

1. The code starts by creating a window screen.
2. The title of the window is "Snake Game".
3. The background color of the window is blue.
4. Next, the code creates two turtle objects: head and food.
5. Head is used to control the snake, while food will be used as the game's object.
6. The code first sets up head so that it looks like a square with white color and starts at (0, 0).
7. Next, it sets up food so that it looks like a square with blue color and places it at (10, 10).
8. Finally, head moves towards food using direction="Stop".
9. The code creates a window screen with the title "Snake Game" and sets the background color to blue.
10. Next, a square head object is created and set to have the color white.
11. The pen up() method is called on the head object, which causes it to rise up from the bottom of the screen.
12. Finally, the goTo() method is used to move the head object towards (0, 0) in the x-direction.
13. Next, two turtle objects are created – one for food and one for head.
14. The food object has a shape of "square" and will be placed at (0, 0) on the screen.
15. The head object has no shape specified and will be placed at (0, 0) in the y-direction.
16. The code starts by creating a few variables to store information about the game.
17. The first is food, which stores information about the shapes and colors of the food that will be displayed on the screen.
18. Next, the code creates a function called speed() that will control how quickly the food moves across the screen.
19. Finally, the shape() and color() functions are used to create different types of food (square, triangle, and circle) with corresponding colors and speeds.
20. The next section of code sets up an event handler for when the user clicks on one of the buttons in the toolbar.
21. This handler calls a function called goto() that takes two arguments: position (in pixels) and direction (which can be "up", "down", "left", or "right").
22. goto() then uses these values to move the pen object to a specific location onscreen and set its properties accordingly.
23. The last section of code displays some text onscreen and starts timing it using time().
24. Then it prints out a score value for each round played as well as high scores for both players at once.
25. The code will create a turtle that will move at 0 speed and have a square shape.
26. The turtle will be white in color and will start at the top left corner of the screen.
27. Once the code has executed, it will output "Score : 0 High Score : 0" in center alignment.
28. The code starts by creating three variables: head, wn, and godown.
29. The head variable stores information about the player's current position on the screen (xcor and ycor), while wn is a listening object that will be used to track keyboard input.
30. Godown stores information about where the player should go next (if they are moving left or right), and goleft and goright store key directions for moving up, down, or left/right respectively.
31. The code then starts to loop through each of these variables in turn.
32. If any of them change (head.direction, wn.listen(), godown.direction), the code updates itself accordingly and sets various properties on head such as its xcor and ycor coordinates, its heading direction (up, down, left/right), as well as colors and shapes for each segment in the game world according to what was just inputted by the user via keyboard presses.
33. Finally, there is a delay timer set to 0.1 seconds which allows time for one frame of animation before starting over again with another round of gameplay; this is important because it ensures that everything looks smooth even when there are lots of objects moving around on-screen at once!
34. The code assigns key directions to the turtle, listens for user input, updates the screen and calculates the player's score.
35. It also creates a new square segment if needed and moves it according to the key presses.
36. If the player is moving their turtle up or down, then head.direction will be set to "up" or "down", respectively.
37. If they are moving left or right, head.direction will be set to "left" or "right".
38. The code also checks whether the player's current position falls within a food zone and if not, then the appropriate x and y coordinates are randomly generated and stored in food and moved accordingly by calling Turtle().goto() with those values.