The Nature Of Code

"This textbook provides artists, designers, and educators the necessary tools and curricula to employ "creative coding" in their school work and professional practice"--

Processing opened up the world of programming to artists, designers, educators, and beginners. The Processing.py Python implementation of Processing reinterprets it for today's web. This short book gently introduces the core concepts of computer programming and working with Processing. Written by the co-founders of the Processing project, Reas and Fry, along with co-author Allison Parrish, Getting Started with Processing.py is your fast track to using Python's Processing mode.

Since its original publication in 1999, this foundational book has become a classic in its field. This second edition, Code Version 2.0, updates the work and was prepared in part through a wiki, a web site allowing readers to edit the text, making this the first reader-edited revision of a popular book. Code counters the common belief that cyberspace cannot be controlled or censored. To the contrary, under the influence of commerce, cyberspace is becoming a highly regulable world where behavior will be much more tightly controlled than in real space. We can - we must - choose what kind of cyberspace we want and what freedoms it will guarantee. These choices are all about architecture: what kind of code will govern cyberspace, and who will control it. In this realm, code is the most significant form of law and it is up to lawyers, policymakers, and especially average citizens to decide what values that code embodies.

Jack the Ripper and legacy codebases have more in common than you'd think. Inspired by forensic psychology methods, you'll learn strategies to predict the future of your codebase, assess refactoring direction, and understand how your team influences the design. With its unique blend of forensic psychology and code analysis, this book arms you with the strategies you need, no matter what programming language you use. Software is a living entity that's constantly changing. To understand software systems, we need to know where they came from and how they evolved. By mining commit data and analyzing the history of your code, you can start fixes ahead of time to eliminate broken designs, maintenance issues, and team productivity bottlenecks. In this book, you'll learn forensic psychology techniques to successfully maintain your software. You'll create a geographic profile from your commit data to find hotspots, and apply temporal coupling concepts to uncover hidden relationships between unrelated areas in your code. You'll also measure the effectiveness of your code improvements. You'll learn how to apply these techniques on projects both large and small. For small projects, you'll get new insights into your design and how well the code fits your ideas. For large projects, you'll identify the good and the fragile parts. Large-scale development is also a social activity, and the team's dynamics influence code quality. That's why this book shows you how to uncover social biases when analyzing the evolution of your system. You'll use commit messages as eyewitness accounts to what is really happening in your code. Finally, you'll put it all together by tracking organizational problems in the code and finding out how to fix them. Come join the hunt for better code! What You Need: You need Java 6 and Python 2.7 to run the accompanying analysis tools. You also need Git to follow along with the examples.

Using Software in Qualitative Research is an essential introduction to the practice and principles of Computer Assisted Qualitative Data Analysis (CAQDAS), helping the reader choose the most appropriate package for their needs and to get the most out of the software once they are using it. This step-by-step book considers a wide range of tasks and processes, bringing them together to demystify qualitative software and encourage flexible and critical choices and uses of software in supporting analysis. The book can be read as a whole or by chapters, building on one another to provide a holistic sense of the analytic journey without advocating a particular sequential process. Accessible and comprehensive, Using Software in Qualitative Research provides a practical but analytically-grounded guide to thinking about and using software and will be an essential companion for any qualitative researcher.

As the first book to share the necessary algorithms for creating code to experiment with design problems in the processing language, this book offers a series of generic procedures that can function as building blocks and encourages you to then use those building blocks to experiment, explore, and channel your thoughts, ideas, and principles into potential solutions. The book covers such topics as structured shapes, solid geometry, networking and databases, physical computing, image processing, graphic user interfaces, and more.

There's a common belief that cyberspace cannot be regulated-that it is, in its very essence, immune from the government's (or anyone else's) control.Code argues that this belief is wrong. It is not in the nature of cyberspace to be unregulable; cyberspace has no "nature." It only has code-the software and hardware that make cyberspace what it is. That code can create a place of freedom-as the original architecture of the Net did-or a place of exquisitely oppressive control.If we miss this point, then we will miss how cyberspace is changing. Under the influence of commerce, cyberpsace is becoming a highly regulable space, where our behavior is much more tightly controlled than in real space.But that's not inevitable either. We can-we must-choose what kind of cyberspace we want and what freedoms we will guarantee. These choices are all about architecture: about what kind of code will govern cyberspace, and who will control it. In this realm, code is the most significant form of law, and it is up to lawyers, policymakers, and especially citizens to decide what values that code embodies.

Drawing from disciplines as diverse as linguistics, cognitive science, psychology, and neuroscience, The Emoji Code explores how emojis are expanding communication and not ending it. For all the handwringing about the imminent death of written language, emoji—those happy faces and hearts—is not taking us backward to the dark ages of illiteracy. Every day 41.5 billion texts are sent by one quarter of the world, using 6 million emoji. Evans argues that

these symbols enrich our ability to communicate and allow us to express our emotions and induce empathy—ultimately making us all better communicators. Vyvyan Evans's Emoji Code charts the evolutionary origins of language, the social and cultural factors that govern its use, change, and development; as well as what it reveals about the human mind. In most communication, nonverbal cues are our emotional expression, signal our personality, and are our attitude toward our addressee. They provide the essential means of nuance and are essential to getting our ideas across. But in digital communication, these cues are missing, which can lead to miscommunication. The explosion of emoji, in less than four years, has arisen precisely because it fulfills exactly these functions which are essential for communication but are otherwise absent in texts and emails. Evans persuasively argues that emoji add tone and an emotional voice and nuance, making us more effective communicators in the digital age.

The Beauty of Code, the Code of Beauty

Generative Design

Keep It Simple, Make It Valuable, Build It Piece by Piece

Life in Code

Code Girls

The Nature of Code

Processing

Algorithms for Visual Design Using the Processing Language

A Programming Handbook for Visual Designers and Artists

A Personal History of Technology

Code

Blockchain and the Law

A Beginner's Guide to Programming Images, Animation, and Interaction

Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation

In this book, Gary William Flake develops in depth the simple idea that recurrent rules can produce rich and complicated behaviors. Distinguishing "agents" (e.g., molecules, cells, animals, and species) from their interactions (e.g., chemical reactions, immune system responses, sexual reproduction, and evolution), Flake argues that it is the computational properties of interactions that account for much of what we think of as "beautiful" and "interesting." From this basic thesis, Flake explores what he considers to be today's four most interesting computational topics: fractals, chaos, complex systems, and adaptation.

ABOUT THE BOOK Jeff Atwood began the Coding Horror blog in 2004, and is convinced that it changed his life. He needed a way to keep track of software development over time - whatever he was thinking about or working on. He researched subjects he found interesting, then documented his research with a public blog post, which he could easily find and refer to later. Over time, increasing numbers of blog visitors found the posts helpful, relevant and interesting. Now, approximately 100,000 readers visit the blog per day and nearly as many comment and interact on the site. Effective Programming: More Than Writing Code is your one-stop shop for all things programming. Jeff writes with humor and understanding, allowing for both seasoned programmers and newbies to appreciate the depth of his research. From such posts as "The Programmer's Bill of Rights" and "Why Cant Programmers... Program?" to "Working With the Chaos Monkey," this book introduces the importance of writing responsible code, the logistics involved, and how people should view it more as a lifestyle than a career. TABLE OF CONTENTS - Introduction - The Art of Getting Shit Done - Principles of Good Programming - Hiring Programmers the Right Way - Getting Your Team to Work Together - The Batcave: Effective Workspaces for Programmers - Designing With the User in Mind - Security Basics: Protecting Your Users' Data - Testing Your Code, So it Doesn't Suck More Than it Has To - Building, Managing and Benefiting from a Community - Marketing Weasels and How Not to Be One - Keeping Your Priorities Straight EXCERPT FROM THE BOOK As a software developer, you are your own worst enemy. The sooner you realize that, the better off you'll be.I know you have the best of intentions. We all do. We're software developers; we love writing code. It's what we do. We never met a problem we couldn't solve with some duct tape, a jury-rigged coat hanger and a pinch of code. But Wil Shipley argues that we should rein in our natural tendencies to write lots of code: The fundamental nature of coding is that our task, as programmers, is to recognize that every decision we make is a trade-off. To be a master programmer is to understand the nature of these trade-offs, and be conscious of them in everything we write.In coding, you have many dimensions in which you can rate code: Brevity of codeFeaturefulnessSpeed of executionTime spent codingRobustnessFlexibility Now, remember, these dimensions are all in opposition to one another. You can spend three days writing a routine which is really beautiful and fast, so you've gotten two of your dimensions up, but you've spent three days, so the "time spent coding" dimension is way down.So, when is this worth it? How do we make these decisions? The answer turns out to be very sane, very simple, and also the one nobody, ever, listens to: Start with brevity. Increase the other dimensions as required by testing. I couldn't agree more. I've given similar advice when I exhorted developers to Code Smaller. And I'm not talking about a reductio ad absurdum contest where we use up all the clever tricks in our books to make the code fit into less physical space. I'm talking about practical, sensible strategies to reduce the volume of code an individual programmer has to read to understand how a program works. Here's a trivial little example of what I'm talking about: if (s == String.Empty)if (s == "") It seems obvious to me that the latter case is... ...buy the book to read more!

Beginning Graphics Programming with Processing 3 A guide to creating exciting computer graphics with the popular Processing language This book aims to teach the Processing programming language to both non-programmers and experienced programmers alike. Using the book, anyone can learn to create visually stunning graphics and animations, regardless of prior experience, and how to utilise them in web pages and Android applications If you are new to programming this unique book will take you through the fundamentals *of graphics and object-oriented programming from first principals using the exciting graphics of the Processing language to bring your programs to life and provide visual feedback of your progress with examples and explanations of all the steps along the way New and experienced programmers alike will soon be creating stunning static and animated graphics programs using lines, shapes and colour, and interacting with the keyboard and mouse to make exciting, dynamic graphics that change with input from the user before moving on to advanced topics such as: - image manipulation - trigonometry - curve physics - acceleration - 3D graphics The book concludes with a comprehensive introduction to Processing's Programming Modes that provides concrete examples of using your new-found graphics programming skills. You will learn how to use: - Javascript mode to embed your graphics into web pages - Android mode to create amazing graphics and games for Android devices The possibilities are truly endless Welcome to the exciting world of graphics programming!*

The nonfiction debut from the author of the international bestseller Sacred Games about the surprising overlap between writing and computer coding Vikram Chandra has been a computer programmer for almost as long as he has been a novelist. In this extraordinary new book, his first work of nonfiction, he searches for the connections between the worlds of art and technology. Coders are obsessed with elegance and style, just as writers are, but do the words mean the same thing to both? Can we ascribe beauty to the craft of writing code? Exploring such varied topics as logic gates and literary modernism, the machismo of tech geeks, the omnipresence of an "Indian Mafia" in Silicon Valley, and the writings of the eleventh-century Kashmiri thinker Abhinavagupta, Geek Sublime is both an idiosyncratic history of coding and a fascinating meditation on the writer's art. Part literary essay, part technology story, and part memoir, it is an engrossing, original, and heady book of sweeping ideas.

You need to get value from your software project. You need it "free, now, and perfect." We can't get you there, but we can help you get to "cheaper, sooner, and better." This book leads you from the desire for value down to the specific activities that help good Agile projects deliver better software sooner, and at a lower cost. Using simple sketches and a few words, the author invites you to follow his path of learning and understanding from a half century of software development and from his engagement with Agile methods from their very beginning. The book describes software development, starting from our natural desire to get something of value. Each topic is described with a picture and a few paragraphs. You're invited to think about each topic; to take it in. You'll think about how each step into the process leads to the next. You'll begin to see why Agile methods ask for what they do, and you'll learn why a shallow implementation of Agile can lead to only limited improvement. This is not a detailed map, nor a step-by-step set of instructions for building the perfect project. There is no map or instructions that will do that for you. You need to build your own project, making it a bit more perfect every day. To do that effectively, you need to build up an understanding of the whole process. This book points out the milestones on your journey of understanding the nature of software development done well. It takes you to a location, describes it briefly, and leaves you to explore and fill in your own understanding. What You Need: You'll need your Standard Issue Brain, a bit of curiosity, and a desire to build your own understanding rather than have someone else's detailed ideas poured into your head.

For over a century the economics profession has extended its reach to encompass policy formation and institutional design while largely ignoring the ethical challenges that attend the profession's influence over the lives of others. Economists have proven to be disinterested in ethics. Embracing emotivism, they often treat ethics a matter of mere preference. Moreover, economists tend to be hostile to professional economic ethics, which they incorrectly equate with a code of conduct that would be at best ineffectual and at worst disruptive to good economic practice. But good ethical reasoning is not reducible to mere tastes, and professional ethics is not reducible to a code. Instead, professional economic ethics refers to a new field of investigation-a tradition of sustained and lively inquiry into the irrepressible ethical entailments of academic and applied economic practice. The Oxford Handbook of Professional Economic Ethics explores a wide range of questions related to the nature of ethical economic practice and the content of professional economic ethics. It explores current thinking that has emerged in these areas while widening substantially the terrain of economic ethics. There has never been a volume that poses so directly and intensively the question of the need for and content of professional ethics for economics. The Handbook incorporates the work of leading scholars and practitioners, including academic economists from various theoretical traditions; applied economists, beyond academia, whose work has direct and immense social impact; and philosophers, professional ethicists, and others whose work has addressed the nature of "professionalism" and its implications for ethical practice.

Based on the New York Times bestseller The Hidden Life of Trees, this interactive, illustrated book for ages 8-10 introduces kids to the forest through outdoor activities, quizzes, fun facts, photographs, and more! Discover the secret life of trees with this nature and science book for kids: Can You Hear the Trees Talking? shares the mysteries and magic of the forest with young readers, revealing what trees feel, how they communicate, and the ways trees take care of their families. The author of The Hidden Life of Trees, Peter Wohlleben, tells kids about the forest internet, aphids who keep ants as pets, nature’s water filters, and more fascinating things that happen under the canopy. Featuring simple activities kids can try on their own or with parents, along with quizzes, photographs, and more, Can You Hear the Trees Talking? covers a range of amazing topics including: : How trees talk to each other (hint: through the wood wide web!) Why trees are important in the city How trees make us healthy and strong How trees get sick, and how we can help them get better This engaging and visually stunning book encourages at-home learning and fun as kids discover the wonder of the natural world outside their windows. "Lush full-color photos and pictures create an immersive experience and the layout facilitates engaged, delighted learning. ...this book may prompt frequent family visits to, and a new appreciation for, neighborhood trees and local forests.” —Washington Parent

How can we capture the unpredictable evolutionary and emergent properties of nature in software? How can understanding the mathematical principles behind our physical world help us to create digital worlds? This book focuses on a range of programming strategies and techniques behind computer simulations of natural systems, from elementary concepts in mathematics and physics to more advanced algorithms that enable sophisticated visual results. Readers will progress from building a basic physics engine to creating intelligent moving objects and complex systems, setting the foundation for further experiments in generative design. Subjects covered include forces, trigonometry, fractals, cellular automata, self-organization, and genetic algorithms. The book's examples are written in Processing, an open-source language and development environment built on top of the Java programming *language. On the book's website (http: //www.natureofcode.com), the examples run in the browser via Processing's JavaScript mode.*

The Hidden Language of Computer Hardware and Software
Thoughts on the Nature of Mass Movements
The Untold Story of the American Women Code Breakers of World War II
The Cosmic Code
The Rule of Code
The Code Book: The Secrets Behind Codebreaking
Using Software in Qualitative Research
The Emoji Code
Getting Started with Processing.py
Reading Utopian Texts
Generative Art
A practical guide using Processing
More Than Writing Code
Software and Everyday Life

Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Providing an introduction to the ideas of computer programming within the context of the visual arts, this thorough book targets an audience of computer-savvy individuals who are interested in creating interactive and visual work through writing software but have little or no prior experience. --

Au Natural is a collection of essays by Lucille Bertuccio, long-time president and co-founder of The Center for Sustainable Living based in Bloomington Indiana. These essays focus not only on the beauty and diversity of the Earth as expressed locally by its myriad flora and fauna but also on living sustainably to protect this planet and its creatures. While reflecting on the lives of insects, birds, and bats these stories also play the role of cautionary tales calling on us to change our behavior. Included are paeans to earth, air, water, and fire, the “elements” necessary to life on Earth, which modern humans have (for the most part) ignored, damaged, and devastated. Beyond this idealistic view of the potential of living in harmony with the Earth, Ms. Bertuccio's essays invite us to open ourselves to the potentialities within ourselves by seeing the broad range and scope of life on this planet.

The award-winning New York Times bestseller about the American women who secretly served as codebreakers during World War II--a "prodigiously researched and engrossing" (New York Times) book that "shines a light on a hidden chapter of American history" (Denver Post). Recruited by the U.S. Army and Navy from small towns and elite colleges, more than ten thousand women served as codebreakers during World War II. While their brothers and boyfriends took up arms, these women moved to Washington and learned the meticulous work of code-breaking. Their efforts shortened the war, saved countless lives, and gave them access to careers previously denied to them. A strict vow of secrecy nearly erased their efforts from history; now, through dazzling research and interviews with surviving code girls, bestselling author Liza Mundy brings to life this riveting and vital story of American courage, service, and scientific accomplishment.

The Second Edition of Johnny Saldaña's international bestseller provides an in-depth guide to the multiple approaches available for coding qualitative data. Fully up to date, it includes new chapters, more coding techniques and an additional glossary. Clear, practical and authoritative, the book: -describes how coding initiates qualitative data analysis -demonstrates the writing of analytic memos -discusses available analytic software -suggests how best to use The Coding Manual for Qualitative Researchers for particular studies. In total, 32 coding methods are profiled that can be applied to a range of research genres from grounded theory to phenomenology to narrative inquiry. For each approach, Saldaña discusses the method's origins, a description of the method, practical applications, and a clearly illustrated example with analytic follow-up. A unique and invaluable reference for students, teachers, and practitioners of qualitative inquiry, this book is essential reading across the social sciences.

Previous ed. titled: Code and other laws of cyberspace, c1999.

This book is concerned with three central issues: the universality of constraints on code-switching, the nature of the relation between language contact and bilingualism, and the social and linguistic components that facilitate code-switching.

"As gripping as a good thriller." --The Washington Post Unpack the science of secrecy and discover the methods behind cryptography--the encoding and decoding of information--in this clear and easy-to-understand young adult adaptation of the national bestseller that's perfect for this age of WikiLeaks, the Sony hack, and other events that reveal the extent to which our technology is never quite as secure as we want to believe. Coders and codebreakers alike will be fascinated by history's most mesmerizing stories of intrigue and cunning--from Julius Caesar and his Caeser cipher to the Allies' use of the Enigma machine to decode German messages during World War II. Accessible, compelling, and timely, The Code Book is sure to make readers see the past--and the future--in a whole new way. "Singh's power of explaining complex ideas is as dazzling as ever." --The Guardian

A Step-by-Step Guide
Programming Embedded Systems
And Other Laws of Cyberspace
With C and GNU Development Tools
Making Interactive Graphics with Processing's Python Mode
A Handbook for Computational Art and Design
Geek Sublime

Patterns of Code-Switching
Beginning Graphics Programming with Processing 3
Code As Creative Medium
The Computational Beauty of Nature
Quantum Physics as the Language of Nature
Vietnamese-English Bilingualism
The Origin and Nature of Life on Earth

A part of Harper Perennial ' s special " Resistance Library " highlighting classic works that illuminate the " Age of Trump " : A boldly packaged reissue of the classic examination of dangerous nationalist political movements. " Its theme is political fanaticism, with which it deals severely and brilliantly. " —New Yorker A stevedore on the San Francisco docks in the 1940s, Eric Hoffer wrote philosophical treatises in his spare time while living in the railroad yards. The True Believer—the first and most famous of his books—was made into a bestseller when President Eisenhower cited it during one of the earliest television press conferences. Called a " brilliant and original inquiry " and " a genuine contribution to our social thought " by Arthur Schlesinger, Jr., this landmark in the field of social psychology is completely relevant and essential for understanding the world today as it delivers a visionary, highly provocative look into the mind of the fanatic and a penetrating study of how an individual becomes one.

The never-more-necessary return of one of our most vital and eloquent voices on technology and culture, the author of the seminal Close to the Machine The last twenty years have brought us the rise of the internet, the development of artificial intelligence, the ubiquity of once unimaginably powerful computers, and the thorough transformation of our economy and society. Through it all, Ellen Ullman lived and worked inside that rising culture of technology, and in Life in Code she tells the continuing story of the changes it wrought with a unique, expert perspective. When Ellen Ullman moved to San Francisco in the early 1970s and went on to become a computer programmer, she was joining a small, idealistic, and almost exclusively male cadre that aspired to genuinely change the world. In 1997 Ullman wrote Close to the Machine, the now classic and still definitive account of life as a coder at the birth of what would be a sweeping technological, cultural, and financial revolution. Twenty years later, the story Ullman recounts is neither one of unbridled triumph nor a nostalgic denial of progress. It is necessarily the story of digital technology ' s loss of innocence as it entered the cultural mainstream, and it is a personal reckoning with all that has changed, and so much that hasn ' t. Life in Code is an essential text toward our understanding of the last twenty years—and the next twenty.

Uniting the foundations of physics and biology, this groundbreaking multidisciplinary and integrative book explores life as a planetary process.

Summary Generative Art presents both the technique and the beauty of algorithmic art. The book includes high-quality examples of generative art, along with the specific programmatic steps author and artist Matt Pearson followed to create each unique piece using the Processing programming language. About the Technology Artists have always explored new media, and computer-based artists are no exception. Generative art, a technique where the artist creates print or onscreen images by using computer algorithms, finds the artistic intersection of programming, computer graphics, and individual expression. The book includes a tutorial on Processing, an open source programming language and environment for people who want to create images, animations, and interactions. About the Book Generative Art presents both the techniques and the beauty of algorithmic art. In it, you'll find dozens of high-quality examples of generative art, along with the specific steps the author followed to create each unique piece using the Processing programming language. The book includes concise tutorials for each of the technical components required to create the book's images, and it offers countless suggestions for how you can combine and reuse the various techniques to create your own works. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. What's Inside The principles of algorithmic art A Processing language tutorial Using organic, pseudo-random, emergent, and fractal processes === Table of Contents Part 1 Creative Coding Generative Art: In Theory and Practice Processing: A Programming Language for ArtistsPart 2 Randomness and Noise The Wrong Way to Draw A Line The Wrong Way to Draw a Circle Adding Dimensions Part 3 Complexity Emergence Autonomy Fractals

Sixth-grader Emmy tries to find her place in a new school and to figure out how she can create her own kind of music using a computer.

Rob Kitchin and Martin dodge examine software from a spatial perspective, analyzing the dyadic relationship of software and space. The production of space, they argue, is increasingly dependent on code, and code is written to produce space. Kitchin and Dodge argue that software, through its ability to do work in the world, transduces space. They develop a set of conceptual tools for identifying and understanding the interrelationship of software, space, and everyday life, and illustrate their argument with rich empirical material. And, finally, they issue a manifesto, calling for critical scholarship into the production and workings of code rather than simply the technologies it enables---a new kind of social science focused on explaining the social, economic, and spatial contours of software.

Since Bitcoin appeared in 2009, the digital currency has been hailed as an Internet marvel and decried as the preferred transaction vehicle for all manner of criminals. It has left nearly everyone without a computer science degree confused: Just how do you " mine " money from ones and zeros? The answer lies in a technology called blockchain, which can be used for much more than Bitcoin. A general-purpose tool for creating secure, decentralized, peer-to-peer applications, blockchain technology has been compared to the Internet itself in both form and impact. Some have said this tool may change society as we know it. Blockchains are being used to create autonomous computer programs known as " smart contracts, " to expedite payments, to create financial instruments, to organize the exchange of data and information, and to facilitate interactions between humans and machines. The technology could affect governance itself, by supporting new organizational structures that promote more democratic and participatory decision making. Primavera De Filippi and Aaron Wright acknowledge this potential and urge the law to catch up. That is because disintermediation—a blockchain ' s greatest asset—subverts critical regulation. By cutting out middlemen, such as large online operators and multinational corporations, blockchains run the risk of undermining the capacity of governmental authorities to supervise activities in banking, commerce, law, and other vital areas. De Filippi and Wright welcome the new possibilities inherent in blockchains. But as Blockchain and the Law makes clear, the technology cannot be harnessed productively without new rules and new approaches to legal thinking.

A poetic translation of the classic Arthurian story is an edition in alliterative language and rhyme of the epic confrontation between a young Round Table hero and a green-clad stranger who compels him to meet his destiny at the Green Chapel. Reprint. 20,000 first printing.

And Other Laws of Cyberspace, Version 2.0

Making Interactive Graphics in JavaScript and Processing

3659 Proposals About Data, Design, and the Nature of Cities

The Oxford Handbook of Professional Economic Ethics

Effective Programming

Au Natural: Ruminations on Nature

Code/space

The Emergence of the Fourth Geosphere

Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs

Can You Hear the Trees Talking?

Learning Processing

Your Code as a Crime Scene

Discovering the Hidden Life of the Forest

Emmy in the Key of Code

Pearls have enthralled global consumers since antiquity, and the Spanish monarchs Ferdinand and Isabella explicitly charged Columbus with finding pearls, as well as gold and silver, when he sailed westward in 1492. American Baroque charts Spain's exploitation of Caribbean pearl fisheries to trace the genesis of its maritime empire. In the 1500s, licit and illicit trade in the jewel gave rise to global networks, connecting the Caribbean to the Indian Ocean to the pearl-producing regions of the Chesapeake and northern Europe. Pearls□a unique source of wealth because of their renewable, fungible, and portable nature□defied easy categorization. Their value was highly subjective and determined more by the individuals, free and enslaved, who produced, carried, traded, wore, and painted them than by imperial decrees and tax-related assessments. The irregular baroque pearl, often transformed by the imagination of a skilled artisan into a fantastical jewel, embodied this subjective appeal. Warsh blends environmental, social, and cultural history to construct microhistories of peoples' wide-ranging engagement with this deceptively simple jewel. Pearls facilitated imperial fantasy and personal ambition, adorned the wardrobes of monarchs and financed their wars, and played a crucial part in the survival strategies of diverse people of humble means. These stories, taken together, uncover early modern conceptions of wealth, from the hardscrabble shores of Caribbean islands to the lavish rooms of Mediterranean palaces.

With three billion more humans projected to be living in cities by 2050, all design is increasingly urban design. And with as much data now produced every day as was produced in all of human history to the year 2007, all architecture is increasingly information architecture. Praised in the New York Times for its "intelligent enquiry and actionable theorizing," Local Code is a collection of data-driven tools and design prototypes for understanding and transforming the physical, social, and ecological resilience of cities. The book's data-driven layout arranges drawings of 3,659 digitally-tailored interventions for vacant public land in San Francisco, Los Angeles, New York City, and Venice, Italy. Between these illustrated case studies, critical essays present surprising and essential links between such designs and the seminal work of urbanist Jane Jacobs, artist Gordon Matta-Clark, and digital mapping pioneer Howard Fisher, along with the developing science of urban nature and complexity. In text and image, Local Code presents a digitally prolific, open-ended approach to urban resilience and social and environmental justice; At once analytic and visionary, it pioneers a new field of enquiry and action at the meeting of big data and the expanding city.

Within literature, history, politics, philosophy and theology, the interpretation of utopian ideals has evolved constantly. Juxtaposing historical views on utopian diagnoses, prescriptions and on the character and value of utopian thought with more modern interpretations, this volume explores how our ideal utopia has transformed over time. Challenging long-held interpretations, the contributors turn a fresh eye to canonical texts, and open them up to a twenty-first century audience. From Moore's Utopia to Le Guin's The Dispossessed, Utopian Moments puts forward a lively and accessible debate on the nature and significance of utopian thought and tradition. Each essay focuses on a key passage from the selected work using it to encourage both the specialist and the reader new to the field to read afresh. Written by an international team of leading scholars, the essays range from the sixteenth century to the present day and are designed to be both stimulating and accessible.

Learning Processing, Second Edition, is a friendly start-up guide to Processing, a free, open-source alternative to expensive software and daunting programming languages. Requiring no previous experience, this book is for the true programming beginner. It teaches the basic building blocks of programming needed to create cutting-edge graphics applications including interactive art, live video processing, and data visualization. Step-by-step examples, thorough explanations, hands-on exercises, and sample code, supports your learning curve. A unique lab-style manual, the book gives graphic and web designers, artists, and illustrators of all stripes a jumpstart on working with the Processing programming environment by providing instruction on the basic principles of the language, followed by careful explanations of select advanced techniques. The book has been developed with a supportive learning experience at its core. From algorithms and data mining to rendering and debugging, it teaches object-oriented programming from the ground up within the fascinating context of interactive visual media. This book is ideal for graphic designers and visual artists without programming background who want to learn programming. It will also appeal to students taking college and graduate courses in interactive media or visual computing, and for self-study. A friendly start-up guide to Processing, a free, open-source alternative to expensive software and daunting programming languages No previous experience required□this book is for the true programming beginner! Step-by-step examples, thorough explanations, hands-on exercises, and sample code supports your learning curve

Generative design, once known only to insiders as a revolutionary method of creating artwork, models, and animations with programmed algorithms, has in recent years become a popular tool for designers. By using simple languages such as JavaScript in p5.js, artists and makers can create everything from interactive typography and textiles to 3D-printed furniture to complex and elegant infographics. This updated volume gives a jump-start on coding strategies, with step-by-step tutorials for creating visual experiments that explore the possibilities of color, form, typography, and images. Generative Design includes a gallery of all-new artwork from a range of international designers□fine art projects as well as commercial ones for Nike, Monotype, Dolby Laboratories, the musician Bjork, and others.

" This is one of the most important books on quantum mechanics ever written for lay readers, in which an eminent physicist and successful science writer, Heinz Pagels, discusses and explains the core concepts of physics without resorting to complicated mathematics. "Can be read by anyone. I heartily recommend it!" -- New York Times Book Review. 1982 edition"--

Processing opened up the world of programming to artists, designers, educators, and beginners. The p5.js JavaScript implementation of Processing reinterprets it for today's web. This short book gently introduces the core concepts of computer programming and working with Processing. Written by the co-founders of the Processing project, Reas and Fry, along with Lauren McCarthy, one of the minds behind p5.js, Getting Started with Processing gets you in on the fun!