index

A

- Accelerometer, 6, 26. *See also* Motion Sensor
- action variable, 84, 85, 86
- add() function, 21
- adjusted angle variable, 52
- ambient mode, Color Sensor, 26, 27
- and logical operator, 14
- angles
  - for turns, 50–54
  - when tracing shapes, 55
- append() method, lists, 19
- appliances, smart, 99
- App library, 3
- arithmetic operators, 13
- art, making
  - framed, 106–107
  - kinetic sculpture, 107
  - overview, 105
  - spiro gyro, 105–106

B

- backing up, during obstacle avoidance, 61
- ball launcher, 107–108
- ball over the wall project, 108–109
- bang-bang (two-step or on-off) controllers, 62, 64, 69–72
- base_power variable, 73, 74, 75
- beating heart, displaying on light matrix, 15
- bed, motorized, 99
- beep() method, 2, 3
- better counter project, 22
- bevel gears, 38
- biomimicry
  - bugs, 113
  - cockroach, 111–112
  - overview, 111
  - robo pet, 112–113
- Booleans, in Python, 17
- Bop It–style game, 84–86
- break command, 12
- bugs, 113
- bumped buttons, 5
- burglar alarm, useless, 102
- buttons, Hub
  - controlling, 5–6
  - Cookie Clicker–type game, 79
  - reaction timer game, 81–84

C

- calling functions, 21
- cam, 44, 45
- camshaft, 44
- capitalization, in Python, 6
- catapult, 107–108
- cat companion, 97
- ceil() function, 31
- center button light, Hub, 5
- check_response() function, 90
- check response My Block, 85, 88
- children, useful inventions for, 95–97
- choose_next_action() function, 90
- choose action My Block, 84–85
- choose next action My Block, 87
- circular motion, turning into linear motion, 44, 45
- closed shapes, tracing, 56
- cockroach, 111–112
- color, Hub status light, 5, 6, 7
- Color mode, Color Sensor, 71
- Color Sensor. *See also* line following
  - cockroach, 112
  - color xylophone, 104
  - framed art, 106
  - line following, 69–71
  - music box, 104
  - obstacle avoidance and line following, 59–60
  - programming, 26–27, 33
  - theremin, 103
- color xylophone, 104
- column input, draw column block, 80
- combination lock, 110–111
- comments
  - in Python, 6
  - in starting code, 2
- comparison operators, 13–14, 27
- competitive clicking game. *See* Cookie Clicker–type game
- compound gearing, 36, 38, 47
- conditional statements, 11–12
- console, Python
  - displaying lists, 19
  - displaying sensor readings, 27-28, 30
  - golden cookie project, 94
  - identifying data types, 17-18
  - overview, 1–2
  - printing to, 7
  - TypeError, 30
- control structures, Python
  - conditional statements, 11–12
  - looping, 9–11
  - overview, 9
  - pausing program, 9
- cook, robot, 97–98
- Cookie Clicker–type game
  - golden cookie project, 93–94
  - one-player version, 79–80
  - overview, 79
  - two-player version, 80–81
- countdown project, 15
- counter(s)
  - cycling through lists, 20
  - for loop, 10–11
  - projects, 22
  - updating variables, 18–19
- coupler, four-bar linkage, 41, 43, 44
- coupler curve, 43
- crank, four-bar linkage, 41, 43, 44
- crank-rocker mechanism, 41, 42–43
- crank-slider mechanism, 44
- C to C project, 15
- cycling through lists, 19–20, 21

D

- data types
  - Booleans, 17
  - converting between floats and integers, 30–32
  - floating-point numbers, 17, 18, 30–32
  - identifying, 17–18
  - integers, 17, 18, 30–32
  - NoneType, 17
  - overview, 17
  - strings, 4, 7, 17
- debugging, 6, 7
- definitions, function, 20
- degrees, turning specific number of, 52–54
- dependable pal robot, 96–97
- derivative controller, 66–67, 73–74
- derivative variable, 67
- dice roll project, 91
- differential assemblies, 40–41
- disperser, seed, 100–101
- display_sequence() function, 90

display score My Block, 88
display sequence My Block, 87
Distance Sensor. *See also* obstacle avoidance
  ball over the wall project, 108–109
  obstacle avoidance and line following, 59–60
  programming, 27–28
  tabletop robot, 109
  theremin, 103
division operator (/), 30–31, 32
double-bevel gears, 38, 39, 40
draw_value() function, 81
draw column My Block, 80
driver, in gear train, 35, 36
drive shaft, 40
duration, beep, 3

E

elif statement, 12
equal signs (=), in Python, 14, 26
equal to operator (==), in Python, 14, 26
error
  derivative controller, 66
  integral controller, 67, 68
  line following with PID controller, 74
  line following with proportional control, 72–73
  proportional controller, 64–65

F

files, playing sound, 3
fish feeder, 97
float() function, 32
floating-point numbers, 17, 18, 30–32
floor division operator (//), 31–32
floor() function, 31, 81
follower, in gear train, 35, 36
Force Sensor, programming, 30
for loops
  Cookie Clicker–type game, 81
  cycling through lists, 20
  overview, 9, 10–11
  projects, 15
  tracing shapes, 54–55
four-bar linkages
  crank-rocker, 42–43
  crank-slider, 44
  overview, 41
  parallelogram linkage, 44
frame, four-bar linkage, 41
framed art, 106–107
functions
  adding parameters, 21
  calling, 21
  defining, 20
  moody, creating, 22
  overview, 20
  returning values, 21–22
furniture, smart, 99

G

gainD variable, 67, 75
gainI variable, 68, 74, 75
gainP variable, 73, 75
gain value
  derivative controller, 66
  integral controller, 67
  proportional controller, 66
game_over variable, 83, 85, 86
game_running variable, 90
games, programming
  Bop It–style, 84–86
  Cookie Clicker–type, 79–81
  memory, 86–90
  overview, 79
  projects, 91–94
  reaction timer, 81–84
gearing down, 36, 37
gearing up, 36, 37
gear ratio, 36
gears
  compound gearing, 36, 38
  differential assemblies, 40–41
  gearing up and gearing down, 36, 37
  overview, 35
  projects, 47
  turntables, 39–40
  types of, 38–41
gear trains, 35, 36, 47
gentle-stop program, 64–68
gestures, Hub
  detecting, 28
  getting, 29
get_ambient_light() method, 26, 27
get_color() method, 26
get_distance_cm() method, 28
get_distance_inches() method, 28
get_distance_percentage() method, 28
get_force_newton() method, 30
get_force_percentage() method, 30
get_gesture() method, 29
get_orientation() method, 29
get_pitch_angle() method, 29
get_reflected_light() method, 26–27
get_roll_angle() method, 29
get_volume() method, 7
get_yaw_angle() method, 29
golden cookie project, 93–94
grabber, for children, 95
Gyro Sensor. *See also* Motion Sensor
  executing turns, 49–54
  overview, 6, 26, 49
  proportional control with, 78
  spiro gyro, 105–106
  tracing shapes, 54–57

H

hashtag (#), in Python, 2, 6
heart, displaying on light matrix, 6
Help Center, MINDSTORMS App, 1, 2
high-shelf reacher, 95
household inventions, 97–100
houseplant helper, 101
Hub
  Bop It–style game, 84–86
  controlling outputs and inputs, 3–6, 7
  Cookie Clicker–type game, 79–80
  dice roll project, 91
  Motion Sensor, 28–29
  printing volume level to console, 7
  reaction timer game, 81–84
  tricky dice project, 92
  using yaw readings to execute turns, 49

I

idlers, in gear train, 35, 36
if/elif/else statement, 12, 15
if/else statement, 12
if statement
  logical operators used with, 14
  overview, 11–12
  reaction timer game, 83
  stopping while loop with, 12
if then else block, 53
Image() function, 91
imports, in starting code, 2
indentation, in Python, 6, 10, 12
index numbers, in lists, 19
index variable, draw column block, 80
initialize movement block, 54
inputs, controlling Hub, 3–6
insects, 113
integers, 17, 18, 30–32
integral controller, 67–68, 73–74
integral variable, 68

interior angle of shapes, 55
inventions
    for children, 95–97
    household, 97–100
    overview, 95
    for plants, 100–101
    useless, 101–102
is_pressed() method, 6, 30

J

juice-box crusher, 100

K

kinetic sculpture, 107

L

last_error variable, 67, 68
left turns, programming, 51
LEGO Education SPIKE Prime set
    Force Sensor, 30
    motors in, 23
    overview, xv
    rack-and-pinion mechanisms, 46
    starting code, 2
LEGO MINDSTORMS Robot Inventor set
    differential assemblies, 40
    gear types in, 38
    motors in, 23
    overview, xv
    starting code, 2
    turntables, 39–40
len() function, 19, 20, 33
length input, move polygon block, 56
libraries, 2
lifting mechanisms, 44
light matrix
    Bop It–style game, 84–86
    controlling Hub, 3–4, 7, 15
    Cookie Clicker–type game, 79–81
    dice roll project, 91
    reaction timer game, 81–84
linear motion, turning rotary motion into,
        44, 45
line following
    bang-bang approach, 69–72
    overview, 59–60, 68–69
    with PID controller, 73–74
    projects, 76–78
    with proportional control, 72–73
    tuning PID controller, 74–75
lists
    cycling through, 19–20, 21
    memory game, 87–89
    overview, 18, 19
    updating, 19
    using index numbers, 19
loaded die, 92
lock, creating, 110
logical operators, 14
loops. *See also* for loops; while loops
    controlling motor, 24
    overview, 9
    tracing shapes, 54–55
lowercase, in Python, 6

M

math library, 31
mechanisms
    cams, 44
    four-bar linkage, 41–44
    overview, 35, 41
    projects, 47
    rack and pinion, 46
    ratchet and pawl, 44–45
    scissor lift, 45–46
memory game, 86–90
methods
    data types for parameters, 30–32
    versus functions, 20
    overview, 2
MIDI (Musical Instrument Digital Interface), 3
MINDSTORMS App. *See also* Python programming; Word Blocks programming
    console, 1–2
    Help Center, 1, 2
    overview, 1
    starting code, 1, 2
mixer, 97
modulo operator, 13
moody functions project, 22
Motion Sensor
    overview, 6, 26
    programming, 28–29
motorized bed, 99
motorized grabber, 95
Motor library, 23
MotorPair library, 24–25
motors
    adjusting for line following, 71–72
    controlling single, 23–24
    controlling two synchronized, 24–25
    converting between floats and integers,
        30–32
    overview, 23
    powering drive wheels, 40
    projects, 33
    stopping before obstacles, 65
    useful mechanisms for, 41–46
move_polygon() function, 56
move_tank() method, 25
movement, programming
    executing turns, 49–54
    line following, 68–75
    overview, 49, 59–60
    projects, 57–58
    stopping before obstacles, 62, 64–68
    tracing shapes, 54–57
    turning to avoid obstacles, 60–62, 63
Movement blocks, Word Blocks, 49
movements, Hub
    detecting, 28
    getting, 29
move() method, 24–25
move polygon block, 56, 57
MSHub library, 2
music, making
    color xylophone, 104
    music box, 104–105
    overview, 103
    theremin, 103
Musical Instrument Digital Interface (MIDI), 3
music box, 104–105
My Blocks
    Bop It–style game, 84–86
    Cookie Clicker–type game, 80
    for executing turns, 50–54
    memory game, 87–88
    tracing shapes, 54–57

N

name tag, smart, 96
negative angle, for left turns, 51, 53
noisy input values, 66
NoneType, in Python, 17
not logical operator, 10, 14
numbers, in Python, 17, 18, 30–32

O

objects, Python, 2
obstacle avoidance
    overview, 59–60
    projects, 77–78
    stopping before obstacles, 62, 64–68
    turning to avoid obstacles, 60–62, 63
on-off (bang-bang) controllers, 64, 69–72

operators
- arithmetic, 13
- comparison, 13–14, 27
- logical, 14
- overview, 13

orientation, Hub
- detecting, 28
- getting, 29

or logical operator, 14
outputs, controlling Hub, 3–6, 7
oven, smart, 99
overshooting turns, testing for, 50–51

P

parallelogram linkage, 44
parameters
- adding to functions, 21
- data types for, 30–32
- methods, 3, 24

pausing
- program with control structures, 9
- using **while** loops for, 10

pawl, 44–45
pentagon, tracing, 55
pet, making robotic, 112–113
pet helper, 97
pick random block, 63
PID controllers. *See* proportional-integral-derivative controllers
pinion, 46
pitch
- beep, customizing, 3
- detecting Hub, 29

pixels on light matrix, controlling, 4, 79–81
plants, useful inventions for, 100–101
play_sound() method, 3, 4
polygons, tracing, 55–56
pooper scooper, 97
power variable, 67, 68
primary color detector project, 33
print() function, 7, 19, 20
printing to console, 7
program, pausing, 9
programming. *See* Python programming; Word Blocks programming
proportional control, line following with, 72–73, 78
proportional gain, 65, 67
proportional-integral-derivative (PID) controllers
- derivative controller, 66–67
- integral controller, 67–68
- overview, 59, 64
- proportional controller, 64–66
- smooth line following with, 73–74
- tuning, 74–75

push-button motor project, 33
Python programming
- basics, 6–7
- controlling Hub outputs and inputs, 3–6
- control structures, 9–12
- console, 1–2, 7, 17–19, 27–28, 30, 94
- converting between floats and integers, 30–32
- data types, 17–18
- first steps, 1–2
- functions, 20–22
- line following, 71, 72–74, 76–78
- lists, 19–20
- motors, 23–25
- obstacle avoidance, 77–78
- operators, 13–14
- overview, xvi, 1
- sensors, 26–30
- starting code, 1, 2
- timers, 13
- turns, executing, 49, 54
- variables, 18–19

Q

quotes (‘’), in Python, 7

R

rack and pinion, 46
rainbow() function, 20–21
randint() function, 57, 83, 86
randomizing direction or turning, in obstacle avoidance, 63
random shape, tracing, 57
range() function, 10, 11, 20
ratchet and pawl, 44–45
reacher, high-shelf, 95
reaction timer game, 81–84
recycling robot, 100
reflected mode, Color Sensor, 26–27, 60, 69–70
regular polygons, tracing, 55–56
repeat loop, 54
reset_yaw_angle() method, 29
response variable, 85, 86
return statement, 21, 22
reversing, during obstacle avoidance, 61
right turns, programming, 50–51, 52
robo cook, 97–98
robo pet, 112–113
Robot Inventor set
- differential assemblies, 40
- gear types in, 38
- motors in, 23
- overview, xv
- starting code, 2
- turntables, 39–40

rocker
- crank-rocker mechanism, 41, 42, 43
- parallelogram linkage, 44

roll, detecting Hub, 29
rotary motion, turning into linear motion, 44, 45
rotation, detecting Hub, 29
round() function, 31, 66
run_for_degrees() method, 23
run_for_rotations() method, 24
run_for_seconds() method, 23, 30
run_to_position() method, 24

S

scissor lift, 45–46
score variable, 79, 80, 85, 86
security projects
- combination lock, 110–111
- treasure box, 109–110
- tripwire, 111

seed disperser, 100–101
sensors, programming
- Color Sensor, 26–27
- converting between floats and integers, 30–32
- Distance Sensor, 27–28
- Force Sensor, 30
- Motion Sensor, 28–29
- overview, 23, 26
- projects, 33
- theremin, 103

sequence list, 87, 89, 90
set_default_speed() method, 25
set_pixel() method, 4
set yaw angle to 0 block, 53
shapes, tracing
- overview, 54
- random, 57
- regular polygons, 55–56
- spiral, 58
- square, 54–55
- star, 57
- triangle, 55
- zigzag, 58

show_image() method, 3, 4, 6

- `sides` input, `move polygon` block, 56
- simple counter project, 22
- single motor, controlling, 23–24
- sleeping face, displaying on light matrix, 7
- sliders, 45
- slow animal project, 47
- smart appliance/furniture, 99
- smart name tag, 96
- smiley face, displaying on light matrix, 2, 3–4
- snake cars, programming, 25, 29
- sounds, controlling Hub, 3
- spacing, in Python, 6
- speed
  - gearing up and gearing down, 36, 37
  - motor, controlling, 23–24, 25
  - when line following, 71–72
  - when stopping before obstacles, 64–65
- SPIKE Prime set
  - Force Sensor, 30
  - motors in, 23
  - overview, xv
  - rack-and-pinion mechanisms, 46
  - starting code, 2
- spinning top project, 47
- spiral, tracing, 58
- spiro gyro, 105–106
- sports projects
  - ball launcher, 107–108
  - ball over the wall, 108–109
  - overview, 107
  - tabletop robot, 109
- spur gears (straight-cut gears), 38
- square, tracing, 54–55
- star, tracing, 57
- `start_at_power()` method, 66
- `start_beep()` method, 3
- `start_sound()` method, 3, 4
- `start_tank_at_power()` method, 73
- `start_tank()` method, 25, 71
- starting code, Python, 1, 2
- `start()` method, 24, 25, 30–31
- `start moving` block, 58
- `start moving with tank steering at power` block, 73
- status light, controlling Hub, 5, 6, 7
- steering
  - controlling, 25
  - rack-and-pinion mechanisms, 46
- step value, `for` loop counter, 11
- `stop()` method, 24, 25
- `stop moving` block, 53
- `stop other stacks` block, 83
- stopping before obstacles
  - overview, 62
  - with PID controller, 64–68
  - wrong way to achieve, 62, 64
- strings, 4, 7, 17
- synchronized motors, controlling, 24–25

T

- tabletop robot, 109
- `tank steering` block, 71
- target angle, for turns, 50
- `target` variable, 73, 75
- there and back again project, 76
- theremin, 103
- threshold, for obstacle avoidance, 61
- `ticks_ms()` function, 83, 86
- time-outs, when avoiding obstacles, 61–62
- timers
  - in Python, 13
  - reaction timer game, 81–84
  - when turning to avoid obstacles, 61–62
- `timer started` variable, 83
- toast tester, 97–98
- torque, 36, 37
- touch sensor, Force Sensor as, 30
- tracing shapes
  - overview, 54
  - random, 57
  - regular polygons, 55–56
  - spiral, 58
  - square, 54–55
  - star, 57
  - triangle, 55
  - zigzag, 58
- traffic light project, 7
- treasure box project, 109–110
- triangle, tracing, 55
- tricky dice project, 92
- triple quotes (""""), in Python, 6
- tripwire, 111
- true/false conditions, 14
- tuning PID controller, 74–75
- `turn_degrees()` function, 54, 55
- turn angle, 55, 56
- `turn degrees` My Block, 53, 54, 76, 77
- turns, executing
  - to avoid obstacles, 60–62, 63
  - overview, 49–50
  - in Python, 54
  - spiro gyro, 106
  - turn degrees, 53
  - turning left, 51
  - turning right, 50–51, 52
  - when line following, 70–71, 76
- turntables, 39–40
- `turn` variable, 73, 74
- two-step (bang-bang) controllers, 62, 64, 69–72
- two synchronized motors, controlling, 24–25
- two-way push-button motor project, 33
- `type()` method, 17–18

U

- updating
  - lists, 19
  - variables, 18–19
- uppercase, in Python, 6
- useful inventions
  - for children, 95–97
  - household, 97–100
  - overview, 95
  - for plants, 100–101
- useless inventions, 101–102

V

- `value` input
  - `draw column` block, 80
  - `for` loop, 10–11, 20
- value(s)
  - assigning to variable, 18
  - returning from function, 21–22
  - updating in variable, 18
- variables
  - creating, 18
  - cycling through lists, 20
  - overview, 18
  - returning values from functions, 21
  - updating, 18–19

W

- `wait_for_distance_closer_than()` method, 27
- `wait_for_distance_farther_than()` method, 27
- `wait_for_new_gesture()` method, 28
- `wait_for_new_orientation()` method, 28
- `wait_for_response()` function, 90
- `wait_for_seconds()` method, 9, 19
- `wait_until_color()` method, 26, 27
- `wait_until_pressed()` method, 5–6, 9, 30
- `wait_until_released()` method, 5–6, 9, 30
- `wait_until()` function, 27
- `wait for response` My Block, 85, 87–88

wait for seconds block, 82
walking mechanisms, 42–43, 44
walk signal project, 7
was_pressed() method, 6
when Left Button pressed Event block, 82, 83
while loops
    C to C project, 15
    cycling through lists, 20
    logical operators, 14
    overview, 9, 10
    stopping with if statement, 12
    turns, executing, 54
    updating variables, 18–19
    with not, 10, 24, 86
while True loop, 10, 12, 14, 81
Word Blocks programming
    art, making, 106–107
    biomimicry, 112–113
    Bop It–style game, 84–86
    Cookie Clicker–type game, 79–81
    dice roll project, 91
    golden cookie project, 93–94
    household inventions, 97–100
    inventions for children, 95–96
    inventions for plants, 100–101
    line following, 69, 71–75, 76–78
    memory game, 87–89
    music projects, 103–105
    obstacle avoidance, 77–78
    overview, 1, 49–50
    reaction timer game, 82–83, 84
    security projects, 110–111
    sports projects, 108–109
    stopping before obstacles, 62, 64–68
    tracing shapes, 54–57
    tricky dice project, 92
    turn degrees, 53
    turning left, 51–52
    turning right, 50–51, 52
    turning to avoid obstacles, 60–62, 63
    useless inventions, 102
write block, 50
write() method, 4

X

xylophone, 104

Y

yaw
    detecting Hub, 29
    using readings to execute turns, 49–54

Z

zigzag, tracing, 58
zigzag strategy, for line following, 69–72