INDEX

Symbols

& (AND operator), 242–243
~ (complement operator), 244
<< (left-shift operator), 239–240
- (minus sign), 79
| (OR operator), 240–242
>> (right-shift operator), 240
:= (walrus operator), 77
^ (XOR operator), 130, 243–244

A

abstraction, 23
abstract syntax tree, 9, 32–33, 39, 42–48
    layers of, 117
ADC instruction (6502), 163
addressing modes (6502), 153–154, 170–172
ahead-of-time compilation, 17
algorithms
    dithering, 67–72
    hill climbing, 105
    KNN, 207–218, 222–224
        accuracy, 218
        machine learning, 206–218
        performance of, 211
    stochastic optimization, 111
    stochastic painting, 89–95
Allen, Paul, 22–23, 201
Allison, Dennis, 30, 42
AND operator (&), 242–243
animated GIFs, 64, 86
APU (audio processing unit), 143, 145
ArgumentParser, 11–12, 33, 66, 95–96, 146
artificial intelligence (AI), 206
ASL instruction (6502), 163
AST (abstract syntax tree), 9, 32–33, 39, 42–48
Atkinson, Bill, 67, 86
Atkinson dithering, 67–72, 86
attribute tables (NES), 180–183
audio on NES, 143

B

background (NES), 177, 187–189
Backus–Naur form (BNF), 28–32
bank switching (NES), 144, 148–149
BASIC, 21–59
    dialects of, 22–23
    history of, 22–23
    variables in, 25
BCC instruction (6502), 163
BCS instruction (6502), 163
BEQ instruction (6502), 163
big-endian order, 81–82
binary
    operations, 239–244
    review of, 237–239
binary-coded decimal (BCD), 121
BIT instruction (6502), 163
bit planes, 179–180
bits, 179, 238
    per pixel, 181
bitwise operations, 237–244
BMI instruction (6502), 164
BNE instruction (6502), 164
BNF (Backus–Naur form), 28–32
Boolean expressions in BASIC, 26, 31
    parsing in NanoBASIC, 46
boolean_expr production rule, 32
BPL instruction (6502), 164
Brainfuck, 3–19
    description of, 4
    example programs, 6–9
    history of, 3–4
    implementation of, 10–14
    testing the interpreter, 15–16
BRK instruction (6502), 164
Brofeldt, Pekka, 212
buffer overflow protection, 112
bytecode, 17, 116, 139
bytes, 238–239

C

C (programming language), 6
cartridges (NES), 144, 148–152
characters in terminals and non-terminals, 29
CHIP-8, 115–140
  description of, 115–117
  history of, 116–117
  implementation of, 122–136
  instructions, 119–122
  memory and registers, 117–119
  playing games on, 137–138
  testing the virtual machine, 136–137
CHR RAM (NES), 149, 177–178
CHR ROM (NES), 149, 177–178
classification
  of digits, 215–217
  of fish, 212–214
  with KNN, 205–219
`CLC` instruction (6502), 164
`CLD` instruction (6502), 164
`CLI` instruction (6502), 165
CLR (Common Language Runtime), 139–140
`CLV` instruction (6502), 165
`CMP` instruction (6502), 165
collision detection (NES), 191
color dithering algorithms 63–72
  in MacPaint, 180–182
  in NES, 183–184
  in stochastic painting algorithm, 90–95
Common Language Runtime (CLR), 139–140
compilation, 17, 57
compilers, 17
complement (~) operator, 244
computational art, 63–112
  creating abstract impressions, 89–112
`consume()` function, 42
COSMAC VIP, 117
`Counter` (Python), 211
`CPM` instruction (6502), 165
CPU (central processing unit) in NES, 143, 152–176
`CPX` instruction (6502), 165
`CPY` instruction (6502), 165
cross validation, 218
CRT televisions, 147, 185
CSV files, 210, 212–213, 215

D

data fork, 80
`DataPoint` class, 209–211
data points, 209–211
`DEC` instruction (6502), 165
declaration vs. imperative languages, 23
DEFLATE algorithm, 86
`DEX` instruction (6502), 165
`DEY` instruction (6502), 165
digital art, 85
`Digit` class, 215–216
digits classification of, 215–217
  handwritten, 215–217, 224–228
  regression on, 224–228
Dijkstra, Edsger, 27, 48
dithering, 63–87
  algorithms for, 67–72
  Atkinson dithering, 67–72, 86
  explanation of, 64–65
  Floyd-Steinberg dithering, 67–69, 86
DMA (direct memory access), 156, 174
domain-specific languages (DSLs), 57
drawing
  backgrounds (NES), 187–189
  using CHIP-8, 128–130
  using Pygame, 224–228
dynamic recompilation, 132

E

`EOR` instruction (6502), 165
error-diffusion dithering, 67–69
error handling
  in BASIC interpreter, 40–41
  in CHIP-8 VM, 126
esoteric programming languages, 3, 10
Euclidean distance, 206, 209, 211, 218

F

Fayzullin, Marat, 149
Fibonacci sequence, 27–28, 135

file formats
  iNES, 149
  MacBinary, 80–83
  MacPaint, 72–87
finite state machine, 152
Fish class, 213–214
fish classification, 212–214
fish weight prediction, 224
flags register (6502), 155–176
flipping sprites (NES), 190–191
Floyd-Steinberg dithering, 67–69, 86
font set (CHIP-8), 126–127
forks (Mac OS), 80–82
FPS (frames per second), 185, 199–200
frames (NES), 185
functional programming, 23–24, 52

G

games
  BrickBreaker, 178, 181, 198
  Chase, 198
  Lan Master, 199
  Thwaite, 182
Gates, Bill, 22–23, 201
getattr() function, 224
getcolors() method (Pillow), 100
getpixel() method (Pillow), 71
GIFs, 64, 86, 96, 107, 227–228
GOSUB statement (BASIC), 24, 26
GOTO statement (BASIC), 24, 26–27
grammar
  for BASIC, 28–32
  context-free, 30
  formal definition of, 28–32
graphics
  on CHIP-8, 128–130
  on NES, 176–185
  in retro image processing, 63–87
  in stochastic painting algorithm, 89–112
grayscale, 64–66, 215

H

halftone, 64
Hamming distance, 209, 218
hblank (NES), 147, 185
hill climbing, 105–106
homebrew games, 143

I

IBM PC, 23, 200
IF statement (BASIC), 26
  parsing in NanoBASIC, 43–44
IfStatement node, 36–37
IF...THEN statements, 26
image processing, 63–87
imperative vs. declarative languages, 23
impressionism, 90
INC instruction (6502), 165
index register, 118
iNES file format, 149
inheritance, 37–39
InterpreterError class, 40–41
interpreters, 3–59
  Brainfuck, 3–19
  components of, 9, 32–33
  implementing, 10–14
  NanoBASIC, 21–59
  overview of, 17–18
  parser, 9, 41–48
  structure of, 9
  runtime, 9, 49–53
  tokenizer, 9, 33–36
INX instruction (6502), 166
INY instruction (6502), 166

J

Java, 17, 18, 116, 117, 139
JIT (just-in-time) compilation, 17, 140
JMP instruction (6502), 166
Joypad class, 154, 173–174
JSR instruction (6502), 166
jump tables, 131–132, 157–161
just-in-time (JIT) compilation, 17, 140

K

k-d tree, 218
Kemeny, John, 22, 55
k-means clustering, 235
KNN (*k*-nearest neighbors), 205–229
  algorithm explanation, 207–209
  classification with, 205–219
  history of, 205
  implementing, 209–212
  regression with, 221–229

Kopec, Danny, 55
Kurtz, Thomas, 22, 55

L

LDA instruction (6502), 166
LDX instruction (6502), 166
LDY instruction (6502), 166
left-shift (<<) operator, 239–240
LET statement (BASIC), 25, 45
  parsing in NanoBASIC, 45
LetStatement node, 39
libraries
  Pillow, 66–67, 71–72, 82–84, 99–102
  Pygame, 123–124, 126, 146–147, 216, 224–228
line numbering (BASIC), 24
lines (stochastic painting), 93–95
LineStatement node, 39
Lisp, 17
little-endian order, 82
LLMs (large language models), 206
logic programming, 23
Logo (programming language), 56
LSR instruction (6502), 166

M

MacBinary, 80–83
machine learning, 205–229
  history of, 206
  KNN algorithm, 207–218
  regression with KNN, 221–229
Macintosh, 63–65, 80–82, 201
Mac OS, 80–82
MacPaint, 63–87
  file format, 72–87
  run-length encoding, 73–79
mappers (NES), 149, 201
match statement, 12, 48, 132, 151
MemMode enum, 153–154, 170–172
memory access modes (6502), 153–154, 170–172
memory in CHIP-8, 117–119
memory-mapped registers, 143, 172–173
memory page, 154
microprocessor, 143, 152–153
Microsoft, 22–23, 140, 201
minus sign (-), 79
mirroring (NES), 172–174, 180–181
MOS 6502, 200
Müller, Urban, 3–4, 6, 10

N

nametables (NES), 180–181
NanoBASIC, 21–59
  description of, 22–24
  example programs, 27–28
  formal grammar, 28–32
  implementation of, 32–53
  parser, 41–48
  running a program, 53
  style and syntax, 24–27
  testing, 53–57
  tokenizer, 33–36
nearest() method, 211
NES (Nintendo Entertainment System), 141–201
  background drawing, 187–189
  cartridges, 148–152
  colors, 183–184
  CPU emulation, 152–176
  description of, 142–144
  emulator structure, 145
  frames and timing, 185
  hardware, 143–144
  history of, 142–143
  instruction execution, 130–136
  main loop, 145–148
  memory access, 170–175
  PPU emulation, 176–195
  software, 144–145
  sprite drawing, 189–191
  testing the emulator, 194–197
NesDev, 142–143, 183, 234–235
neural networks, 206, 228, 235
nibble, 119
non-terminal, 28
NOP instruction (6502), 166
NROM mapper, 149
NumPy, 123, 146, 189, 216, 223–224
Nystrom, Robert, 234

O

object-oriented programming (OOP), 23
OCR (optical character recognition), 215
one-time pad, 244
opcodes
  CHIP-8, 119–122
  6502, 152–169
OR (|) operator, 240–242
`ORA` instruction (6502), 166
overfitting, 218

P

PackBits algorithm, 75–77
palettes (NES), 180–184
Panic Playdate, 64, 86
`ParserError` class, 40–41
parsers, 9, 41–48
  recursive descent, 41–48
pattern tables (NES), 178–180
`PHA` instruction (6502), 167
`PHP` instruction (6502), 167
picture processing unit. *See* PPU
Pillow library, 66–67, 71–72, 82–84, 99–102
pixels
  in CHIP-8, 128–130
  in dithering, 63–87
  in NES, 176–195
  in stochastic painting, 89–112
`PLA` instruction (6502), 167
`play_sound` property, 125
`PLP` instruction (6502), 167
PPU (picture processing unit), 143, 145, 176–195
  accessing memory, 193–194
  accessing registers, 191–193
  description of, 176–185
  drawing backgrounds, 187–189
  drawing sprites, 189–191
  implementation of, 185–194
PRG RAM (NES), 148–149, 151
PRG ROM (NES), 148–149, 151
`PRINT` statement (BASIC), 25–26
  parsing in NanoBASIC, 44
`PrintStatement` node, 39
production rules, 28–32
program counter, 118
programming languages
  BASIC, 21–59
  Brainfuck, 3–19
  C, 6
  creating your own, 234
  Logo, 56
  NanoBASIC, 21–59
  paradigms of, 23–24
  Turing-complete, 4–5
`putpixel()` method (Pillow), 71
Pygame, 123–124, 126, 146–147, 216, 224–228
Python
  `Counter`, 211
  `match` statement, 12, 48, 132, 151
  performance, 199–200
  `Protocol`, 209–210
  `struct` module, 150
  type hints, 12, 35
  walrus operator, 77

Q

QR codes, 238

R

RAM
  in CHIP-8, 117–119
  in NES, 143, 148–149, 151
RAM size, 143, 151
`random()` function, 103
recursion, 52
recursive descent parsing, 41–48
registers
  in CHIP-8, 117–119
  in 6502, 155–156
regression
  of digits, 224–228
  with KNN, 221–229
  predicting fish weights, 224
resolution
  of CHIP-8 games, 128
  of MacPaint, 72
  of NES, 180
resource fork, 80
retro computing, 63–87
Retro Dither, 85

RETURN statement (BASIC), 26
ReturnStatement node, 39
RETURN_T token, 45
right-shift (>>) operator, 240
RLE (run-length encoding), 73–79
ROL instruction (6502), 167
ROM files, 122, 144–145, 149–152
ROR instruction (6502), 167
RTI instruction (6502), 167–168
RTS instruction (6502), 168
run-length encoding (RLE), 73–79

S

SBC instruction (6502), 168
scanlines, 147, 185
scrolling (NES), 201
SeaTurtle, 56, 234
SEC instruction (6502), 168
SED instruction (6502), 168
SEI instruction (6502), 168
setZN() method, 175
shunting yard algorithm, 48
6502 microprocessor, 143, 152–176
    addressing modes, 153–154, 170–172
    CPU emulation, 152–176
    registers, 155–156
sound
    on CHIP-8, 125
    on NES, 143, 201
speed
    of NES emulator, 199–200
    of stochastic painting, 95
sprites
    on CHIP-8, 129–130
    on NES, 177, 189–191
stack, 155, 175
stack pointer, 156
STA instruction (6502), 168
standard library, 11, 71
Statement node, 37
static type checking, 12
status register (6502), 156, 175–176
step() method, 132, 169, 187
stochastic optimization, 111
stochastic painting, 89–112
    algorithm explanation, 98–99
    implementation of, 99–107
    results of, 107–110
strings, escaping, 58
struct module, 150
STX instruction (6502), 168
STY instruction (6502), 168
SVG (Scalable Vector Graphics), 97–98
switch statement, 131
System 1 (Mac OS), 84

T

take_same() function, 76–77
TAX instruction (6502), 168
TAY instruction (6502), 169
terminal, 29
testing
    Brainfuck interpreter, 15–16
    CHIP-8 VM, 136–137
    dithering, 78–79
    NanoBASIC interpreter, 53–57
    NES emulator, 194–197
    OCR with KNN, 216–217
text processing, 229
Thwaite (game), 182
tiles (NES), 178–180
timers
    in CHIP-8, 118, 125
    in NES, 185
tokenizer, 9, 33–36
Torvalds, Linus, 22
transpiler, 18
TSX instruction (6502), 169
TXA instruction (6502), 169
TXS instruction (6502), 169
TYA instruction (6502), 169
type hints, 12, 35

U

under-fitting, 218
unit tests, 15–16, 78–79, 194–197, 213–217, 224
unpack() function, 150

V

variables in BASIC, 25
variable table, 49–51

vblank (NES), 147, 185
vector graphics, 94, 97–98
virtual machines (VMs), 115–140
    CHIP-8, 115–140
    emulators, compared to, 116
    history of, 116–117
    Java, 116
    runtime environments, 139–140

W

walrus operator (`:=`), 77
Weisbecker, Joseph, 116
`write_memory()` method, 174–175

X

XML, 97
XOR (^) operator, 130, 243–244