

Générateur de code QR en pure python

Vous avez probablement déjà vu des codes QR. Ils sont partout, aussi bien sur des sites Web que dans des publicités. Ils ressemblent un peu à des codes-barres mais, au lieu de présenter des bandes verticales, ils sont constitués de motifs de forme carrée.

Dans cet article vous allez apprendre à créer votre propre code QR en utilisant python.

1. Brève histoire des codes QR

Le premier système de code QR a été inventé en 1994 par l'entreprise japonaise **Denso Wave**, une filiale de Toyota. L'entreprise avait besoin d'un moyen de suivre avec précision les véhicules et les pièces au cours de la fabrication. Pour y parvenir, elle a cherché à concevoir un type de code-barres capable d'encoder les kanji, les kana et les caractères alphanumériques.

2. Définition

Le code QR (code à réponse rapide), en anglais QR code (Quick Response code) stocke les informations et les rendent accessibles rapidement par un simple scan.

Chaque appareil doté d'une photo, qu'il s'agisse d'un smartphone ou d'une table peut lire et traiter les code QR à l'aide d'une application scanner.

3. Composition d'un code QR

Un QR code se compose de deux éléments :

- 3.1 Trois carrés aux coins du QR code, ils sont destinés au scannage
- 3.2 Le QR code lui-même, qui est intégré comme modèle et contient les informations



Figure 1 Un code QR se compose de trois carrés et d'un modèle contenant le code.

4. Utilité des codes QR :

Ils sont utilisés partout où les données doivent être lues rapidement. Vous pouvez les trouver sur des billets, des distributeurs automatiques, les systèmes de transfert d'argent (par exemple Wave au Sénégal), des timbres, des prospectus, la géolocalisation, des affiches et bien d'autres endroits.

Souvent, **une URL est intégrée dans le code**, à laquelle les personnes intéressées peuvent ensuite accéder via une application de numérisation. Mais il peut aussi contenir des textes tels que des codes de bons ou des informations qui, par exemple, accélèrent le processus de vérification lors d'un contrôle de billets.

5. Création du code QR en python

Installation avec les images

```
Pip install qrcode
```

Une fois les packages installé, nous allons un code QR basique nommé **test.png**

5.1. Exemple simple

La fonction *make* de la bibliothèque *qrcode* permet de générer un code QR , elle reçoit en argument les données.

```
#importer la bibliothèque qrcode
import qrcode

#Generer le code QR
img = qrcode.make('vos données ici')
img.save("test.png")
```

Ce code QR lit "vos données ici".

❖ Versions et taille de l'image :

Il existe 40 versions de code QR. Ces versions contrôlent la taille du code :

La version 1 donne la taille la plus petite, elle créera un QR Code matriciel 21X21.

La version 40 correspond à la plus grande taille.

5.2. Exemple avancé :

Le code ci-dessous est une personnalisation du code QR. Il génère un code QR qui pointe vers mon site web.

```
qr = qrcode.QRCode(  
    version = 1,  
    error_correction=qrcode.constants.ERROR_CORRECT_L,  
    box_size=10,  
    border=4,  
)  
qr.add_data("https://sites.google.com/view/amadoumbaye/accueil")  
qr.make(fit= True)  
  
img = qr.make_image(fill_color="red", back_color="black")  
img.save("moncode.png")
```



Figure 2 Code QR Exemple avancé

L'objet `QRCode` permet de personnaliser le code QR.

Ici nous avons une image 21X21. Le paramètre `version = 1` permet de choisir la version une.

`error_correction` permet de contrôler la correction d'erreur utilisée pour le code QR. Il varie de **7%** (`ERROR_CORRECT_L`) à **30%** (`ERROR_CORRECT_H`).

`box_size` permet de définir le nombre de pixel dans chaque case du code QR.

`border` contrôle l'épaisseur de la bordure. La valeur par défaut de la bordure est 4 pixels.

La fonction `add_data` de l'objet `QRCode` prend en argument les données c'est-à-dire le contenu du code QR.

La fonction `make_image` génère le code QR. Il est aussi utilisé pour définir la couleur de remplissage et la couleur d'arrière-plan du code QR à l'aide des arguments `fill_color` et `back_color`.