```
import qrcode

data = input("Enter the data for the QR code: ")

qr = qrcode.QRCode(version=1,
                   error_correction=qrcode.constants.ERROR_CORRECT_L,
                   box_size=20,
                   border=2)

qr.add_data(data)
qr.make(fit=True)

img = qr.make_image(fill_color="black", back_color="white")

filename = "generated_qr_code.png"
img.save(filename)

print(f"QR code generated and saved as {filename}")
```

import qrcode

- **Imports the qrcode module**: This module is used to generate QR codes.

data = input("Enter the data for the QR code: ")

- **Prompts the user to enter data**:
  - data = input("Enter the data for the QR code: "): This line displays a prompt asking the user to enter the data they want to encode into a QR code. The input data provided by the user is stored in the data variable.

qr = qrcode.QRCode(version=1,
error_correction=qrcode.constants.ERROR_CORRECT_L,
box_size=20,
border=2)

- **Creates an instance of the QRCode class with specific settings**:
  - version=1: Specifies the version of the QR code, which controls the size of the QR code. Version 1 is the smallest (21x21 modules).
  - error_correction=qrcode.constants.ERROR_CORRECT_L: Sets the error correction level to 'L' (Low), which can correct up to 7% of errors or damage.
  - box_size=20: Sets the size of each box (module) in the QR code to 20 pixels.
  - border=2: Sets the border (quiet zone) of the QR code to 2 boxes (modules).

qr.add_data(data)

- **Adds the user-provided data to the QR code**:
  - qr.add_data(data): This line adds the data entered by the user to the QR code. This is the information that will be encoded into the QR code.

qr.make(fit=True)

- **Generates the QR code matrix**:
  - qr.make(fit=True): Optimizes the QR code matrix size based on the provided data. The fit=True parameter ensures that the entire data fits within the QR code without extra space.

img = qr.make_image(fill_color="black", back_color="white")

- **Creates an image of the QR code**:
  - img = qr.make_image(fill_color="black", back_color="white"): Converts the QR code matrix into an image. The fill_color parameter sets the color of the QR code modules to black, and the back_color parameter sets the background color to white.

filename = "generated_qr_code.png"

- **Defines the filename for the QR code image**:
  - filename = "generated_qr_code.png": This line sets the filename for the QR code image to "generated_qr_code.png".

img.save(filename)

- **Saves the QR code image to a file**:
  - img.save(filename): This line saves the generated QR code image as a PNG file with the name specified in the filename variable.

print(f"QR code generated and saved as {filename}")

- **Prints a confirmation message**:
  - print(f"QR code generated and saved as {filename}"): This line prints a message to the user confirming that the QR code has been generated and saved as the specified filename.

Summary

1. The script imports the qrcode module.
2. Prompts the user to enter data for the QR code.
3. Creates a QRCode object with specified settings for version, error correction, box size, and border.
4. Adds the user-provided data to the QR code.
5. Optimizes the QR code size to fit the data.
6. Generates an image of the QR code with specified fill and background colors.
7. Defines the filename for the output image.
8. Saves the QR code image to a file.
9. Prints a confirmation message indicating that the QR code has been successfully generated and saved.