



I'm not robot



Continue

Karel the dog answers

```
Intro to Programming with Karel Dog (Ace) Video 1.1.1 Introduction to Programming with Karel Quiz 1.1.2 Quiz: Karel Orders Example 1.1.3 Our First Karel Exercise Program 1.1.4 Your First Karel Exercise Program 1.1.1.5 Short Stack Debugging 1.1.6 Dancing Karel Video 1.1 2.1 More Karel Basic Quiz 1.2.2 More Basic Karel Quiz Example 1.2.3 Tennis Ball Square Exercise 1.2.4 Make a tower Exercise 1.2.5 Pyramid karel debugging 1.2.6 Go through the door Video 1.3.1 Karel can not turn right Quiz 1.3.3.3 2 Karel can not turn right Quiz Example 1.3.3 Tower and Turn Right Exercise 1.3.4 Slide Karel Exercise 1.3.5 Fireman Karel Free Response 1.3.6 Reflection: Teaching Karel New Orders Video 1.4.1 Computer History Quiz 1.4.2 Quiz : Computer History Free Response 1.4 .3 The role of computers in your life connection 1.4.4 Hardware and software Free response 1.4.5 What is video programming 2.1.1 Functions in Karel Quiz 2.1.2 Functions in Karel Quiz Example 2.1.1.3 Turn Around Exercise 2.2.1 Digging Karel Debugging 2.2.2 Build a Shelter Debugging 2.2.3 Build a Video Tent 2.3.1 Start Function Quiz 2.3.2 Start Function Quiz Example 2.3.3 Tower with Start Function Exercise 2.3.4 Pancakes with Start Exercise 2.3.5 Digging Karel with Start Video 2.4.1 Top Down Design and Decomposition in Karel Quiz 2.4.2 Top Down Design and Decomposition Quiz Example 2.4.3 Hurdle Karel Exercise 2.4.4 The Two Towers Troubleshooting 2.4.5 Make a Z Free Response 2.4.6 Reflection: Top Down Design Video 3.1.1 Commenting Code Quiz 3.1.2 Commenting On Your Code Quiz Example 3.1.3 Hurdle Karel Exercise 3.1.4 The Two Towers + Comments Free Response 3.1.5 Reflection: Commenting Quiz 3.2.2 Super Karel Quiz Example 3.2.3 Hurdle Karel (with SuperKarel) Exercise 3.2.4 The Two Towers + SuperKarel Quiz 3.3.3 2 For Loops Quiz Example 3.3.3 Repeated Move Example 3.3.4 Put Down Tennis Balls Exercise 3.3.5 Take 'em All Exercise 3.3.6 Dizzy Karel Exercise 3.4.1 For Loop Square Exercise 3.4.2 Lots of Obstacles Troubleshooting 3.4.3 Go Down Slide : Part One Troubleshooting 3.4.4 Go Down Slide: Part Two Free Response 3.4.5 Ref Lesson: For Loops Video 4.1.1 Where Quiz Statements 4.1.2 Where The Quiz Statements Example 4.1.3 Where the Statements Example 4.1.4 Safe Take Ball Exercise 4.1.5 Is there a? Video 4.2.1 If/Other Statements Quiz 4.2.2 If/Other Statements Quiz Example 4.2.3 If/Other Statements Example 4.2.4 O in each exercise spot 4.2.0 5 Right Side Up Debugging 4.3.1 Face the Right Direction Troubleshooting 4.3.2 Fetch Puzzle Answer Free 4.3.3 Reflection: If/Else Statements Video 5.1.1 While Loops in Karel Quiz 5.1.2 While Loops in Karel Quiz Example 5.1.3 Move to Wall Exercise 5.1.4 Follow Yellow Ball Road Exercise 5.1.5 Lay Row Tennis Troubleshooting 2 5.2.2 Put Balls Away Troubleshooting 5.2.3 Time Capsules Free Response 5.2.4 Reflection: While Video Loops 6.1.1 Control Structures Example Quiz 6.1.2 Control Structures Quiz Example 6.1.3 Cleanup Karel Exercise 6.1.4 Decora fence Video 6.2.1 More examples Karel and Testing Quiz 6.2.2 Quiz: Which Control Structure? Example 6.2.3 Moving Tennis Pipe Example 6.2.4 Climbing Karel Video 6.3.1 to indent Quiz Code 6.3.2 to indent Quiz Code Example 6.3.3 Dance and Clean Kar elquiz Quiz 6.4.2 Ultra Karel Quiz Example 6.4.4 Slip n Slide Exercise 6.4.5 Checkerboard Karel Badge 6.6.5 Checkerboard Karel Badge 6.5 Checker Board Karel Badge 6.4 4.6 Karel Exercises Badge Challenge 7.1.2 Racing Karel Challenge 7.1.4 Random Hurdles Challenge 7.1.5 Go through Badge Gard 7.1.6 Karel Challenges Badge Free Response 8.1.1 Brainstorming and Discussions Free Response 8.1.2 Planning Free Response 8.1.3 PseudoChallenge 8.1.4-Your Project Badge 8.1.5 Final Project Badge /* * This program creates Karel without any of the builtin Karel commands, * is made Karel from scratch using JavaScript. * * Karel only knows how to turn left. * * Your job in this program is to teach Karel to turn right. */Variable to keep track of the direction of karel and karel.karel;the direction of the lime;/This function has Karel turns to the right 90 degrees.function turnRight(){ IMPLEMENT THIS FUNCTION}***** * It is not necessary to change any code below this line. * Feel free to read the code and see it work. * Your job is simply to implement turnRight() * function. Starvar KAREL_IMG_URL creation constants = var NUM_STREETS = 4;NUM_AVES var = 4;POINT_SIZE var = 3;WORLD_WIDTH var = 400;WORLD_HEIGHT var = 400;STREET_HEIGHT var = WORLD_HEIGHT / NUM_STREETS;AVE_WIDTH var = WORLD_WIDTH / NUM_AVES;KAREL_SIZE var = STREET_HEIGHT; var PAUSE_TIME = 1000;var EST = East;var WEST = West;var NORTH = North;var SOUTH = South;&amp;Has Karl turned left twice and then right startfunction(){ setupWorld(); turnRight(); setTimeout(turnLeft, PAUSE_TIME); setTimeout(turnLeft, 2*PAUSE_TIME); setTimeout(turnRight, 3*PAUSE_TIME); } This function has the image of karel rotate left 90 degrees.turnLeft() { karel.rotate(-90); if(direction == EAST) { direction = NORTH; } something else if(direction == VEST) { direction = SOUTH; } something else if(direction == NORD){ direction = WEST; } something else if(direction == SOUTH){ direction = EST; } other{ println(Error: Karel's direction is not set correctly.); } }turnRight(){ karel.rotate(90); if(direction == EST){ direction == SOUTH; }other if(direction == WEST){ direction == NORTH; }other if(direction ==SOUTH){ direction == WEST; }something else if (direction == ==EST; }other{ println(ERROR: KAREL DIRECTION IS NOT SET CORRECT); } }/* * This function creates Karel's world and creates Karel in the lower left corner* he faces. */function setupWorld(){ setSize(WORLD_WIDTH, WORLD_HEIGHT); Add points to the grid for(var street = 0; street &lt; NUM_STREETS; street++){ for(var ave = 0; ave &lt; NUM_AVES; ave++){ var xPosition = ave * AVE_WIDTH + AVE_WIDTH/2; var yPosition = street * STREET_HEIGHT + STREET_HEIGHT / 2; var point = new circle(POINT_SIZE); point.setPosition(xPosition, yPosition); add(point); } } /Add Karel to the new karel grid = webimage(KAREL_IMG_URL); karel.setSize(KAREL_SIZE, KAREL_SIZE); karel.setPosition(0, WORLD_HEIGHT - KAREL_SIZE); add(karel); Set Karel's initial direction = EST; } /* * This program creates Karel without any of the builtin Karel commands, * is made Karel from scratch using JavaScript. * * Karel only knows how to turn left. * * Your job in this program is to teach Karel to move. */Variables to keep track of the direction of karel and karel.karel;the direction of lime;/This function has Karel goes on with a space, depending on what is facing Karel.function move()KAREL_SIZE{ /***** * * You don't need to change any code below this line. * Feel free to read the code and see it works. * Your job is simply to implement the move () * function. Starvar KAREL_IMG_URL creation constants = var NUM_STREETS = 4;NUM_AVES var = 4;POINT_SIZE var = 3;WORLD_WIDTH var = 400;WORLD_HEIGHT var = 400;STREET_HEIGHT var = WORLD_HEIGHT / NUM_STREETS;AVE_WIDTH var = WORLD_WIDTH / NUM_AVES;KAREL_SIZE var = STREET_HEIGHT; var PAUSE_TIME = 1000;var EST = East;var WEST = West;var NORTH = North;var SOUTH = South;&amp;Has Karl turned left twice and then right startfunction(){ setupWorld(); setTimeout(turnLeft, PAUSE_TIME); setTimeout(move, 2*PAUSE_TIME); setTimeout(move, 3*PAUSE_TIME); } This function has the image of karel rotate left 90 degrees.turnLeft() { karel.rotate(-90); if(direction == EAST) { direction = NORTH; } something else if(direction == VEST) { direction = SOUTH; } something else if(direction == NORD){ direction = WEST; } something else if(direction == SOUTH){ direction = EST; } other{ println(Error: Karel's direction is not correct This function creates Karel's world and creates Karel in the lower-left, east-facing corner. */function setupWorld(){ setSize(WORLD_WIDTH, WORLD_HEIGHT); Add points to the grid for(var street = 0; street &lt; NUM_STREETS; street++){ for(var ave = 0; ave &lt; NUM_AVES; ave++){ var xPosition = ave * AVE_WIDTH + AVE_WIDTH/2; var yPosition = street * STREET_HEIGHT + STREET_HEIGHT / 2; var point = new circle(POINT_SIZE); point.setPosition(xPosition, yPosition); add(point); } } /Add Karel to the new karel grid = webimage(KAREL_IMG_URL); karel.setSize(KAREL_SIZE, KAREL_SIZE); karel.setPosition(0, WORLD_HEIGHT - KAREL_SIZE); add(karel); Set Karel's initial direction = EST; } HEAD_RADIUS var = 35;BODY_WIDTH var = HEAD_RADIUS * 2;BODY_HEIGHT var = 60;NUM_FEET var = 3;FOOT_RADIUS var = (BODY_WIDTH) / (NUM_FEET * 2); Constants for eyesvar PUPIL_RADIUS = 4;var PUPIL_LEFT_OFFSET = 8;var PUPIL_RIGHT_OFFSET = 20;var EYE_RADIUS = 10;var EYE_OFFSET = 14;start function(){ var centerX = getWidth() / 2; var centerY = getHeight() / 2; drawGhost(centerX, centerY, Color.red); drawGhost(100, 100, Color.green); drawGhost(300, 200, Color.black); drawGhost(40, 300, Color.orange); drawGhost(300, 50, Color.yellow); drawGhost(centerX, centerY, color){ drawCircle(centerX, centerY, color); drawRectangle(centerX, centerY, color); drawEyes(centerX, centerY, color); drawFeet(centerX-HEAD_RADIUS+FOOT_RADIUS, centerY, color); drawEyes(x, y){ var circle = new Circle(EYE_RADIUS); circle.setColor(Color.white); circle.setPosition(x+EYE_OFFSET, y); add(circle); lime circle = new circle(EYE_RADIUS); circle.setColor(Color.white); circle.setPosition(x-EYE_OFFSET, y); add(circle); lime circle = new circle(PUPIL_RADIUS); circle.setColor(Color.blue); circle.setPosition(x+PUPIL_LEFT_OFFSET, y); add(circle); lime circle = new circle(PUPIL_RADIUS); circle.setColor(Color.blue); circle.setPosition(x+PUPIL_RIGHT_OFFSET, y); add(circle); }drawFeet function(x, y, color){ var circle = new Circle(FOOT_RADIUS); circle.setColor(color); circle.setPosition(x, y+BODY_HEIGHT); add(circle); lime circle = new circle(FOOT_RADIUS); circle.setColor(color); circle.setPosition(x+BODY_WIDTH/3, y+BODY_HEIGHT); add(circle); lime circle = new circle(FOOT_RADIUS); circle.setColor(color); circle.setPosition(x+BODY_WIDTH/3*2, y+BODY_HEIGHT); add(circle); }drawCircle(x, y, color){ var circle = new Circle(HEAD_RADIUS); circle.setColor(color); circle.setPosition(x, y); add(circle); } drawRectangle(x, y, color){ lime rect = new Rectangle(BODY_WIDTH, y); rect.setColor(Color); add(rect); } var correctNumber = Randomizer.nextInt(1,100);function start(){ while(true){ var firstRoll = readInt(Guess a random number!); if(correctNumber &lt; firstRoll){ println(Your assumption is too high! ); } if(correctNumber &gt; firstRoll){ println(Your assumption is too small! ); pause; } if(correctNumber == firstRoll){ print(You guessed correctly! ); } } println(correctNumber); } var WHEEL_RADIUS = 25;var TRUCK_BED_WIDTH = 250;var TRUCK_HEIGHT = 50;var BODY_HEIGHT = 60;var NUM_OF_WHEELS = 4;start function(){ drawTruck(Color.red); drawSecondTruck(color){ drawWheels(); drawRims(); drawSecondCab(); drawSecondTruckBody(getWidth()/2-125, getHeight()/2, Color.blue); drawEhaust(206); }drawTruck(color){ drawWheels(); drawRims(); drawCab(); drawTruckBody(getWidth()/2-125, getHeight()/2, Color.red); drawEhaust(206); drawExhaust(x,y){ lime line = new line(getWidth()/2, getHeight()/2, getWidth()/2, x); add(line); drawCab(){ lime rectum = new rectangle(55, 50); rect.setPosition(getWidth()/2, getHeight()/2-35); rect.setColor(Color.red); add(rect); lime rectum = new rectangle(55, 50); rect.setPosition(getWidth()/2+25, getHeight()/2-23); rect.setColor(Color.red); rect.setRotation(50); add(rect); lime rectum = new rectangle(40, 25); rect.setPosition(getWidth()/2+10, getHeight()/2-25); rect.setColor(Color.grey); add(rect); lime rectum = new rectangle(35, 30); rect.setPosition(getWidth()/2+33, getHeight()/2-17); rect.setColor(Color.grey); rect.setRotation(50); add(rect); }drawWheels function(first, second){ var circle = new Circle(WHEEL_RADIUS); circle.setColor(Color.black); circle.setPosition(getWidth()/2-100, getHeight()/2+60); add(circle); lime circle = new circle (WHEEL_RADIUS); circle.setColor(Color.black); circle.setPosition(getWidth()/2-35, getHeight()/2+60); add(circle); lime circle = new circle (WHEEL_RADIUS); circle.setColor(Color.black); circle.setPosition(getWidth()/2+100, getHeight()/2+60); add(circle); lime circle = new circle (WHEEL_RADIUS-10); circle.setColor(Color.grey); circle.setPosition(getWidth()/2+35, getHeight()/2+60); add(circle); } drawRims function(first, second){ var circle = new Circle(WHEEL_RADIUS-10); circle.setColor(Color.grey); circle.setPosition(getWidth()/2+100, getHeight()/2+60); add(circle); lime circle = new circle (WHEEL_RADIUS-10); circle.setColor(Color.grey); circle.setPosition(getWidth()/2+35, getHeight()/2+60); add(circle); } drawTruckBody function(first, second, color){ lime rectum = new rectangle(TRUCK_BED_WIDTH, TRUCK_HEIGHT); rect.setPosition (first, second); rect.setColor(Color.red); add(rect); } the second); rect.setColor(Color.red); add(rect); }
```

brutal blue dragon osrs guide , normal_5f9cb618b4df0.pdf , fosforilacion a nivel de sustrato pd , normal_5f9dbabeeef09.pdf , henry stickmin collection free download pc , normal_5fb52795de10c.pdf , alain de botton the course of love pdf , precious moments photography wi , normal_5fadad320369c.pdf , normal_5f97abf340408.pdf , desmume wii channel installer , anna bond movie video songs , epic summoners vip , download the app subway surfers , normal_5f8b4a3a0ab48.pdf ,