.autofetch": true
}
```

Essential Keyboard Shortcuts

Navigation (Learn Day 1-2):

- `Ctrl+P` (Cmd+P) - Quick open file
- `Ctrl+Shift+P` (Cmd+Shift+P) - Command palette
- `Ctrl+B` (Cmd+B) - Toggle sidebar
- `Ctrl+` ` - Toggle terminal
- `Ctrl+Tab` - Switch between files

Editing (Learn Day 3-4):

- `Alt+Up/Down` (Option+Up/Down) - Move line
- `Shift+Alt+Up/Down` - Copy line
- `Ctrl+/` (Cmd+/) - Toggle comment
- `Ctrl+D` (Cmd+D) - Select next occurrence
- `Ctrl+Shift+L` (Cmd+Shift+L) - Select all occurrences

Debugging (Learn Day 5-7):

- `F5` - Start/continue debugging

- `F9` - Toggle breakpoint
- `F10` - Step over
- `F11` - Step into
- `Shift+F11` - Step out

🎯 DEPRESSION MANAGEMENT INTEGRATION

Morning Routine (Every Day: 6:00-7:30)

Physical Activity (20 min) - Choose one:

- **Brisk walk**: 15-20 min outdoors (boosts serotonin, vitamin D)
- **Yoga**: 10-15 min sun salutations (reduces cortisol)
- **Cardio**: 15 min bodyweight exercises (endorphins)

Mindfulness (15 min):

- 10 min guided meditation (Headspace, Calm)
- 5 min journaling intentions

Breakfast (30 min):

- High-protein (eggs, Greek yogurt, nuts)
- Complex carbs (oatmeal, whole grain)
- Hydrate: 16oz water
- **Avoid**: Sugar, energy drinks

Setup (25 min):

- Review yesterday's wins
- Check today's goals
- Workspace preparation
- Mental state check (rate 1-10)

Break Protocol (Every 50-60 minutes)

DO:

- Stand and stretch
- Walk around (inside or outside)
- 10 push-ups or squats
- Look at distant objects (20-20-20 rule)
- Drink water (8oz)
- Deep breathing (4-7-8 technique)

DON'T:

- Stay seated
- Check social media
- Watch TV/YouTube
- Start another task
- Skip the break

Red Flags & Actions

Warning Sign	Severity	Action
Suicidal thoughts	🔴 CRITICAL	Stop immediately. Call 988 (US) or crisis line
Can't get out of bed	🔴 SEVERE	Take rest day. Talk to someone
Panic attacks	🔴 SEVERE	Stop studying. Deep breathing. Seek help
Mood ≤3/10 for 3+ days	🟠 HIGH	Reduce to 6-hour days. Add self-care
Sleep ≤5 hours (2+ nights)	🟠 HIGH	Enforce 9pm screens off. Consider rest day
Difficulty concentrating (20+ min)	🟠 MEDIUM	Take extended break. Switch topics
Skipping meals (3+ times/week)	🟡 MODERATE	Set meal alarms. Prep meals ahead
Occasional frustration	🟢 NORMAL	Take break. This is expected!

Evening Wind-Down (8:30-10:00)

Phase 1: Disengage (8:30-9:00)

- Close all coding tools by 8:30
- Light activity (dishes, walk)
- Process day emotionally

Phase 2: Relax (9:00-9:30)

- Shower/bath
- Non-coding reading
- Calming music
- Stretching

Phase 3: Sleep Prep (9:30-10:00)

- Dim lights, blue light filter
- 10-min meditation
- Journal 3 gratitudes
- Lights out by 10pm

Sleep Hygiene:

- Room temp 65-68°F
- Complete darkness
- No caffeine after 2pm
- 8 hours target

🤖 AI TOOL USAGE GUIDELINES

5 GOOD Prompts (Promote Learning)

1. Concept Explanation:

"Explain JavaScript's 'this' keyword with 3 simple examples, starting basic. Don't give complex scenarios yet."

2. Debugging Guidance (not solutions):

"My for loop isn't iterating correctly. What are the 3 most common mistakes beginners make with for loops that I should check?"

3. Concept Verification:

"I think const means value never changes, but I just modified a const object's property. Can you explain what's happening?"

4. Alternative Approaches:

"I solved Two Sum with nested loops. Are there other approaches for similar problems? Don't show code, just describe concepts."

5. Study Guidance:

"I have 2 hours for JavaScript practice. What topics from Eloquent JavaScript Ch2 should I prioritize if I'm shaky on conditionals?"

5 BAD Prompts (Create Dependency)

1. Direct Solutions:

❌ "Solve LeetCode Two Sum for me"
Why bad: Learn nothing, defeats purpose

2. Complete Code Generation:

❌ "Write a complete calculator app with all functions"
Why bad: Won't understand code you copy

3. Homework Completion:

❌ "Complete my Day 3 project and give me the HTML"
Why bad: Cheating yourself of learning

4. No-Context Fixes:

❌ "Why isn't this working? [paste code]"
Why bad: Miss debugging learning process

5. Vague Requests:

❌ "Teach me JavaScript"
Why bad: Too broad, need specific targeted learning

When to Use Which Tool

Perplexity - Best for:

- Current information, best practices
- Finding resources
- Comparing approaches
- Recent framework updates

ChatGPT - Best for:

- Conversational explanations
- Breaking down complex topics
- Practice exercise generation
- Interactive Q&A

Claude - Best for:

- Long-form explanations
- Structured learning plans
- Detailed code review
- Multi-step reasoning

Daily AI Limits & Tracking

Rules:

- Maximum 30 min per 4-hour block
- Must try independently 30 min first
- Log every interaction
- No copying code directly

Daily AI Log:

Time: _____ | Tool: _____ | Topic: _____
Tried first? Y/N (duration: ___)
Query: _____
Helpful? Y/N
What I learned: _____

📊 PROGRESS TRACKING SYSTEM

Daily Log Template

markdown

DAY ___ LEARNING LOG
Date: _____ | Day of Week: _____

TIME & ENERGY
Total Study Time: _____ hours (target: 10)
Start Time: _____ | End Time: _____
Morning Energy (1-10): _____
Evening Energy (1-10): _____
Overall Mood (1-10): _____
Sleep Last Night: _____ hours

BLOCKS COMPLETED
- [ ] Block 1: JavaScript (110 min)
- [ ] Block 2: HTML/CSS (110 min)
- [ ] Block 3: Git (110 min)
- [ ] Block 4: Project (110 min)
- [ ] Block 5: LeetCode (110 min)
- [ ] Block 6: Review (80 min)

CONCEPTS LEARNED
1. _____
2. _____
3. _____
4. _____
5. _____

UNDERSTANDING RATINGS (1-5)
JavaScript concepts today: _____/5
HTML/CSS concepts today: _____/5
Git concepts today: _____/5
Overall confidence: _____/5

Rate 3 or below = review tomorrow

WORK COMPLETED
JavaScript exercises: _____ / _____
HTML/CSS exercises: _____ / _____
Git commands practiced: _____ / _____
LeetCode problems: _____ / 2
Project progress: _____%

GIT ACTIVITY
Commits today: _____
Branches created: _____
Pushed to GitHub: Y/N

CHALLENGES & WINS
Struggled with:
1. _____
2. _____

Breakthrough moments:
1. _____
2. _____

Today's wins:
1. _____
2. _____

AI TOOL USAGE
Total time: _____ min (limit: 60)
Times consulted: _____
Most helpful for: _____
Tried solving independently first? Y/N
Time spent before asking: _____ min

SELF-CARE CHECKLIST
- [ ] Morning routine (90 min)
- [ ] Physical activity 3+ times
- [ ] All scheduled breaks taken
- [ ] 8+ cups water
- [ ] 3 healthy meals
- [ ] No screens after 9pm
- [ ] Evening wind-down completed
- [ ] In bed by 10pm

TOMORROW'S PLAN
Focus topics:
1. _____
2. _____
3. _____

Specific goals:

- [ ] _____
- [ ] _____
- [ ] _____

Weekly Review (End of Week 1)

markdown

WEEK 1 COMPREHENSIVE REVIEW

QUANTITATIVE METRICS
Total study hours: _____ / 70 target
Days completed: _____ / 7
JavaScript exercises: _____ / 20
HTML/CSS exercises: _____ / 10
Git exercises: _____ / 10
LeetCode problems: _____ / 14 (target: 5+ minimum)
Projects completed: _____ / 5
Total Git commits: _____
Estimated lines of code: _____

KNOWLEDGE ASSESSMENT (Rate 1-5)

JavaScript
- [ ] Data types & typeof: _____/5
- [ ] Variables (let/const/var): _____/5
- [ ] Operators & comparisons: _____/5
- [ ] Conditionals (if/else): _____/5
- [ ] Loops (for/while): _____/5
- [ ] Functions: _____/5
- [ ] Arrow functions: _____/5
- [ ] Template literals: _____/5
- [ ] Arrays basics: _____/5
- [ ] Objects basics: _____/5
- [ ] Destructuring: _____/5
- [ ] DOM manipulation: _____/5

Average JS Score: _____ /5

HTML/CSS
- [ ] HTML5 structure: _____/5
- [ ] Semantic elements: _____/5
- [ ] Forms: _____/5
- [ ] CSS selectors: _____/5
- [ ] Box model: _____/5
- [ ] Flexbox: _____/5
- [ ] Typography/colors: _____/5

Average HTML/CSS Score: _____ /5

Git
- [ ] Init & config: _____/5
- [ ] Staging/committing: _____/5
- [ ] Viewing history: _____/5
- [ ] Branching: _____/5
- [ ] Merging: _____/5
- [ ] Remote (GitHub): _____/5

Average Git Score: _____ /5

PRACTICAL SKILLS (Can do without docs?)
- [ ] Create HTML5 page from memory
- [ ] Explain let vs const with examples
- [ ] Write for loop counting 1-10
- [ ] Create object, access properties
- [ ] Arrow function with template literals
- [ ] Style div with box model
- [ ] Git: init, commit, push workflow
- [ ] Create branch, merge

Skills mastered: _____ / 8

MILESTONE VALIDATION

✅ VS Code Setup
- [ ] ESLint working
- [ ] Prettier formatting on save
- [ ] GitLens showing info
- [ ] Live Server launching
- [ ] 5+ shortcuts memorized

✅ GitHub Account
- [ ] Account created
- [ ] First repo pushed
- [ ] 30+ total commits
- [ ] README written
- [ ] Can push/pull without looking up

✅ JavaScript ES6+
- [ ] Explain let vs const without notes

- [ ] Written 3+ arrow functions
- [ ] Used template literals 5+ times
- [ ] Destructured arrays/objects
- [ ] Comfortable with syntax

✅ HTML/CSS Pages
- [ ] Built 3+ complete pages
- [ ] Semantic HTML used
- [ ] Flexbox for layouts
- [ ] Responsive basics
- [ ] Validated (no errors)

✅ LeetCode
- [ ] Solved 5+ Easy independently
- [ ] Can explain approaches
- [ ] Understand time complexity basics
- [ ] Tried multiple approaches

✅ Daily Routine
- [ ] 10-12 hour days (5+ days)
- [ ] Morning routine (5+ days)
- [ ] All breaks taken (5+ days)
- [ ] Evening wind-down (5+ days)
- [ ] 7-8 hours sleep (5+ nights)

Milestones achieved: _____ / 6

WELLBEING ASSESSMENT

Physical Health
- Average energy: _____/10
- Sleep quality: _____/10
- Physical activity days: _____ /7
- Meal consistency: _____/10

Mental Health
- Average mood: _____/10
- Stress level: _____/10
- Motivation: _____/10
- Difficult days: _____

Red Flags Observed
- [ ] Low energy ≤4/10 (3+ days)
- [ ] Sleep issues ≤5 hrs regularly
- [ ] Skipped meals 3+ times
- [ ] Isolated 2+ days
- [ ] Severe frustration
- [ ] Physical symptoms

Red flags: _____ (0 is ideal)

WINS & CHALLENGES

5 Major Wins:
1. _____
2. _____
3. _____
4. _____
5. _____

3 Biggest Challenges:
1. _____
2. _____
3. _____

What worked well:
_____

What needs improvement:
_____

WEEK 2 ADJUSTMENTS

Schedule changes:
_____

Focus areas:
_____

Self-care improvements:
_____

WEEK 2 GOALS
1. _____
2. _____
3. _____
4. _____
5. _____

OVERALL GRADE: _____ / 100

Grading:
- Study hours (20 pts): 60+ = 20, 50-59 = 15, 40-49 = 10
- Concepts (20 pts): Avg 4+ = 20, 3-3.9 = 15, 2-2.9 = 10
- Projects (20 pts): 5 = 20, 4 = 16, 3 = 12
- LeetCode (20 pts): 10+ = 20, 7-9 = 15, 5-6 = 10
- Consistency (20 pts): 6-7 days = 20, 4-5 = 15, 3 = 10

READY FOR WEEK 2?
- [ ] Yes, confident
- [ ] Yes, but need review
- [ ] Unsure, significant struggles
- [ ] No, repeat Week 1

Review before Week 2:

_____

🎯 WEEK 1 SUCCESS METRICS

Minimum Goals (Must Achieve)

- ✅ 40+ study hours
- ✅ VS Code + 5 extensions
- ✅ GitHub + 1 repo pushed
- ✅ 2 projects completed
- ✅ 5 LeetCode solved
- ✅ Basic HTML/CSS/JS working

Standard Goals (Target)

- ✅ 60+ study hours
- ✅ VS Code fully configured
- ✅ 20+ Git commits
- ✅ 3-4 projects completed
- ✅ 8-10 LeetCode solved

- ✅ Fundamentals comfortable (3.5/5)

Stretch Goals (Exceptional)

- ✅ 70+ study hours
- ✅ All 5 projects + bonuses
- ✅ 12+ LeetCode solved
- ✅ Blog post written
- ✅ Helped another learner
- ✅ Open source contribution

🚨 EMERGENCY PROTOCOLS

Stuck on Problem (30+ min)

Step 1 (5 min): Break, walk away
Step 2 (10 min): Rubber duck - explain aloud
Step 3 (10 min): Google error message
Step 4 (15 min): Check docs (MDN, JavaScript.info)
Step 5 (20 min): Ask AI with good prompt
Step 6: Post on Discord/Reddit with attempts

Never spin wheels 2+ hours in Week 1

Feeling Like Quitting

Immediate (5 min):

1. Stand, walk outside
2. Drink water
3. 10 deep breaths
4. Text someone

Short-term (1 hour):

1. Review wins list
2. Read Day 1 journal
3. Watch inspiration video
4. Extended break

Medium-term (rest of day):

1. Reduce to minimum goals
2. Self-care activity
3. Journal why you started
4. Sleep on it

Long-term:

- Rest day OK
- Week 1 hardest
- Imposter syndrome normal
- Everyone struggles

Crisis Resources:

- Suicide Prevention: 988 (US)
- Crisis Text: HOME to 741741
- Better Help, Talk Space
- Local mental health

✅ FINAL WEEK 1 CHECKLIST

Setup

- [ ] VS Code installed
- [ ] Git configured
- [ ] GitHub account
- [ ] 5+ extensions
- [ ] settings.json configured
- [ ] Live Server tested

Resources Bookmarked

- [ ] Eloquent JavaScript
- [ ] JavaScript.info
- [ ] MDN Web Docs
- [ ] LeetCode account
- [ ] Git documentation
- [ ] Stack Overflow

Skills Demonstrated

- [ ] 3+ web pages built
- [ ] 500+ lines JavaScript
- [ ] 30+ Git commits
- [ ] 5+ LeetCode solved
- [ ] ES6+ features used
- [ ] Flexbox layouts
- [ ] DOM manipulation

Habits Established

- [ ] Wake same time daily
- [ ] Morning routine consistent
- [ ] Scheduled breaks taken
- [ ] Evening wind-down
- [ ] Daily logging
- [ ] Healthy AI usage

Mental Health

- [ ] No unaddressed red flags

- [ ] Talked about experience
- [ ] Balanced study/self-care
- [ ] Slept 7-8 hours mostly
- [ ] Physical activity maintained

Ready for Week 2

- [ ] Week 1 concepts reviewed
- [ ] Identified topics to practice
- [ ] Week 2 goals set
- [ ] Motivated/determined
- [ ] Support system in place

🎓 CONCLUSION

Week 1 is the hardest. Your brain is forming new neural pathways, learning a completely new language, and everything feels overwhelming. **This is normal.**

By following this guide hour-by-hour, you've eliminated ambiguity and decision fatigue. You know exactly what to do each morning.

Remember:

- **Consistency beats intensity**
- **Understanding beats memorization**
- **Self-care enables performance**
- **Progress compounds daily**
- **You don't need perfection**

Week 2 Preview

Higher-order functions, advanced array methods (map, filter, reduce), more complex projects, API integration basics, Node.js introduction.

For Now

Focus on Day 1. Just Day 1. Wake up, follow the schedule, celebrate small wins.

You're not just learning to code. You're becoming a developer.

📚 COMPLETE URL REFERENCE

JavaScript

- **Eloquent JavaScript**: https://eloquentjavascript.net/
  - Ch1: https://eloquentjavascript.net/01_values.html
  - Ch2: https://eloquentjavascript.net/02_program_structure.html
  - Ch3: https://eloquentjavascript.net/03_functions.html
  - Ch4: https://eloquentjavascript.net/04_data.html
- **JavaScript.info**: https://javascript.info/

- Variables: https://javascript.info/variables
- Data Types: https://javascript.info/types
- Arrow Functions: https://javascript.info/arrow-functions-basics
- Destructuring: https://javascript.info/destructuring-assignment

HTML/CSS

- **MDN HTML Intro**: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML
- **MDN CSS Basics**: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics
- **MDN Box Model**: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Box_model
- **MDN Flexbox**: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
- **CSS-Tricks Flexbox**: https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Git

- **MIT Missing Semester**: https://missing.csail.mit.edu/2020/version-control/
- **Official Git Docs**: https://git-scm.com/doc
- **Pro Git Book**: https://git-scm.com/book/en/v2
- **Learn Git Branching**: https://learngitbranching.js.org/

Tools

- **VS Code Docs**: https://code.visualstudio.com/docs
- **VS Code JavaScript**: https://code.visualstudio.com/docs/languages/javascript
- **LeetCode**: https://leetcode.com/problemset/
- **GitHub Guides**: https://guides.github.com/

LET'S BUILD. 🚀

Research compiled from 15+ sources including bootcamp experiences, learning science, mental health resources, and 2024-2025 development best practices. All resources verified current.