

Symbols

- π . *See* pi
- &. *See* concatenation
- *, 82. *See* BasicX math: multiplication
- /. *See* BasicX math: division
- ;. *See* concatenation
- ^. *See* BasicX math: exponentiation
- _. *See* line continuation character
- \. *See* Comments

Songs

- “Happy Birthday to You”, 41
- “Hokey-Pokey”, 184, 195
- “Shave and a Haircut, Six Bits”, 41

2

- 24-hour format, 165

A

- Aaron, Hank, 24, 44
- absolute value. *See* BasicX math
- acceleration, 129
- ADC pin. *See* I/O pins: A-to-D
- Adjustable Sensor Mount, 3, 285, 296, 356
 - attaching to Mouse, 285
- alarm clock, 220
- algorithm, xvi
- alligator clamps, 342
- ambient measurements
 - qualitative, 311–17
 - quantitative, 309, 318–24
- ammeter, 308
- analog-to-digital. *See* A-to-D
- API programming, 38
- arccosine, 108
- arcsine, 108
- arctangent, 108
- arguments, 36, 38, 202–5
 - ByVal vs ByRef, 205
 - passing
 - ByRef, 204
 - ByVal, 203, 215, 226, 290, 299
 - constants, 203
 - numeric literals, 203, 288
- arithmetic
 - progression, 141
 - sequence, 126, 128, 131
 - series, 126, 132, 141
 - sum, 141
- astronomy, 65, 114, 313
- A-to-D

- conversion, 285, 307, 312. *See also* subprograms: I/O: [GetADC](#)
- pins. *See* I/O: A-to-D pins
- audio transducer, 40, 152, 339, 345–46
 - controlling pitch, 345
 - current consumption, 345
 - programming, 345–46
- automaton, vii, 257, 303
- averaging methods, 94, 141, 255, 289–92, 298, 322, 350
 - choosing data types, 291, 322
- Avogadro’s number, 46

B

- bacteria, 155
- baseball, 221
- BasicX, vi, xvii, xviii, 3
 - library, 36
 - multiple users, 5
 - running multiple instances, 10
 - software, 3
 - version, 4
- BasicX buttons
 - Change Directory, 11
 - Clear Serial Window, 9
 - Close File, 313, 315
 - Compile and Run, 13
 - Create File, 313, 315
 - Execute, 14
 - New Project, 11
 - Open Editor, 10
 - Open Port, 6
 - Save Project, 15
 - Stop Processor, 9
 - Undo, 16
- BasicX Editor, xix, 10–11, 16–17
- BasicX installer, 3
- BasicX math. *See also* subprograms
 - absolute value, 98, 155
 - addition, 25, 43, 45, 56
 - advanced, 97–115
 - basic, 43–66
 - dividing by zero, 46
 - division, 45, 56
 - exponentiation, 49, 56, 127, 132
 - Integer division, 48, 56, 131
 - logarithms, **100–103**
 - base-10 algorithm, 101
 - common, 100, 320
 - natural, 98, 100, 133, 320
 - of non-positive numbers, 101
 - properties of, 100
 - with any base, 101
 - multiplication, 45, 56
 - proofs, 61, 79
 - subtraction, 45, 56
 - trigonometry, **103–9**, 149–52
- BasicX projects, xvii, 10, 16
 - basic code file, 29
 - compiling and running, 13–15
 - creating, 10–13, 17
 - map file, 29
 - opening, 27–28
 - project file, 27
 - Save As...issues, 16
 - saving, 4, 15–16, 27, 29
 - storing, xvii
- batteries
 - alkaline, 2, 174, 177, 329
 - charging, 327
 - dropped, 328
 - handling and care, 327–29
 - life, 341
 - mixing, 328
 - Nickel-Cadmium, 2, 327
 - Nickel-Metal Hydride, 2, 177, 327
 - shunt, 329
 - wired in series, 307
- battery pack, 2, 3, 177–78
 - attaching to the Mouse, 174
 - caution, 174, 329
 - maximum voltage, 177, 307, 329
 - polarity, 8
 - safety connector, 8
- baud rate, 7
- BCcomponents, 320
- beat frequency, 345
- binary system, xvi
- biology, 111, 133, 153, 155. *See* population growth
- bits, 67
- blind trust, 52
- Boolean expressions, 77, 258, 274
 - Select Case test, 278
- breadboard, 343
- Brindle, Alex, 243, 324
- BX-24 microcontroller, vi, xvii, 1–3, 8, 70, 171
 - damaging, 176
 - I/O pins. *See* I/O pins
 - internal clock, 159
 - output signals. *See* pulses
 - pin numbers, 175
 - pulses. *See* pulses
 - signal pins, 175
- bytes, 67

C

- calculators vs BX-24, 43, 46, 50, 53, 56, 98, 101, 257
- calibration
 - infrared sensor, 288–89
 - Mouse motion. *See* Mouse: calibration
 - photoresistor, 317
 - thermistor, 317, 320–21
 - constants, 320

- Call, 36, 85, 160, 184, 202
 - cameras, 284
 - capture data to a file, 313, 315
 - Case, 274
 - Case Else, 274–76, 278
 - catenary, 112, 155, 220
 - CCD array, 283–84
 - Cha-Cha, 187, 195
 - Challenge Problems, xv
 - chemistry, 111, 112, 218
 - chess, programming is like, 324
 - Chip Dialog Box, 34
 - CIA Factbook, 99
 - civics, 266
 - Clemson University, vii
 - Code Editor. *See* BasicX Editor
 - Code Window. *See* BasicX Editor
 - coin toss, 250, 261–62, 275, 281
 - COM port, 5, 6, 7
 - comments, xvi, **xix–xx**, 15, 18, 43
 - motivation for using, xix
 - communications method, 21
 - computer arithmetic. *See* BasicX math
 - computer model, 129, 144–46, 149
 - resolution, 135, 146–49, 155
 - time step, 147–49
 - concatenation, **23–24**, 25, 45, 70, 132, 277
 - of lines of code, 25–26
 - Const, **72**, 186
 - constants. *See also* Const
 - common, 72
 - declaring, 72, 75
 - echoing values, 75
 - global, 200, 216–17, 301
 - declaring, 200
 - location, 72
 - module-level, 200
 - naming, 200
 - static, 200
 - using, 75, 121
 - continuous loops. *See* infinite loops
 - cos (cosine), 105
 - cosecant, 113
 - cosine wave, 152
 - cotangent, 113
 - counters, 118, 136, 141
 - discrete data types, 121, 125, 130
 - limits, 120, 136
 - counting backwards, 121, 123
 - counting forwards, 119, 122
 - counting increments. *See* loop increments
 - CStr. *See* subprograms: communications
 - cube roots, 53, 56, 133, 154
 - current. *See* electrical current
 - curve fitting. *See* calibration
- D**
- data types, **67–68**, 68
 - Boolean, 84–85
 - Byte, 82, 84
 - conversions, 81–85. *See also* subprograms: conversions
 - discrete, 119, 122, 127, 168
 - Integer, 47, 67, 82, 87
 - Long, 85, 121, 128, 168
 - of counters, 119, 122
 - of functions, 215
 - Single, 38, 47, 67, 82–83, 85–86, 88, 128
 - storage size, 68
 - String, 67, 81–82, 85
 - value range, 68, 119, 121, 155, 291
 - date and time operations, 159–69
 - dB (decibels), 102, 341
 - DB-9 serial cable. *See* serial cable
 - Debug.Print. *See* subprograms: communications
 - decimal system, xvi
 - definite loops, 117
 - degrees, 104
 - degrees to radians, 104, 218
 - Delay. *See* subprograms: tasking
 - determinants, 62, 80
 - dice, 249, 255, 268, 271, 276, 281
 - digital clock, 165–67, 282
 - digitals, 284
 - Dim, **69**, 71–72, 120, 137, 201
 - DiMaggio, Joe, 25
 - discrete data types. *See* data types
 - discriminant, 282
 - Distributive Property of Multiplication, 62
 - Do-Loops, 117, **135–58**, 344. *See also* loops
 - ending
 - with Exit Do, 271–72, 277, 303
 - with Loop, 136
 - infinite, 135, 136, 142
 - Until. *See* Until
 - While. *See* While
 - Download Port, 6
 - Downloader, xvii, 6, 10
 - problems minimizing, 10, 14, 19
- E**
- e*. *See* BasicX math; exponential growth
 - calculating, 133
 - earth science, 112–113, 313, 322, 325
 - earthquakes, 112–113
 - EEPROM, xvii, xix, 14–15, 28–29, 67
 - electrical “short”, 176, 287, 339
 - electrical current, 319, 340–41
 - defined, 308
 - human tolerances, 308
 - electrical ground, 175
 - electrical potential, 307, 310
 - electrical power, 341
 - electrical pulses. *See* pulses
 - electronics, vii, 1, 63, 103, 307–12
 - elegant code, 291, 304
 - Else, 261–63, 278
 - ElseIf, 263–67, 278
 - multiple logic expressions, 263
 - End Function, 215
 - End If, 258–59, 278
 - omitting, 260
 - End Select, 274, 278
 - End Sub, 18, 21, 198
 - engineering, 171
 - environment parameter, 22
 - equivalent resistance, 319
 - error trapping, 357
 - errors
 - associated with logarithms, 101
 - associated with absolute value, 98
 - associated with For-To-Next loops, 120, 260
 - Compile, 20, 23–24, 38, 44, 47–49, 52, 70–71, 73, 75, 83, 120, 122, 185, 251, 260
 - compiling while minimized, 10
 - Division by Zero, 46
 - Download, 28
 - due to negative exponents, 50
 - due to processor rounding, 46, 50, 55, 107, 297
 - Duplicate Name Detected, 137
 - Exponentiation Requires Float Mantissa, 49, 130
 - Halt, 28
 - Halting, 14
 - Installation, 5, 331
 - Missing Right Parenthesis, 160, 184
 - negative roots (nonexistent), 53, 98
 - run-time, 55, 291
 - Semicolon expected, 23
 - Serial Data, 9, 14
 - Syntax, 24, 44, 48, 50, 70, 160, 260
 - Type Mismatch, 24, 44, 47, 70, 75, 77, 98, 121, 123, 129–30
 - Unknown Identifier, 71, 118
 - Excel
 - converting text to columns, 151, 316
 - curve fitting (trendline), 296
 - importing data from Downloader window, 150, 155
 - importing data from file, 313, 316
 - plotting data with, 149, 154–56, 218–19, 294, 296, 316
 - tutorial, 151, 156, 315, 325
 - Exit Do. *See* Do-Loops
 - Exit For. *See* For-To-Next loops
 - Exit Sub. *See* subprograms
 - exponential growth, 98, 99, 144
 - exponents, 49
 - external data, 283
 - “eyes”, 283
- F**
- F5 key, 13, 19
 - factorials, 62, 133, 158
 - Fibonacci Series, 134, 155, 170
 - file extensions
 - .bas, 29
 - .bxp, 27
 - .mpp, 29
 - .txt, 313, 315–16
 - finances
 - compounded interest, 63, 155, 218
 - IRA, 63
 - loan payment, 64
 - mutual fund, 63, 134, 158

floating numbers. *See* data types:
Singles

floating-point division. *See* BasicX
math: division

fonts, xviii–xix, 12

football, 306

formatting. *See also* subprograms
code, 26
output data, 82, 130, 154

For-To-Next loops, 39, **117–34**, 135,
290, 344. *See also* loops
ending
with **Exit For**, 270–71
with **Next**, 118
step size. *See* **Step**
using counters outside the loop,
120

FreqOut. *See* subprograms: I/O
frequency, 149, 345

functions, 197, 215–16, 322
beginning. *See* **Public
Function**
built-in. *See* subprograms
data types of, 215
ending. *See* **End Function**
for infrared sensor, 290
syntax, 215

G

George School, 243

Gleick, James, 281

Gripper, 172, 306, 353
programming, 358
pulse widths, 357–58
servo, 357
tight grip, 358

Grumbach, James, 94

H

Hello World!, 10

Hertz, Heinrich, 149

Hodgin, Michael, 331

horizon scanning, 114

hyperbolic cosine, 112, 134

hyperbolic sine, 112

Hz (Hertz), 149, 345

I

I/O pins, xvii, 34
analog input, 34, 284
A-to-D, 286, 307, 312
digital resolution, 312, 318
digital signal, 34. *See also* sensors
high output, 35–36
initial state, 34
low output, 35
maximum current, 340–41
numbering, 175, 178, 286
output, 34
signal pins. *See* BX-24
microcontroller; RAMB
states, 35, 36, 184

If-Then statements. *See* logic
statements

imaginary numbers, 53

incremental operations, 127

inequalities, 137, 138, 259

infinite loops.¹ *See* continuous loops

infrared (IR) range finder. *See*
sensors: infrared.

input, xvi, 283

Integer division, 155

intelligence, 67, 257

inverse trig functions, 108

Is, 274

J

Joule heating, 341

K

Karaoke, 41

Kelly, Catie, 243, 324

Kelvin scale, 219, 321

keywords, xv, 68, 200, 337

L

Law of Cosines, 106, 114, 218

Law of Sines, 106, 218

LEDs (onboard), **33–37**, 73, 131
as range indicators, 299, 304, 306
as temperature indicators, 326
blinking, 37, 39, 124–25, 142,
153, 170, 195, 199, 242
programming with constants, 200
programming with subprograms,
202, 205, 219
toggling with code, 36–38
toggling with software, 35
using to display numbers, 131,
220, 280

Leibniz, Gottfried, 134

light-emitting diodes. *See* LEDs
(onboard); RAMB: LEDs

line continuation character, 25

line following. *See* Mouse: line
following

logarithms. *See* BasicX math

logic block. *See* logic statements

logic operators
And, 250–51, 268
Or, 251–53, 269
use of parentheses, 251, 269

logic statements
If-Then, 245, 257–73, 356
beginning with **If**, 258
Else. *See* **Else**
ElseIf. *See* **ElseIf**
ending. *See* **End If**
real world analogy, 257–58
statements vs blocks, 259, 266
syntax (full), 269
syntax (simple), 258
use of parentheses, 258, 269

If-Then vs Select Case, 275

Select Case, 257, 274–78
Case. *See* **Case**
Case Else. *See* **Case
Else**
ending. *See* **End Select**
syntax, 274
test expression, 274–75

long division, 88

loops, 39, 117, 135. *See also* For-To-
Next loops and Do-Loops
and robot motion, 187, 190, 192
definite, 135. *See also* For-To-
Next loops
Do vs For-To-Next, 126, 135, 153
increments, 122–23, 139
indefinite, 135 *See also* Do-Loops
infinite. *See* Do-Loops
nested, 126, 134, 158, 166, 191,
196, 218, 226, 314
summing using, 126

M

machine language, xvi, 6, 13, 19
Main, 18, 21, 197–98, 259

Makeit, Willie, 30, 69

mantissa. *See* operand

math phobia, 43

mathematical model. *See* computer
model

mathematics. *See* BasicX math
Maximum String Length, 22, 23, 29
maze navigation. *See* Mouse: maze
navigation

modular programming, 197, 199

Monitor Port, 7

Morse code, 41

motherboard. *See* RAMB

motion detector, 306, 325

Mouse, vii, 3, 172
attaching servos, 172–73
attaching the RAMB, 173
building, 172–74
calibration
linear
coarse, 209, 212
precision, 236–40
turning
coarse, 194, 210, 212
precision, 238–40

dancing, 187, 195

drawing, 196, 243, 324

line following, 195, 326

linear motion
coarse, 184–89, 192, 208, 225
precision, 229–34, 304–6

maze navigation, 196, 220, 305

motion
and servo rotation, 184–86,
189–92
choppy, 188, 223, 303
constants, 186–89, 228–34
continuous, 192, 229–34
inconsistencies, 191, 193, 212,
225
linear. *See* Mouse: linear
motion
random, 253–54, 273, 282, 304
template, 241
turning. *See* Mouse: turning
using procedures, 208–11, 273

obstacle avoidance, 305–6

path tracing, 195–96, 211, 214,
221, 241, 242

phenolic ball, 173

photophobic, 326

¹ For the *real* definition, see Do-
Loops

- phototropic, 326
 - programming. *See* servo programming
 - racing, 195
 - servo pin numbers, 178, 186
 - speed control. *See* servo programming
 - tabletop navigation, 293, 305
 - tail wheel, 173
 - troubleshooting, 193
 - turning
 - 360°, 191, 194, 213
 - center-pivot vs sidewinder, 192
 - coarse, 189–92, 210, 225
 - precision, 235–36
 - sidewinder, 191–92
- N**
- natural growth rate, 142
 - needle in a haystack, 252–53
 - negative numbers, 48
 - nested loops. *See* loops
 - NetMedia, vi, xvii, 1, 3, 36, 197, 215, 245, 259
 - network, 4
 - issues, 4–5, 331–35
 - multiple users, 331
 - neutral position. *See* servos: center position
 - Newton, Isaac, 281
 - non-linear devices, 296, 309, 317
 - Notepad, 10, 16, 301
 - null string, 22
 - numeric literals, 75, 160, 188, 200, 203
 - numerical rounding, 49, 82, **86–91**. *See also* subprograms
- O**
- obstacle avoidance. *See* Mouse: obstacle avoidance
 - Ohm’s Law, 319, 340
 - ohmmeter, 308, 320
 - Olson, Ian, 90
 - operands, 49. *See also* arguments
 - operating systems, 1
 - [Option Explicit](#), 18
 - Order of Operations, 54, 56
 - other programming languages, xviii
 - output, xvi, 17
- P**
- pain threshold, 102
 - painful experience, 8
 - parabolas, 62
 - parameters, 198. *See also* arguments
 - passing, 203, 204
 - passing with globals, 201
 - passwords, 134, 281
 - path tracing. *See* Mouse: path tracing
 - Patton, Brian, vi
 - pausing. *See* [Delay](#)
 - PC vs microcontroller programming, 37, 171, 283, 307
 - Peins, Dave, xiv
 - Pepys, Samuel, 281
 - perfect squares, 132, 154, 270, 280
 - period, 179
 - pH scale, 112
 - photoresistor. *See* sensors: photoresistor
 - physics
 - energy, 156, 310
 - kinematics, 63, 129, 133, 139, 146, 155
 - pendulum, 62
 - radioactive decay, 98
 - RC circuits, 103
 - rocket flight, 146, 155
 - sound, 102, 152, 219, 341, 345
 - beats, 152
 - levels, 342
 - waves, 152
 - time constant, 103
 - water flow, 65
 - pi, 104
 - calculating, 134, 154
 - declaring, 75, 104, 216
 - echoing, 75
 - using, 47, 75, 78, 104, 149
 - piezo buzzer, 35, 40, 134, 155, 170, 219, 221, 242, 304, 339
 - attaching to the Mouse, 343
 - current consumption, 341
 - operating voltage, 341
 - programming, 343–44
 - volume-control, 343
 - wires, 342
 - ping-pong, 304
 - pins. *See* I/O pins
 - plastic insulators, 173, 310
 - population growth, 98–99, 110, 133, 142–45, 155, 157
 - Table, 110, 157
 - power function, 51
 - prime numbers, 281
 - Print. *See* [Debug.Print](#)
 - printing, 18, 44, 125
 - probability, 262, 268
 - procedures, 197–214
 - beginning. *See* [Public Sub](#)
 - built-in. *See* subprograms
 - ending. *See* [End Sub](#)
 - for Mouse motion, 213, 226, 230, 232, 234–35, 237, 240–42
 - omitting parentheses, 198
 - syntax, 205
 - processor rounding, 51, 55
 - processor speed, 36, 119, 125, 162, 164, 170
 - Progress Bar, 6, 9, 14
 - proximity sensor, 284
 - pseudocode, 258, 314
 - [Public Function](#), 215
 - [Public Sub](#), 18, 21, 197
 - pulse trains, 179, 181, 183, 223
 - pulse width modulation. *See* PWM
 - pulse widths, 179
 - declaring with constants, 186, 228–34
 - for your Gripper, 358
 - for your Mouse, 194, 213, 228, 234, 238–39
 - for your Turret, 356
 - range, 180, 224, 228, 234, 348, 353–57
 - servo positions, 180, 223
 - units, 179
- pulses, 223
 - controlling servos with, 179–93
 - duration. *See* pulse widths
 - high, 179, 184
 - low, 179
 - measured, 179
 - necessary delay between, 185, 226
- PWM, 223, 225–36
 - precision calibration. *See* Mouse: calibration
 - precision Mouse motion. *See* Mouse: linear motion
 - speed control. *See* servo programming
- Pythagorean theorem, 52, 105, 115
- Q**
- QBasic, vi, xvii
 - quadratic equation, 61, 282
 - quadratic formula, 61, 282
- R**
- R/C models, 171, 180
 - radar detector, 306
 - radians, 104, 106, 108
 - radians to degrees, 104, 218
 - RAM, xvi, 14, 68
 - RAMB, vii, 1–2, 3, 339
 - attaching to the Mouse, 173
 - communicating with, 5
 - damaging, 176
 - ground pins, 175
 - I/O pins. *See* I/O pins
 - LEDs, 9, 13, 33
 - pin numbers, 175
 - pin rows, 175
 - power, 177, 339–40
 - power pins, 175
 - power receptacle, 8
 - power switch, 9, 28
 - regulated voltage, 177, 310
 - reset button, 14, 36
 - signal pins, 175, 286, 311
 - unregulated voltage, 177
 - voltage regulator, 177
 - random numbers, 280, 282
 - creating, 245–49
 - range of, 248, 255
 - [Rnd](#). *See* subprograms: math seeding, 245
 - using, 249–54
 - range equation, 297, 303, 349
 - programming with, 298
 - range finder. *See* sensors: infrared
 - real time clock. *See* RTC
 - reference resistance, 321
 - regedit, 332
 - Registry Editor, 332
 - registry keys, 331
 - REMark statements. *See* Comments
 - remote sensing, 284
 - reserved keywords. *See* keywords
 - resident program, xvi, 14
 - resistance, 320
 - defined, 308
 - measuring

- directly, 308
 - indirectly, 308–9, 318–20
 - resistors, 308–9
 - color code, 308
 - fixed, 308, 319
 - circuit symbol, 308
 - variable, 309, 319
 - circuit symbol, 309
 - photoresistors, 309. *See also* sensors: photoresistor
 - potentiometers, 309, 318, 343
 - thermistors, 309. *See also* sensors: thermistor
 - wired in series, 310, 319
 - resonant frequency, 341, 345
 - rice on a chessboard, 127
 - Rnd.** *See* subprograms: math
 - Roach, 3
 - Robodyssey Advanced Motherboard. *See* RAMB
 - Robodyssey Systems, vi, **vii**, xiv, 1, 171, 284–85, 309, 318, 347, 353, 356
 - robot, vii, 3, 43, 171
 - dancing. *See* Mouse: dancing
 - motion. *See* Mouse
 - navigation, 114
 - propulsion. *See* servos
 - walking, 3, 347
 - wheeled, 3
 - robota*, 117
 - role-playing game, 277
 - roots of negative numbers, 53, 98
 - rounding subprograms. *See* subprograms: rounding
 - Royal Society, 282
 - RTC, 95, 159–69, 303, 313–14
 - and Mouse motion, 195
 - resetting, 160
 - resolution, 38, 160, 163, 314
 - setting, 160
 - Timer.** *See* subprograms: RTC
- S**
- sand fleas, 133, 155
 - scientific notation, 46
 - search and rescue, 305
 - secant, 113
 - SeedPRNG**, 246
 - Select Case statements. *See* logic statements
 - Self Tests, xv
 - sensors, xvi
 - force, 309
 - infrared, 3, 114, 172, 356
 - accuracy, 289, 349, 352
 - ambiguous readings, 294–96, 306
 - calibration. *See* calibration
 - connecting wires, 286
 - continuous output, 292–93
 - data types, 287–88
 - digital signals, 284, 288–89, 294
 - averaging, 289–91, 352
 - effects of
 - color, 351
 - distance, 352
 - lighting, 351
 - object size, 352
 - orientation, 350
 - GP2D12, 283
 - limitations, 294–96, 349–52
 - mounting. *See* Adjustable Sensor Bracket
 - powering, 284, 286–87
 - range finding, 296–300, 304
 - receiver, 283–84
 - resolution, 352
 - transmitter LED, 283–84
 - faint glow, 284, 287
 - uncertainty, 290, 349
 - photoresistor, 309, 311–17, 325. *See also* ambient measurements calibration. *See* calibration color sensing, 326 digital signals, 312–16 sensor pack, 309 thermistor, 309, 317–24, 325. *See also* ambient measurements; temperature conversion calibration. *See* calibration conversion equation, 320 digital signals, 317 precision, 325 uncertainty, 321
 - serial cable, xvii, 2, 5, 9, 13, 28, 173
 - servo programming
 - advanced, 225–42
 - basic, 184–94, 225
 - Gripper, 358
 - Mouse. *See* Mouse
 - necessary delay, 185, 188–89, 225–26
 - omitting, 303
 - speed control
 - using **Delay**, 188–89, 208, 225
 - using PWM, 223, 226, 229–36, 242
 - Turret Mount, 356–57
 - using constants, 186–89, 228–34
 - servomotors. *See* servos
 - servos, 171–96
 - attaching to the Mouse, 172–73
 - Blue Bird BMS-380, 174, 177, 180, 347–48, 353
 - built-in processor, 179, 224
 - calibration. *See* Mouse: calibration
 - caution using, 171–72, 177, 180, 184, 187–88, 354–55
 - center position, 180, 182–83, 224, 227–28, 353, 355
 - finding, 225–29, 354
 - comparisons, 180, 347–48
 - connecting to the RAMB, 178, 348, 353
 - connecting wires, 172
 - continuous rotation, 183, 224, 347
 - control arm, 353
 - controlling with pulses, 179–93, 223–41
 - for Gripper, 353
 - for Mouse, 180
 - for Turret Mount, 353
 - Futaba S3004, 172, 177, 180–81, 184–86, 224, 347–48
 - HiTec HS-311, 347–48, 353, 357
 - limitations, 180, 347–48, 353–56
 - modified, 180, 183–84, 224, 347
 - modified vs unmodified, 180
 - operating voltage, 177, 353
 - positions, 180
 - power, 175, 348
 - programming. *See* servo programming
 - pulse widths. *See* pulse widths pulses. *See* pulses
 - purpose, 172
 - rewiring, 183
 - rotation
 - and Mouse motion, 184–86, 189–92
 - direction, 180–83, 223, 348, 355
 - speed, 181–82, 224, 348
 - size, 348
 - speed control. *See* servo programming
 - torque, 196, 348
 - unmodified, 180–82, 224, 347, 353
 - caution using, 242, 347, 353
 - maximum position, 348, 355
 - minimum position, 348, 354
 - quick slewing, 356
 - uses. *See* Gripper; Mouse; R/C models; Turret Mount; winch
 - Sharp GP2D12 sensor. *See* sensors: infrared
 - shunt, 329
 - sin (sine), 105
 - sine wave, 149, 151
 - Single precision numbers. *See* data types: Singles
 - solid-state devices, 33
 - square roots, 51, 83, 132, 154
 - square wave, 179
 - standard deviation, 349
 - standoffs, 173, 227, 285
 - starting directory, 6
 - Status Bar, 14
 - Status Window, xvii, 6, 12–13, 19
 - Step**, 121–22
 - default, 121
 - string literals, 21, 26, 69
 - strings, 19, 21, 24, 44
 - subprograms, 18, 36, 197–221
 - arguments. *See* arguments calling. *See* **Call** communications
 - Debug.Print**, 12, **18–23**, 44, 81, 125, 162
 - conversion, 92
 - CBool**, 84
 - CByte**, 84, 166
 - CInt**, 82
 - CLng**, 85
 - CSng**, 83, 130
 - CStr**, 24–25, 44, 70, 81
 - Fmt**, 82, 84, 123, 130
 - ValueS**, 85, 282
 - creating, 38, 197, 199, 215

- declaring. *See* [Public Sub](#);
[Public Function](#)
 - ending. *See* [End Sub](#); [End Function](#)
 - ending prematurely with [Exit Sub](#), 272
 - formatting
 - [CStr](#) vs [Fmt](#), 82
 - [Fmt](#), 82, 84, 123
 - functions. *See also* functions
 - I/O
 - [FreqOut](#), 345–46
 - [GetADC](#), 287–88, 312, 314, 318
 - [PulseOut](#), 184–86, 189–92, 225, 237, 345, 354, 358
 - [PutPin](#), 36–37, 344–45
 - importing, 301
 - [Main](#). *See* [Main](#)
 - math
 - [Abs](#), 98
 - [ACos](#), 108
 - [ASin](#), 108
 - [Atn](#), 108
 - [ATn](#) vs [ATan](#), 108
 - [Cos](#), 106
 - [Exp](#), 99, 144–5
 - [Exp10](#), 99
 - [Log](#), 100, 320
 - [Log10](#), 100, 320
 - [Pow](#), 49, 51
 - [Randomize](#), 245–46
 - confusion using, 245–46
 - [Rnd](#), 246–49, 261
 - [Sin](#), 106, 154
 - [Sqr](#), 51, 56, 98
 - [Tan](#), 106
 - operands, 24
 - [Private](#), 198, 205
 - procedures. *See* procedures
 - procedures vs functions, 217
 - [Public](#), 198
 - [Public](#) vs [Private](#), 198
 - rounding, 92
 - [CBool](#), 89
 - [CByte](#), 89
 - [CInt](#), 89, 248
 - [CInt](#) vs [Fix](#), 92
 - [CLng](#), 89
 - [Fix](#), 86
 - [FixB](#), 88, 166
 - [FixI](#), 87, 247, 261
 - [FixL](#), 88
 - RTC
 - [GetDate](#), 160
 - [GetDayOfWeek](#), 162
 - [GetTime](#), 160
 - [GetTimeStamp](#), 160
 - [PutDate](#), 160
 - [PutTime](#), 160, 168
 - [PutTimeStamp](#), 160, 168
 - [Timer](#), 168–69, 220, 303
 - tasking
 - [Delay](#), 33, 37–38, 119, 163, 344
 - [Delay](#) (homemade), 169, 220
 - [Sleep](#), 38
 - subroutines. *See* subprograms
 - Success, 85, 282
 - summing methods, 127–28, 136, 141, 143, 263
 - Sumo wrestling, 326
 - superposition of waves, 152
 - symbolic code, xvi
 - system library. *See* subprograms
- T**
- tabletop navigation. *See* [Mouse](#):
tabletop navigation
 - [tan](#) (tangent), 105
 - temperature conversion, 216, 218, 321–24
 - terminology, xvi, xvii
 - thermistor. *See* sensors: thermistor
 - thinking machine, 257, 278, 299
 - threshold of human hearing, 102
 - [Timer](#). *See* subprograms: RTC
 - [To](#), 277
 - trapezoids, 62
 - triangulation method, 284
 - trig identity, 105
 - trigonometry. *See* [BasicX](#) math
 - truncation, 86
 - Turret Mount, 172, 295, 306, 353, 356
 - calibration. *See* [Mouse](#):
calibration
 - panning, 352, 356
 - programming, 356
 - range, 357
 - servo, 356
- U**
- unit circle, 104
 - [Until](#), 136, 140
 - location of, 137, 140–41
 - updated information, 325
 - USB, 2, 5
- V**
- [ValueS](#). *See also* Success
 - variables, 68, 129
 - Boolean, 77, 85
 - declaring, 69, 75
 - on one line, 72
 - with subprograms, 203, 215
 - echoing values, 75
 - global, 201, 217, 226
 - declaring, 201
 - [SeedPRNG](#), 246
 - initializing, 69, 75, 202
 - limitations, 68
 - local, 199, 200
 - module-level, 200
 - naming, 68, 75, 84
 - reassigning, 71
 - re-declaring, 199
 - static, 200
 - using, 75
 - Why bother?, 76
- VB Boxes, xviii
- VDB, 310
 - attaching to [Mouse](#), 310
 - connecting to the RAMB, 311
 - terminal block, 311
- Visual Basic, vi, xvii, xviii
- voltage. *See also* RAMB; BX-24; servos
 - analog, 284, 312
 - analog vs digital, 284
 - DC, 341
 - definition, 307
 - measuring with the BX-24, 287–88, 308
 - range of I/O pins, 312, 318
 - range of IR sensor, 284
 - regulated, 177
 - unregulated, 177
- voltage difference (drop), 307
- voltage divider board. *See* VDB
- voltage dividers, 307, 309–10. *See also* VDB; sensors:
 - photoresistors, thermistors
 - theory, 318–20
- voltage potential. *See* electrical potential
- voltage regulator. *See* RAMB
- voltmeter, 307, 310
- W**
- [While](#), 138, 140
 - location of, 140–41
 - widgets, 91
 - Willie Makeit, 30
 - winch, 172, 196
 - wiring
 - attaching to the RAMB, 176
 - color conventions, 176, 286, 311
 - light-to-dark, 176
 - working directory, 4, 6, 10, 29, 34
 - Wunderle, Frank, 331
- Y**
- Yahtzee, 249, 255
 - Yoon, Jake, 306