

# CFD

 Follow @derekknex

Chapters

+

- [Preface](#)  
[Coding for](#)  
[Designers](#)
- [Chapter 1](#)  
[Breaking](#)  
[Barriers](#)

- [Ones and Zeros](#)
- [Hard to Soft](#)
- [Bits and Bytes](#)
- [Black and White](#)
- [Coding Color](#)
- [Encode and Decode](#)
- [Saved Image](#)

- **Chapter 2**  
**Structure,**  
**Style, &**  
**Behavior**

- [Structure](#)
- [Style](#)
- [Behavior](#)

- **Chapter 3**  
**Programming &**  
**Visual Design**

- [Design](#)
- [Elements and Elements](#)
- [Principles and Patterns](#)
- [Constructs and Components](#)

- **Chapter 4**  
**Interactive**  
**Code**

- [Authoring, Compiling, and Executing](#)
- [Frame Rate](#)
- [Event Loop](#)
- [Sync and Async](#)
- [Interfacing](#)
- [Client and Server](#)
- [Anatomy of HTML, CSS, and JavaScript](#)
- [Work. Right. Better.](#)

- **Chapter 5**  
**80/20**  
**JavaScript**

- [Environment](#)
- [Mindset](#)
- [Subset](#)
- [Keywords](#)
- [Expressions](#)
- [Operators](#)
- [Statements](#)
- [Functions](#)
- [Errors](#)

- ## Chapter 6

# Deconstructing

# Designs

- [Process](#)
- [Deconstruct](#)
- [Reconstruct Structure](#)
- [Reconstruct Style](#)
- [Reconstruct Behavior](#)

Light

-

## **Preface**

# **Coding for**

# **Designers**

V

irtually all resources aimed at teaching people *how-to* code introduce the code itself too soon. In doing so, many fundamental and empowering concepts remain hidden and out of reach. These resources often lack explanations of *why* a coding concept exists as it does. Additionally, these resources rarely consider designers directly. *Coding for Designers* is the response to this unfortunate reality.

 [Coding for Designers Book](#)  
[Coding for Designers Book](#)

#

## The Audience

If you self-identify as an illustrator or graphic, visual, UI, UX, motion, mobile, web, or game designer then this is for you. Other non-coders will benefit too, but self-identifying designers are the focus. This book—the one I wish existed when I began coding—is also for an earlier time-slice of myself. He will not reap the benefits of its existence, but you will.

Many claim that designers *need* to learn to code. I disagree. Yes, learning to read and author code is a powerful ability, but so too is visually manifesting ideas and information. I want only to empower you further.

In reading this book, you will:

- Form a mental model for learning coding concepts faster and more deeply
- Improve your ability to read, understand, and author code
- Become a better asset to yourself, your team, and potential employers
- Increase confidence for engaging in—not shying from—coding conversations
- Gain the power to see and design digital creations through a more capable lens

This book is not for you if you want to jump straight into code though. We cover many ideas, topics, and concepts before diving into actual programming code. This intentionality helps frame why certain programming concepts exist as they do. It also encourages a beneficial mindset with accompanying techniques for reading, navigating, and understanding code.

#

## The Goals

The goals of this book are:

1. Instill a mental model that helps designers learn coding concepts faster, more deeply, and more intuitively
2. Lower the barrier of entry into JavaScript

The former enables the latter. Neither goal is efficiently achievable with the traditional approach.

#

## The Approach

The plan of attack is simple. In chapter one, *[Breaking Barriers](#)*, we will demystify a series of concepts that intimidate most designers and non-coders. From the seemingly magical ones and zeros, to the code you will soon write, we will see simple, repetitive, and reusable concepts at play.

In chapter two, *[Structure, Style, and Behavior](#)*, we will cover the three codeable features that enable computers to render our dynamic and interactive creations. We will do so from a 2D context with the web platform and a 3D context with the [Unity](#)<sup>®</sup> platform. These perspectives will solidify the distinct role that structure, style, and behavior play regardless of dimensionality.

Chapter three, *[Programming and Visual Design](#)*, will cover the Elements, Principles, and Constructs of Visual Design relative to programming's counterparts. This work will help shape a mental model tuned specifically for designers.

In chapter four, *[Interactive Code](#)*, we will introduce many concepts that are invaluable when reading and authoring code. We'll explore many topics including frame rate, the event loop, interfaces, and code anatomy. In exploring them you will further improve and shape your mental model.

Chapter five, *[80/20 JavaScript](#)*, will reinforce your learning in the context of JavaScript. We'll take an optimized approach to enable your productivity. This will empower you to author JavaScript without knowing the entire language and all its idiosyncrasies.

Finally in chapter six, *[Deconstructing Designs](#)*, we will put everything we have learned to practice. We will do so by analyzing and deconstructing Twitter's Tweet Box user interface. We will then reconstruct a majority of it to learn how to design and implement a popular and non-trivial interactive application.

Let's do this.

Powered by Typeform



# Chapter 1

## Breaking

### Barriers