Copilot

Certainly! Let's break down the code step by step:

1. **Importing Libraries**:
    - `import qrcode`: This line imports the `qrcode` library, which allows us to create QR codes.
    - `from PIL import Image`: We also import the `Image` module from the Python Imaging Library (PIL) to work with images.
2. **QR Code Configuration**:
    - `qr=qrcode.QRCode(version=1, error_correction=qrcode.constants.ERROR_CORRECT_H, box_size=10, border=10,)`: Here, we create a `QRCode` object named `qr`.
        - `version=1`: Sets the QR code version (size). Higher versions allow more data, but the QR code becomes denser.
        - `error_correction=qrcode.constants.ERROR_CORRECT_H`: Specifies the error correction level (H for high).
        - `box_size=10`: Determines the size of each QR code module (square block).
        - `border=10`: Sets the border width around the QR code.
3. **Adding Data**:
    - `qr.add_data("https://koushikkoush.github.io/Portfolio/")`: We add the URL "https://koushikkoush.github.io/Portfolio/" to the QR code. You can replace this with any other content you'd like to encode.
4. **Generating the QR Code**:
    - `qr.make(fit=True)`: This generates the QR code based on the added data. The `fit=True` parameter ensures that the QR code adjusts its size to fit the data.
5. **Creating the Image**:
    - `img=qr.make_image(fill_color="darkblue",back_color="lightgray")`: We create an image representation of the QR code.
        - `fill_color="darkblue"`: Sets the color of the QR code modules (the actual pattern).
        - `back_color="lightgray"`: Specifies the background color.
6. **Saving the Image**:
    - `img.save("Portfolio.png")`: Finally, we save the QR code image as "Portfolio.png" in the current working directory.

Now you have a QR code that encodes your portfolio URL! Feel free to customize the colors and content as needed. 😊