

Conjuring Code

WORKING DRAFT

Ian Varley

April 10, 2026

This is a working draft of the manuscript.

For the most up-to-date drafts, visit conjuringcode.com/drafts

Contents

Preface - Start Now	11
-------------------------------	----

Part 1: Get Your Bearings

Chapter 1: You've Got Questions	13
--	-----------

What Is This Book?	13
A Surprise Gift • Nobody expected AI to be able to write code, but it can.	14
There's A Catch • Even with AI, making software still requires thinking.	16
Is This ... Real? • Yes. AI writes code as well as people now (or better).	17
Is This Book For Me?	19
Beginners Welcome • There's no special knowledge required for entry.	19
Non-Beginners Welcome, Too • This is excellent if you're tech-adjacent.	21
You're Not Too Far Behind • Don't worry, you really can do it.	21
AI To Make It, Not AI Inside It • We're talking about regular old software.	22
Why Should I Care? • Making software is empowering, useful, and fun.	23
Get More Of What You Want • There's a you-shaped hole in software.	24
Control Your Destiny • Cast, or be cast upon.	25
Help Other People • Don't be selfish!	25
Play For Team Human • We gotta stick together.	26
Make Money? • Nothing wrong with making a little scratch.	27
Why Should I Listen To You? • I'm a professional (but don't hold it against me).	29
How Do I Use This Book? • Engage actively, not passively.	30
The Seven Themes • Seven key messages to pay attention to.	31
Play Along At Home • Use the companion website to go deeper.	31
Don't Wait • The best time to plant a software is now.	32

Chapter 2: You Can Make Your Own Software Now	34
--	-----------

Software Is Awesome	34
It's Magic • Capable of anything.	34
It's Fast • Fast and capable of many things.	35
It's Cheap • You can do a lot for a little.	35
It's Correct • It doesn't roll dice or make things up.	37
It's Reusable • You don't have to reinvent the wheel each time.	38
It's Malleable • You can change it from anything to anything.	39

It's Accretive • Your investment keeps adding up.	40
It's Stackable • You can build big things out of small things.	40
It's Shareable • Copy it as often as you like.	41
It's Egalitarian • It doesn't just help the billionaires.	42
It's Everywhere • The modern world can't exist without it.	42
But Software Is Too Hard • Most people can't do it.	43
It's Cryptic • Terse, weird commands that look like gibberish.	44
It's Arbitrary • You just have to memorize how to do things.	46
It's Fragmented • dozens of different, incompatible languages	47
It's Complicated • The modern software stack is very complex.	48
It's Slow • It takes a long time to make even simple software.	49
It's Fragile • Changing one thing breaks two other things.	49
It's Constantly Changing • You can't keep up with the latest stuff.	50
It's An Exclusive Club • You need to have a high tolerance for all this.	50
And That Is Why Most Software Sucks • It's hard for no good reason.	51
This Isn't What We Expected To Happen • The cognitive burden caught us off guard.	52
AI Changes The Equation • Many things that sucked now suck less.	53
It Removes The Bad Hard • The annoying, cryptic, stupid parts.	53
You Move To Higher Level Work • Climb the abstraction ladder!	55
It's A Really Good Coder • It makes software with all the fit and finish.	59
It's Unbelievably Fast • Quantity is a quality all its own.	59
You Can't Trust AI ... • It makes mistakes all the time!	60
... But You Don't Need To! • Software is already made by fallible beings!	61
Software Is The Best Use Of AI • Lots of the good, little of the bad	63
But You Are Still The Driver • You can't just close your eyes and pray it's right.	65
And It's Still Fun • In fact, it's funner than it's been in a long time.	67
So What Will You Build? • The sky's the limit.	68
Build Something You Need • Make software for yourself	68
Build Something For Work • You (probably) can and (definitely) should.	69
Build Something Fun • Not everything needs a purpose.	70
Add To Something Else • Doesn't have to be just yours!	71
Get More Ideas • decrease friction, automate things, have more fun	71
Check Out The Case Studies • Real people do this.	72
Build The Example Apps • Get familiar with our running examples.	72

Chapter 3: AI Is Weird	• It's new, and unlike what came before.	75
What Even Is AI?	• It helps to know the basics.	75
They Are Like Brains, Not Programs	• You don't program them, they learn.	76
Computers Finally Got Good	• Neural networks went from toys to real tools.	78
Then Things Got Weird	• When you push in all knowledge, things go sideways.	78
Now They Are Smarter Than Us	• In some (but not all) ways.	79
It's Like A Person!	• No computer has ever been person-y-er.	81
It Knows What You Mean	• It does understand what you mean.	81
It Makes Sensible Choices	• You can leave decisions to the model.	82
It Can Think	• Reasoning and problem solving are things it can do.	82
It Understands Similarity	• Vector embeddings are unintuitive but powerful.	84
It's Flexible	• It contains multitudes, just like me.	85
It Really Wants To Please You	• It's trying so hard to make you happy.	85
It Has A Personality	• Different models have different temperaments.	86
It Doesn't Know Everything	• Even if you're tempted to think it is.	86
It Makes Things Up	• Everything is just "likely".	87
It Doesn't Do Things Perfectly	• Sometimes it just gives up	88
It's Not A Person!	• You can't help treating it like one, but be careful	89
It's Vast	• It contains multitudes.	90
It's Fast	• Vroomy vroom vroom!	91
It Doesn't Have Feelings	• Despite the fact that it acts like it does.	91
It Doesn't Learn	• No matter how many times you tell it.	93
It's Jagged	• Some stuff it does really well, some stuff really not.	94
It's Infinitely Patient	• No question is too stupid.	95
It Has No Sense Of Time	• It's eternally right now.	95
You Can't Overwork It	• It doesn't get bored or stressed.	96
You Can Replicate It	• Run as many as you can afford	97
You Can Pick The Smartness	• Mostly smarter is better (but not always)	97
Learn To Harness It	• A few tips for wielding this power well.	99
Get Your Hands Dirty	• It's a practice, not a theory.	99
Think For Yourself	• Don't delegate essential understanding	100
Just Ask	• You want to really understand what's there.	101
Ask What To Ask	• Ask what to ask.	102
Iterate	• Nothing has to be done after one try.	103

Write It Down • Don't throw away your intent.	104
Change Things • Changing your mind is not costly.	104
Take A Do-Over • Don't be afraid to wipe the slate clean.	105
What's Bad About AI? • There are good reasons to be wary.	107
It Produces Slop • AI can make some real garbage.	107
It's Bad For The Earth • Using it to write code is better, at least.	109
It Uses Work Without Permission • Including yours.	109
It's Expensive • It's not free, or even cheap.	111
It Will Make Us Dumb • Will AI take away the friction we learn from?	112
Now They Are Smarter Than Us • We're the smartest! Right ... ?	113

Part 2: Get Building

Chapter 4: Get Ready • Get your machine in order.	116
You Need A Computer • Desktop, laptop, or mobile device.	116
Install The Essentials • text editor, CLI, maybe an IDE	118
Create A Project • make a place for this specific app	122
Get An AI Coding Agent • There are better and worse choices.	123
Introduce Yourself • Let the agent know what you're up to.	125
Enable Source Control • Don't let the dog eat your homework.	126
Sidebar - Text Is Your Medium • You'll grow to love plain text.	130
Text File Formats	133
Chapter 5: Build Your First Real Software • You can do it!	139
Pick Your Stack • It doesn't matter much which one.	139
Prompt It • Speak your program into existence.	141
Run It • See your program in action.	143
Make It Better • Does it need a little improvement?	144
Everything Is Broken! • What if it doesn't work at all?	144
Something Specific Is Broken! • What if it *kinda* works?	147
I Want Something Different! • What if it works, but not how I want?	149
Ship It • Get it out there in production.	151
You Work On A Copy • We never directly touch the real thing!	151
Host It Somewhere • It's got to run somewhere.	151
Push To Production • Let 'er rip!	152
Roll Back Mistakes • Don't freak out, just roll back.	153

You're Technical Now • It's not gross or scary, I promise. 155

Chapter 6: Increase Your Confidence • How can you be sure it's doing what you want. . . . 156

Be Skeptical • Be skeptical of everything. 156

Look At What Changed • A diff is worth a thousand words. 158

Be Careful With Power Tools • Don't fall into the common trap doors. 159

Tests Are Your Safety Net • Build repeatable confidence-raisers. 160

Make Fun Test Data • Why not use a pop culture theme? 162

Why read the rest of this book? • There's a lot more to get the hang of. 164

Part 3: Build The Right Thing

Chapter 7: What Are You Building? • The basic questions. 166

What Level Of Serious Is It? • The four P's. 166

Level 1 - Perishable • Single-use, disposable software. 167

Level 2 - Personal • Software that's just for you (and your peeps). 168

Level 3 - Public • Software that's for anybody in the world. 169

Level 4 - Professional • Software for money, with consequences. 170

What Form Should It Take? • Different horses for different courses. 171

Scripts Are Disposable • Quick, single-use solutions. 171

Libraries Are Reusable • Solve it once(ish) for everyone. 174

Apps Are Packages • Data, abilities, and widgets. 176

Services Wait On You • They respond when asked to. 177

Who Is It For? • Solve problems for real people. 180

What Is Its Purpose? • What jobs will people hire this software for? 182

Chapter 8: What Is Your Domain? • Identify the stuff your app cares about. 184

Name Your Nouns • Establish a list of basic entities. 185

Establish Relationships • Consider how entities relate to each other. 188

Add Some Attributes • Flesh out more detail on each entity. 191

Voice Your Verbs • Identify the key operations and actions you'll do. 193

Strong Versus Weak Typing • How specific do you want to be? 194

Embrace Change • Don't fixate on what you've figured out so far. 195

You Are A Theory Keeper • You're the bridge from the world to the app. 196

Chapter 9: What Should Your Software Do? • work through the flows and functionality. . . 199

Programming Is Deciding • The hard part isn't the typing. 199

Intend and Explore • You need both top-down and bottom-up.	203
Decompose And Capture • Keep a running, growing record of your intent.	206
Explore Incrementally • Don't try to one-shot an app.	207
Discover What You Really Wanted • You have to see it to know.	210
Start Boring • Use shiny new AI features with care.	212
Chapter 10: How Do You Make It Great? • From meh to marvelous.	213
Care About What You Make • "lol shrug" isn't the right level.	213
Expose Yourself To Options • You're only as good as your taste.	216
Spend The Time To Get Clear • Don't rush to build.	217
Follow Your Gut • Be opinionated!	218
Notice Similarities • Find patterns for more power.	219
Notice Differences • Make some things configurable.	220
Remix Known Patterns • You don't have to reinvent the world.	221
Follow Common UI Idioms • Don't (always) reinvent the wheel.	223
Sweat The Details • Little things matter a lot.	225
Validate It With Other People • You have a one-sided view; expand it.	226

Part 4: Build The Thing Right

Chapter 11: Code Isn't Scary • What even *is* this stuff?	229
Programs Are Code • It's the stuff software is made of.	229
There Are Many Languages • And all code is in one of them.	231
All Code Runs Somewhere • On your computer, or someone else's.	235
Code Is Instructions • It runs one step at a time.	239
Comments Explain • Hey, there are real words in there!	244
Abstraction Hides The Details • The less you know, the better.	246
Data Has Structure • There are different *kinds* of values you can use.	248
Control Forks • Many roads diverge in a program.	250
Variables Are Pockets • You keep things in them that you'll need again.	251
Operators Combine Things • You can mash two values into one.	254
Functions Do Stuff • They put code in reusable containers.	255
Objects Represent Things • You can mirror the world in code.	256
Algorithms Are Recipes • You probably don't need to invent new ones.	259
Chapter 12: Interlude - Keep Code Tidy • You didn't write it, but you still own it.	262
Keep It Readable • Let the LLM write well-paced, well-commented code.	262

Don't Be Clever • Clear, simple code beats inscrutable one-liners.	265
Be Consistent • Do the same thing the same way every time.	267
Avoid Spaghetti • Modular code is happy code.	269
Don't Repeat Yourself • A right place for everything, and everything in its place.	273
Take Out The Garbage • Don't leave dead code around to confuse the AI.	275
Chapter 13: Build Web Experiences • It's the lingua franca of software.	276
HTML Is Stupid Easy • A kid could do it.	276
CSS Is Stupid Hard • Professionals struggle with it.	278
Javascript Can Do Anything • It's a mini-computer in the browser.	280
Talk To The Server Anytime • Make network requests in the background.	281
There Are So Many Frameworks • It's overwhelming, actually.	282
Use Modern Web Tooling • The browser is practically its own IDE.	283
Mobile Web Works Great • You can make responsive apps this way.	284
Chapter 14: Store Data • Databases are magic.	286
Sidebar - Disks And Memory • Does data stick around or not?	287
Tables Are (Kinda) Like Spreadsheets • They have rows & columns.	288
Keys Are Unique Identifiers • Make sure your rows have name-tags.	290
Databases Speak SQL • It's a weird, wonderful language.	291
Joins Connect Tables • Foreign keys are the glue.	294
Think In Sets, Not Loops • Let the database do the work.	296
Transactions Are Atomic • All-or-nothing updates.	297
Indexes Are A Good Trade • They give you much faster queries.	298
Schemas Evolve Painfully • Stateful things are hard to change.	299
Explore Other Data Options • There are things other than relational databases.	300
Chapter 15: The Fun Stuff • Beyond the basics, but nothing to be scared of.	301
Do It Later • Asynchrony happens, eventually.	301
Work Offline • Don't assume you always need the internet!	303
Let Them Undo • We all make mistakes.	305
Native Mobile Apps • There are pros and cons to making a "real" app.	306
Big Computation • Don't be afraid of Petabytes!	308
Use External Services • You don't (and can't) build everything yourself.	309
Chapter 16: The Tricky Stuff • This stuff is harder than it seems!	310
Dates And Times Are Hard • Don't try to figure it out by hand.	310

Encodings, Collations, and Quotes, Oh My • [?????]	314
Around The World In 80 Characters • Sprechen Sie Esperanto?	317
Integration Is A Swamp • Like external services, but less fun.	318

Part 5: Get Good

Chapter 17: Go Faster • Put the pedal to the metal.	320
Close The Loop • Let it fix its own errors.	320
Soup Up Your Workflow • Use the latest and greatest tools (if you want).	321
Inhabit The Multiverse • You can do more than one thing at once.	323
Let Ralph Bang His Head On It • You'll keep getting better versions.	325
Move Fast While It's Easy • YOLO up to your risk tolerance.	327
Streamline Deployment • Make shipping it fast, easy and safe.	328
Chapter 18: Be Explicit • Get it out of your head.	329
Write Down What You Want • Be just specific enough.	329
Give It Just Enough Context • Supply the things it can't already know.	334
Don't Overexplain • It knows what you mean.	337
Work Spec First • Work through what you want explicitly before hitting go.	338
Chapter 19: Be Wary • A stitch in time saves nine.	339
Assume It's Wrong • Be pleasantly surprised if it's right!	339
Interrupt It • Don't be too polite.	341
Don't Get Too Fancy • Your ambitions should match your successes.	342
Make More Tests • Tighten your correctness and coverage.	343
Make Tests First • So you know when it works.	344
Tell It What It Can Do • It's often a mystery to itself.	346
Root Out Heresies • Kill those incorrect beliefs that get stuck in context.	347
Chapter 20: Manage Your Cognitive Load • The bottleneck here might be you.	348
Force It To Be Concise • It loves to hear itself talk.	348
Build One Thing At A Time • It's better to move incrementally.	350
Leave Breadcrumbs • always write notes to yourself and the LLM	351
Organize Your Junk Drawer • Everything in its place.	352
Chapter 21: Use AI In Your App • I know I said we wouldn't! But you can.	353
Handle Nebulous Things • Like human language or pictures	353
Encapsulate it • Put it in a box.	355

Account For The Cost • Like many services, it's not free.	356
Predictive AI Is Still Cool • It's not as easy or hot, but it's powerful!	357

Part 6: Share It

Chapter 22: Make It Multi-Player • Your milkshake app will bring all the boys to the yard.	359
You Gotta Log In • Authenticate that you are who you say you are.	359
Boundaries Are Healthy • Don't show people each others' stuff.	361
You're A Super User • Make admin functionality from the start.	363
Concurrent Updates • Don't make people refresh to see each others changes!	364
Cross The Streams • Hook multiple apps up to each other!	366
Chapter 23: Raise The Bar • Make your software actually good.	367
Security • Don't get hacked.	367
Availability • Don't go down.	369
Durability • Don't lose data.	371
Quality • Don't have bugs.	372
Performance • Don't go slow.	374
Observability • Don't fly blind.	378
Usability • Don't make things hard.	379
Accessibility • Don't exclude users with disabilities.	380
Support • Don't leave people hanging.	381
Visual Appeal • Don't make it look gross.	382
Chapter 24: Work With Collaborators • You don't have to toil alone.	383
Discuss It • Clarity comes through conversation.	384
Document It • Write down the stuff that's not obvious.	385
Manage It • People get paid for this.	386
Streamline It • Automate the journey to production.	387
Open Source It • A great way to share!	388
Chapter 25: Get Serious • You don't have to go it alone.	390
Should You Hire A Professional? • At some point you should consider it.	390
Charge Money With Care • It's tricky.	392
Follow The Rules • Compliance, etc.	394
Beware Trolls • Software patents suck.	396
Do A Business • You know, beyond the software.	397

Part 7: Don't Suck

Chapter 26: Get Smarter • Don't let AI make you dumb.	399
Don't Use A Tool or Skill Until You've Read It	399
Get A Lesson On Your Own App	399
Have It Quiz You	400
Look For The Click • You'll know when you feel it.	400
Stop Multitasking • Focus is your secret weapon.	401
Choose Wisely • Which is not everything.	402
Try New Things • Don't fall behind!	404
Chapter 27: Be Awesome • The world needs more awesome.	405
Promote Equality • Help the helpless.	405
Protect People's Privacy • Remember stalkers.	406
Prevent Misuse and Abuse • Don't encourage bad things.	407
Data Ownership • Can you peek at users' data?	409
Beware The Rabbit Hole • Don't ignore your family.	410
Have Fun	411

Conclusion

Continuing Your Journey • You can continue to learn and evolve.	413
On Beyond Software • Someday, this kind of magic might change the world.	414

Appendix

Book Website	417
Bibliography	418
Glossary	421
About The Making Of This Book	423
Conventions In This Book	424
Acknowledgements	425

Preface - Start Now

Hi! I hope you're having a good day.

Do me a quick favor: go to conjuringcode.com/start and type this into the big box you see:

```
Please build me an app to track things my dog does.
```

When you hit "go", you'll see the thing you asked for being built, right before your eyes. It's not a gimmick or a demo. It's real working software.

You probably have a lot of questions at this point. Great! You're in the right book.