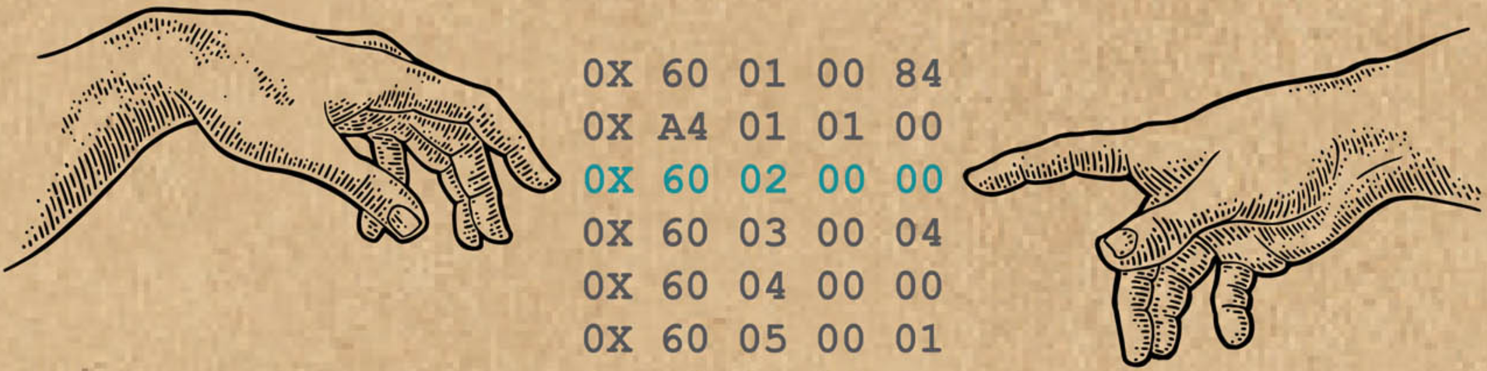
30-SECOND CODING

The 50 essential principles that instruct technology, each explained in half a minute

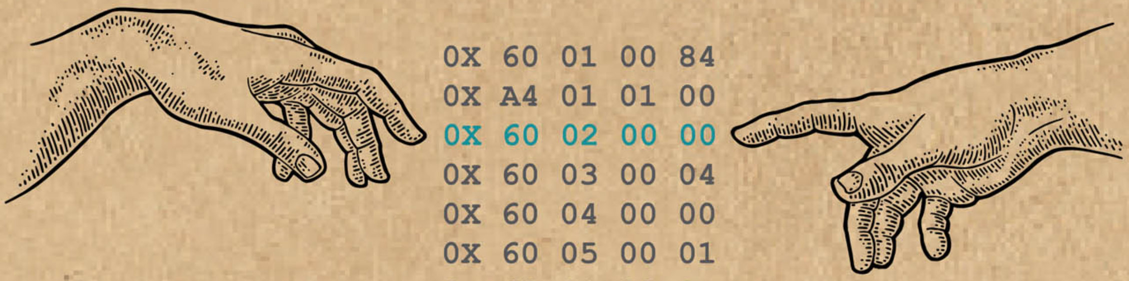


```
0X 60 01 00 84
0X A4 01 01 00
0X 60 02 00 00
0X 60 03 00 04
0X 60 04 00 00
0X 60 05 00 01
0X 60 01 00 84
0X A4 01 01 00
0X 60 02 00 00
0X 60 03 00 04
0X 60 04 00 00
0X 60 05 00 01
```

Editor
Mark Steadman

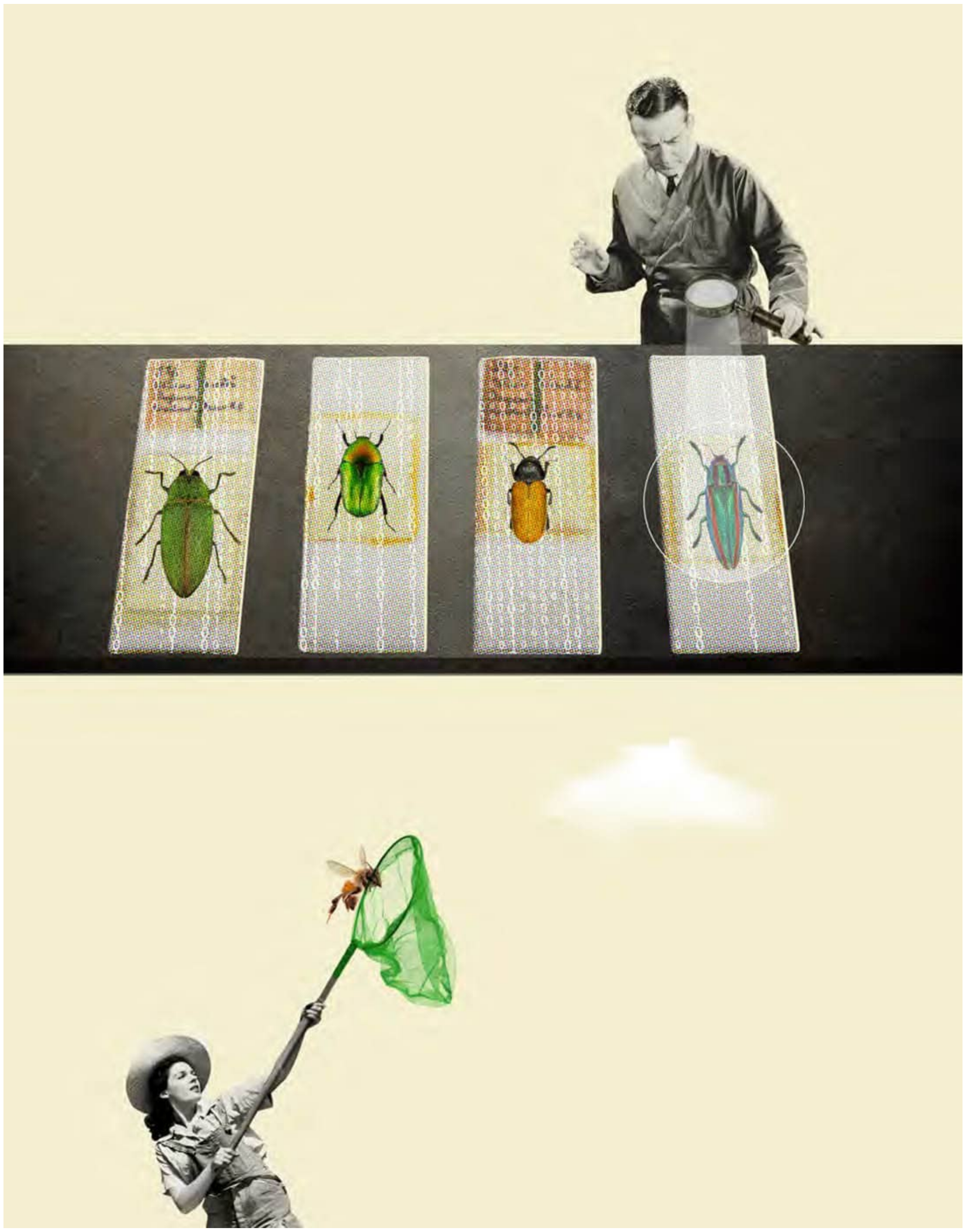
30-SECOND CODING

The 50 essential principles that instruct technology, each explained in half a minute



```
0X 60 01 00 84
0X A4 01 01 00
0X 60 02 00 00
0X 60 03 00 04
0X 60 04 00 00
0X 60 05 00 01
0X 60 01 00 84
0X A4 01 01 00
0X 60 02 00 00
0X 60 03 00 04
0X 60 04 00 00
0X 60 05 00 01
```

Editor
Mark Steadman



30-SECOND CODING

The 50 essential principles that instruct technology, each explained in half a minute

Editor
Mark Steadman

Contributors
Adam Juniper
Suze Shardlow
Mark Steadman

Illustrator
Nicky Ackland-Snow

IVY PRESS

CONTENTS

Introduction

First Computers

GLOSSARY

The Industrial Revolution

The Difference Engine

Human Computers

Task-Specific Computers

Mechanical Computers

Codebreakers

Instruction Sets & Stored Programs

The Fetch-Execute Cycle

The Transistor Revolution

The Personal Computer Age

Profile: Ada Lovelace

Instructing Computers

GLOSSARY

Assembly Language

Fortran: The First High-Level Language

Procedural Languages

Compiled Code

Object-Oriented Programming (OOP)

Code Libraries

Running Code in the Cloud

Profile: Grace Hopper

Code Concepts

GLOSSARY

Binary & Bits

Data Types

Data Structures: Arrays

Variables

If-Then-Else: Conditional Statements

Loops & Iterations

Functions

Making Code Portable

Buffering & Caches

Eventual Consistency

Profile: Linus Torvalds

Profile: Steve Wozniak

What Coders Do

GLOSSARY

User Interface & User Experience

Database Operation: CRUD

Web Development

Scripting

Engineering

Agile Development & the Scrum

Hacking

Debugging

Scaling & Pseudocode

Profile: Tim Berners-Lee

Solving Problems with Code

GLOSSARY

Algorithms

The FizzBuzz Test

Sorting & Big O Notation

The Two Generals Problem

Compression & the Huffman Tree

Search Engine Optimization

Face Detection

Unicode

End-to-End Encryption

Pattern Matching Languages

Profile: Larry Page

Modern Concerns & Concepts

GLOSSARY

AI: Artificial Intelligence

Black Box

Computers Cannot Determine Truth

What is Blockchain?

Detecting Bots

Integrated Development Environment

Profile: Alan Turing

Appendices

Resources

Notes on Contributors

Index

Acknowledgments

INTRODUCTION

Mark Steadman

Code is one of the most significant building blocks in modern society. Every time we send an emoji to a friend, we're sending a tiny piece of code (a string of letters and numbers) across a virtual wire. That string of numbers and letters, called hexadecimal code, is then read by our friend's phone and associated with an image. When we pick up the phone to talk to our friend, code converts our analogue voice to digital data, which is encoded at one end and then decoded at the other.

What we call 'code' is a set of instructions, written in a particular language. That language depends on a number of things, like how easy it is for us humans to read and write it, how quickly a computer can understand it, the number of other computers that speak the same language and the features that language provides. Coding (or programming) can be as simple as adding two numbers together, or as complex as constructing a vast neural network that can perform complicated machine learning tasks. Code can enable mass change within a society, or it can help you rescue a few minutes from your busy day.

You don't need to have aced your maths exams in order to be a great coder. As long as you can think logically, putting one thought in front of the other, you can code. Nor do you need to memorize thousands of obscure commands, because most of us still turn to Google when we can't remember how a particular aspect of our programming language works!

You don't even need what you might consider a traditional computer in order to write code. A tablet or a smartphone will do, and there are apps that can help you learn, and achieve practical results. In this book we'll cover the key events in the evolution of computer programming, from the first-ever human computers to modern cloud infrastructures that can help scale bedroom businesses up to massive corporations.

Coding is no longer the preserve of the stereotypical basement nerd. It’s everywhere, from those blocky square barcodes that we scan with our phones, to the encryption that keeps our WhatsApp conversations from being seen by prying eyes. It’s the job of these pages to open your eyes to that wider world so you can use it as you choose, whether that’s to build the next TikTok or just understand why it’s a good idea to turn the computer off and on again.

```
### COMMODORE BASIC ###
 7167 BYTES FREE
READY.
```

```
   03 LW-PARM-LEN
   03 LW-PARM-VAL
/
PROCEDURE DIVISION USING LW-PARM.
/
A001-MAINLINE SECTION.

     MOVE LW-PARM-VAL TO WW-NUM-ELEMS
     PERFORM B001-NEW-ARRAY
```

```
MOVE LW-PARM-VAL TO WW-NUM-ELEMS
PERFORM B001-NEW-ARRAY

COMPUTE WW-INCREMENT =
   WW-NUM-ELEMS / 200000
END-COMPUTE

PERFORM VARYING WW-ARRAY-IDX FROM 1 BY WW-INCREMENT
   UNTIL WW-ARRAY-IDX > WW-NUM-ELEMS
   MOVE WW-ARRAY-IDX TO WW-DATA
```

A tour of this book

In this book we present the ideas closest to the hearts of computer programmers (you can add your own joke about a well-known science-fiction franchise here). In academic circles, what you might know as programming is known as ‘computer science’, and it is a subject that isn’t that old. For that reason, we’ve begun by devoting a whole chapter to the emergence of the computer as we understand it today, before breaking down the means by which we can instruct them. Whether it’s introducing the Difference Engine or face detection, each entry is broken into several parts. The centrepiece is the **30-second code** – the explanation itself. If you’re short on time, the **3-second bit** squeezes the essence of that into a single sentence, while the **3-minute byte** offers a broader context.

Computing is also more about the people involved than you likely imagined the first time you saw the blue screen of death. For that reason, this book introduces many of the names behind innovations in code and computing in general. Check out the **3-second biographies** and you’ll find a little more about many of the big names past and present. The book also features more detailed profiles of some folk who you’ve likely heard of even beyond the geek-o-sphere!

FIRST COMPUTERS

FIRST COMPUTERS

GLOSSARY

algorithm

The term for the mathematical aspect of a computer program; often an over-simplification. For example, Google is often described as having an algorithm in the singular, but many aspects go towards the system's ranking of websites.

base

In maths, the base is the number on which the counting scale is based. 'Base 10' means writing numbers using ten symbols, including zero, so 0–9.

Bernoulli numbers

Named for mathematician Jacob Bernoulli. He described a probability – the same each time – being calculated over repetition of the same event, for example a coin toss or dice throw.

Boolean logic

In mathematics, something that is Boolean (such as the result of a certain formula) can have only two states: true or false. The term leapfrogged into computer science not just to refer to early systems but, for example, if you encounter a checkbox on a web form it'll likely be linked to an on/off or true/false field in a database.

binary

Literally means composed of two things, or having two parts. In computing, a system of counting that uses base 2, and can be depicted: 0, 1, 10, 11, 100, 101, 111, 1000 and so on.

bit
A bit may only have two states: on or off (usually represented as 0 or 1). The word is a portmanteau of binary digit.

DOS
Disk Operating System, as opposed to DoS (denial of service – a cyber attack).

floating point
A method of mathematical representation for real numbers that are especially large or small, which takes the form [significand] X [base] to the power of [exponent]: 2.5951 X 10^4 = 25951. You might also choose a limit on the number of digits in the significand to reduce the overall computational load on the computer – limit it to three decimal places and we would only record 2.595*10^4 (in code * = multiply).

general-purpose computer
A computer that can be programmed to carry out different operations (traditionally arithmetic or logical). Earlier devices could perform only the operation they were designed for.

loop
In programming, a section of the program repeated (either until a condition is met or, by mistake, infinitely causing the program never to end).

memory
The element of a computer that stores information for the active program, as opposed to 'storage'.

operating system
System software that manages the computer hardware, provides common resources for applications, and allows the launching and termination of

applications.

PC
Personal Computer – a name coined to distinguish early PCs from the expensive room-sized computers of the time.

processor
Also known as the Central Processor Unit, or CPU. This component of a digital computer performs the calculations.

program
As a noun, a precise set of instructions to tell a computer what to do and how. It can also be used as a verb, meaning the act of creating a program.

punch card
A card specifically designed to store information. A batch of identical cards would be printed, for example with the numbers 1 to 10 written on them. The operator could store information by punching a hole through the appropriate number. A tabulating machine could read the card from the position of the hole(s) in it.

semiconductor
A material that can conduct less than a fully conductive material (such as copper) and more than insulators (such as ceramics). Silicon, an element, is a semiconductor, and (unlike metal) it becomes more conductive as its temperature increases.

silicon
This is a base metal, but in computing usually refers to the components (integrated circuits) made from it, and especially central processing units, for example, ‘This computer has Apple silicon’.

software
A collection of data that can be read by the computer as instructions on how to work, as opposed to the hardware that actually performs the work.

statement
A single line of code correctly written that makes a command.

tabulating machine
The machine that processed punch cards and recorded totals or performed similar actions.

variable
In programming, all data being manipulated needs to be stored in the computer’s memory. A space is allocated for each piece of data and given a name so the program can access it – this is the variable.

THE INDUSTRIAL REVOLUTION

the 30-second code

When you imagine computer code, it’s hard not to picture someone with questionable (or obsessive) hygiene hunched over a computer. Before we even get to the fact that this simply isn’t a fair representation of the community, it’s also important to realize that a concept of programming existed long before there were little glowing screens or QUERTY keyboards to complete the stereotype. In fact, the point at which humanity first began to hand over repetitive tasks to machines was the Industrial Revolution, the vanguard of which was the textile industry. In 1750, Britain imported around 1,100 tonnes of cotton, which was spun on handwheels. By the year 1800 that figure was 24,000 tonnes and growing fast. The key driving factor was the arrival of stationary steam engines powering large factories, or ‘mills’, processing the growing supply of cotton from the American colonies. Britain’s rapid industrialization quickly made it the leading global power, which didn’t go unnoticed by Napoleon. The French ruler enthusiastically supported his growing textile industry in Lyon, where Joseph-Marie Jacquard was developing his loom-related inventions. His 1801 Jacquard loom featured punch cards to weave patterned silk automatically, giving France the edge. It was made property of the French state; there were 11,000 in use in France a decade later.

3-SECOND BIT

The Industrial Revolution created the conditions for the punch-card system of data storage long before computing had been invented.

3-MINUTE BYTE

Punch cards allowed needles to pass through holes in the card, or not when no hole existed. By being either there or not, the holes acted with exactly the kind of binary precision that appealed to Charles Babbage (see here), who even ordered a portrait of Jacquard to be woven on one of his looms, which required 24,000 punch cards.

RELATED TOPICS

See also

THE DIFFERENCE ENGINE

MECHANICAL COMPUTERS

3-SECOND BIOGRAPHIES

JAMES WATT

1736–1819

Scottish inventor who made the steam engine practical in 1765, effectively launching the first Industrial Revolution

JOSEPH-MARIE JACQUARD

1752–1834

French weaver and merchant awarded the Legion of Honour for inventing the automated loom

30-SECOND TEXT

Adam Juniper