

Python – QR Codes et Codes-barres

HTML

CSS

Python

3

M



1. Introduction

Un **QR code** (*Quick Response Code*) est un **code-barres bidimensionnel** capable de stocker des informations sous forme de texte.

Le QR code **ne fait rien tout seul** : il **contient des données**, que le smartphone interprète.

2. Types de données encodables dans un QR code

Exemple de données

- QR code contenant une URL → ouverture d'un site web
- QR code contenant une vCard → ajout d'un contact
- QR code contenant un vEvent → ajout d'un évènement au calendrier

2.1 Texte simple

Contient un message affiché à l'écran après scan.

Exemple :

Bienvenue à la Journée Portes Ouvertes !

Usage :

- message d'information
- consignes
- affichage rapide

2.2 URL (lien web)

Contient l'adresse d'un site web.

Exemple :

<https://www.carnus.fr>

Usage :

- redirection vers une page HTML
- accès à des informations détaillées
- très utilisé en communication

2.3 vCard – Carte de visite numérique

Format standard reconnu par les smartphones.

Exemple :

```
BEGIN:VCARD
VERSION:3.0
FN:Charles Carnus
TITLE:Responsable info
TEL:+33112345678
EMAIL:lycee@carnus.fr
ADR;;;Avenue de Bourran;Rodez;;12000;France
END:VCARD
```

Effet après scan :

- proposition d'ajout automatique au carnet d'adresses

2.4 vEvent – Évènement

Permet d'ajouter un évènement à un agenda.

Exemple :

```
BEGIN:VEVENT
SUMMARY:Journée Portes Ouvertes
DTSTART:20260131T090000
DTEND:20260131T163000
LOCATION:Avenue de Bourran, Rodez
URL:https://www.carnus.fr
END:VEVENT
```

Effet après scan :

- ajout direct dans le calendrier du smartphone

2.5 Autres usages possibles

- Connexion Wi-Fi
- SMS / e-mail prérempli
- Géolocalisation
- Sécurité : un QR code peut masquer un lien malveillant

3. Génération de QR codes en Python

Objectif

- Comprendre **comment Python encode les données**
- Faire le lien avec les formats vus en cours (URL, vCard, vEvent)

1. QR code – URL (exemple)

```
import qrcode

url = "https://www.exemple.fr"

img = qrcode.make(url)
img.save("qr_url_exemple.png")
```

- QR le plus simple
- Sert principalement de **QR dynamique** (redirection)

Ou

```
import qrcode

data = "https://www.exemple.fr"
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
qr.add_data(data)
qr.make(fit=True)
img = qr.make_image(fill='black', back_color='white')
img.show()
```

2. QR code – Texte

```
import qrcode

texte = "Bienvenue à la Journée Portes Ouvertes"

qr = qrcode.make(texte)
qr.save("qr_texte_exemple.png")
```

3. QR code – vCard (carte de visite)

```
import qrcode

vcard = """BEGIN:VCARD
```

```

VERSION:3.0
Mots clés 1 + informations 1
Mots clés 2 + informations 2
Mots clés 3 + informations 3
Mots clés 4 + informations 4
END:VCARD
"""

img = qrcode.make(vcard)
img.save("qr_vcard_exemple.png")

```

Ou

```

import qrcode

vcard_data = """BEGIN:VCARD
VERSION:3.0
FN:X Y
TEL:+33612345678
EMAIL:xy@exemple.fr
URL:https://exemple.fr
END:VCARD"""

qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
qr.add_data(vcard_data)
qr.make(fit=True)
img = qr.make_image(fill_color="black", back_color="white")
img.save("vcard_qr.png")

```

4. QR code – vEvent (évènement)

```

import qrcode

vevent = """BEGIN:VEVENT
Mots clés 1 + informations 1
Mots clés 2 + informations 2
Mots clés 3 + informations 3
Mots clés 4 + informations 4
END:VEVENT
"""

img = qrcode.make(vevent)
img.save("qr_vevent_exemple.png")

```

Ou

```
import qrcode

vevent_data = """BEGIN:VEVENT
SUMMARY:Mots clés 1
DESCRIPTION:Informations 1
DTSTART:20260129T100000Z
DTEND:20260129T110000Z
END:VEVENT"""

qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
qr.add_data(vevent_data)
qr.make(fit=True)
img = qr.make_image(fill_color="black", back_color="white")
img.save("vevent_qr.png")
```

Ces scripts sont des exemples de cours. Le mini-projet demandera :

- une **adaptation**
- une **organisation différente**
- une **réflexion personnelle**

LES CODES-BARRES

1. Introduction

Un **code-barres** est un système de représentation graphique de données **linéaires (1D)**, utilisé principalement pour **identifier** un objet.

2. Rôle et usages des codes-barres

Usages courants

- Identification de produits
- Gestion des stocks
- Logistique
- Billetterie
- Bibliothèques

Ce que contient un code-barres

✓ un **identifiant**

✗ pas de description

✗ pas de prix détaillé

✗ pas de texte long

3. Types de codes-barres (les plus courants)

Code 128

- Alphanumérique
- Très utilisé
- Flexible

Usage :

- référence produit
- numéro interne
- identifiant logistique

EAN-13

- 13 chiffres
- Standard commercial européen

Usage :

- produits de grande distribution

4. Génération de codes-barres en Python

Bibliothèque utilisée

```
pip install python-barcode
```

5. Exemple 1 – Code-barres Code 128

```
import barcode
from barcode.writer import ImageWriter

reference = "PROD-2026-001"

code128 = barcode.get_barcode_class('code128')
barcode_image = code128(reference, writer=ImageWriter())
barcode_image.save("barcode_code128_exemple")
```

- accepte lettres et chiffres

6. Exemple 2 – Code-barres EAN-13

```
import barcode
from barcode.writer import ImageWriter

code_produit = "590123412345"

ean = barcode.get_barcode_class('ean13')
barcode_image = ean(code_produit, writer=ImageWriter())
barcode_image.save("barcode_ean13_exemple")
```

- exactement **12 chiffres** (le 13^e est calculé)
- réservé aux produits commerciaux

QR code = information Code-barres = identification

Dans le mini-projet, vous devrez **choisir le bon type de code** en fonction de l'usage demandé.

Les scripts Python fournis dans les annexes sont des **exemples de cours**. Ils ne doivent pas être copiés tels quels dans le mini-projet.

Ressources Utiles

- [Documentation Python](#)
- [Tutoriel sur les QR codes avec Python](#)