Generate QR Using Python

```
!pip install pyqrcode
```

```
Collecting pyqrcode
  Downloading PyQRCode-1.2.1.zip (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.9/41.9 kB 1.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pyqrcode
  Building wheel for pyqrcode (setup.py) ... done
  Created wheel for pyqrcode: filename=PyQRCode-1.2.1-py3-none-any.whl size=36221 sha256=fd757c5e5c299212d4c1908e544cded1ad45fdc4502329f
  Stored in directory: /root/.cache/pip/wheels/f5/db/22/ade11e0c0854d1bb2fcc673dc90f6495ee2c0d2b6eb41492e3
Successfully built pyqrcode
Installing collected packages: pyqrcode
Successfully installed pyqrcode-1.2.1
```

```
!pip install pypng
```

```
Collecting pypng
  Downloading pypng-0.20220715.0-py3-none-any.whl.metadata (13 kB)
Downloading pypng-0.20220715.0-py3-none-any.whl (58 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.1/58.1 kB 1.9 MB/s eta 0:00:00
Installing collected packages: pypng
Successfully installed pypng-0.20220715.0
```

```
#library to generate QR code
import pyqrcode
#module for handleling image file
from PIL import Image

#prompt user to give input link
link = input("Enter link: ")
#create QR code for the provided link
qr_code = pyqrcode.create(link)
#save the QR code as a png file
qr_code.png("QRcode.png", scale=5)
#open and display the generated QR code image
Image.open("QRcode.png")
```

```
Enter link: open.spotify.com
```

URL Shortner using python

```
!pip install pyshorteners
```

```
Collecting pyshorteners
  Downloading pyshorteners-1.0.1.tar.gz (10.0 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from pyshorteners) (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->pyshorteners) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->pyshorteners) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->pyshorteners) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->pyshorteners) (2024.12.14)
Building wheels for collected packages: pyshorteners
  Building wheel for pyshorteners (setup.py) ... done
  Created wheel for pyshorteners: filename=pyshorteners-1.0.1-py3-none-any.whl size=17478 sha256=c5cd613fe7a7538767f493fa60124469e696dc6
  Stored in directory: /root/.cache/pip/wheels/40/25/54/000cc118ff192ee36c95b1374ee4c42d5d39143d940de5908a
Successfully built pyshorteners
Installing collected packages: pyshorteners
Successfully installed pyshorteners-1.0.1
```

```
#libray for URL shortening
import pyshorteners

#prompt user to give input
longURL = input("Enter the URL: ")
#create an instance of url shortener
type_tiny = pyshorteners.Shortener()
#shorten the long url using tinyurl service
shortURL = type_tiny.tinyurl.short(longURL)
#print the shortened url
print("New URL is: ", shortURL)
```

```
Enter the URL: colab.research.google.com
New URL is:  https://tinyurl.com/2yzb5nhv
```

Captcha in python

```
!pip install captcha
```

```
Collecting captcha
  Downloading captcha-0.6.0-py3-none-any.whl.metadata (2.1 kB)
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages (from captcha) (11.1.0)
Downloading captcha-0.6.0-py3-none-any.whl (102 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.2/102.2 kB 2.3 MB/s eta 0:00:00
Installing collected packages: captcha
Successfully installed captcha-0.6.0
```

```
# import necessary libraries
from captcha.image import ImageCaptcha #for generating captcha images
from PIL import Image #for handeling image files
import random #for generating random captcha text
import string #for using predefined character sets (letter & digits)

#function to generate random captcha text
def generate_captcha_text(length=4):
  return ''.join(random.choices(string.ascii_letters + string.digits, k=length))

#function to generate captcha image
def generate_captch_image(captcha_text, image_width=300):
  #initialize the captcha image generator
  image = ImageCaptcha(image_width)
  #define the output image file name based on the captcha text
  image_file = f"{captcha_text}.png"
  #create and save captcha image
  image.write(captcha_text, image_file)
  return image_file

#main program execution
#generate rendom captcha text
captcha_text = generate_captcha_text()
#generate and save captcha image
image_file = generate_captch_image(captcha_text)
#output generated captcha image
print(f"Generated CAPTCHA")
Image.open(image_file)
```

```
Generated CAPTCHA
```