

用QRcode生成二维码

生成二维码，最好用的库莫过于QRcode了。安装方式如下：

```
pip install qrcode
```

教程说明 ¶

因为是在jupyter上呈现二维码图片，所以要使用matplotlib，并加上“%matplotlib inline”的声明，直接运行则不需要。

```
from matplotlib import pyplot as plt
%matplotlib inline
```

为了让每一个单元格的代码都可以独立运行，下面都增加了这两句话。

1.生成简单二维码

就一句代码：qrcode.make(data)。data为文本内容。

In [1]:

```
from matplotlib import pyplot as plt
%matplotlib inline
import qrcode
data='helloworld'
img = qrcode.make(data)
plt.imshow(img)
plt.axis('off') #不显示坐标
plt.show()
```



如果要生成网址二维码呢？其实只要把data中的文本换成网址就好了。

In [2]:

```
from matplotlib import pyplot as plt
%matplotlib inline
import qrcode
data = "http://www.vvboard.com.cn"
img = qrcode.make(data)
plt.imshow(img)
plt.axis('off') #不显示坐标
plt.show()
```



生成图片文件更加简单了，“img.save("qr.jpg")”就搞定了。

In [3]:

```
# 保存二维码为文件
img.save("qr.jpg")
```

2.生成不一样的二维码

QRcode支持生成不同样式的二维码，通过参数可以设置二维码的大小、颜色等。下面的代码可以生成绿码（这是健康码中的极品啊）。

In [4]:

```
from matplotlib import pyplot as plt
%matplotlib inline
import qrcode
# 实例化二维码生成类
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
# 设置二维码数据
data = "http://www.vvboard.com.cn"
qr.add_data(data)

# 启用二维码颜色设置
qr.make(fit=True)
img = qr.make_image(fill_color="green", back_color="white")

# 显示二维码
# img.show()
plt.imshow(img)
plt.axis('off') #不显示坐标
plt.show()
```



代码中参数的意义如下：

- `version`参数----二维码的格子矩阵大小，可以是1到40，1最小为2121，40是177177
- `error_correction`参数----二维码错误容许率，默认`ERROR_CORRECT_M`，容许小于15%的错误率
- `box_size`参数----二维码每个小格子包含的像素数量
- `border`参数----二维码到图片边框的小格子数，默认值为4

请调整一下参数，看看生成的二维码有什么变化。

3.生成带LOGO二维码

In [5]:

```
from matplotlib import pyplot as plt
%matplotlib inline
from PIL import Image
import qrcode
# 实例化二维码生成类
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
# 设置二维码数据
data = "http://www.vvboard.com.cn"
qr.add_data(data)

# 启用二维码颜色设置
qr.make(fit=True)
img = qr.make_image()
img = img.convert("RGBA")

icon = Image.open("logo.png")
img_w, img_h = img.size
factor = 4
size_w = int(img_w / factor)
size_h = int(img_h / factor)

icon_w, icon_h = icon.size
if icon_w > size_w:
    icon_w = size_w
if icon_h > size_h:
    icon_h = size_h
icon = icon.resize((icon_w, icon_h), Image.ANTIALIAS)

w = int((img_w - icon_w)/2)
h = int((img_h - icon_h)/2)
icon = icon.convert("RGBA")
img.paste(icon, (w, h), icon)

# 显示二维码
# img.show()
plt.imshow(img)
plt.axis('off') #不显示坐标
plt.show()
```



参考链接：

官方说明：<https://github.com/lincolnloop/python-qrcode#advanced-usage>)
(<https://github.com/lincolnloop/python-qrcode#advanced-usage%EF%BC%89>)

Python3实现生成和识别二维码：<https://www.jianshu.com/p/dda608dbaca5>
(<https://www.jianshu.com/p/dda608dbaca5>)

In []: