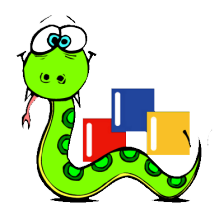
**手把手教你用wx做一个二维码生成器，并打包成可执行程序**

[[](https://blog.csdn.net/xufive/category_9418763.html)](https://blog.csdn.net/xufive/category_9418763.html)

**1. 前言**

我们通常所说的二维码，是指 Quick Response，简写为QR。QR 是近几年来移动设备上超流行的一种编码方式，它比传统的条形码（Bar Code）能存更多的信息，也能表示更多的数据类型。尽管在网上很容易找到各种各样的二维码生成器，但鲜有定制功能。接下来，我将手把手教大家用 wxPython 做一个可以指定前景、背景色以及中心插图的二维码生成器，并最终打包成一个exe程序。本文主要技术要点：

* 使用wxPython做桌面应用程序
* 将图标等资源文件隐藏到代码中
* 用pyinstaller打包成单一的可执行程序

二维码生成器界面如下图所示：  


**2. 二维码生成器代码**

关于如何使用 wxPython 设计桌面程序，我在多篇博客里面均有讲述，这里不再赘述。主程序名为QrCreator.py，里面用到了一个图标文件，名为QrCreator.ico，和主程序在同一级目录下。请自备一个图标文件，并以QrCreator.ico命名。

**QrCreator.py**

# -\*- coding: utf-8 -\*-

import os

import wx

import wx.lib.filebrowsebutton as filebrowse

import wx.lib.colourselect as csel

from PIL import Image

import qrcode

APP\_TITLE = "二维码生成器"

WILDCARD = "PNG(\*.png)|\*.png|JPG(\*.jpg)|\*.jpg"

class MainFrame(wx.Frame):

"""主窗口"""

def \_\_init\_\_(self, parent=None):

wx.Frame.\_\_init\_\_(self, parent, -1, APP\_TITLE, size=(800, 500), style=wx.DEFAULT\_FRAME\_STYLE & ~(wx.RESIZE\_BORDER|wx.MAXIMIZE\_BOX))

icon = wx.Icon('QrCreator.ico')

self.SetIcon(icon) # 设置图标

panel = wx.Panel(self, -1) # 顶级Panel

leftPanel = wx.Panel(panel, -1) # 左侧Panel

rightPanel = wx.Panel(panel, -1) # 右侧Panel

line = wx.StaticLine(panel, -1, style=wx.VERTICAL) # 分隔线

sizer = wx.BoxSizer(wx.HORIZONTAL) # 顶级Sizer

sizer.Add(leftPanel, 1, wx.EXPAND|wx.LEFT|wx.TOP|wx.BOTTOM, border=20) # 左侧Panel加入Sizer

sizer.Add(line, 0, wx.EXPAND|wx.ALL, border=10) # 分隔线加入Sizer

sizer.Add(rightPanel, 1, wx.EXPAND|wx.RIGHT|wx.TOP|wx.BOTTOM, border=20) # 右侧Panel加入Sizer

panel.SetSizer(sizer) # 顶级Panel关联顶级Sizer

panel.Layout()# 顶级Panel布局生效

self.InitLeftPanel(leftPanel) # 初始化左侧Panel

self.InitRightPanel(rightPanel) # 初始化右侧Panel

self.CenterOnScreen() # 窗口居中

def InitLeftPanel(self, panel):

"""初始化左侧窗口"""

# 使用GridBagSizer布局左侧窗口

sizer = wx.GridBagSizer(5, 5) # 参数的意义是Sizer内的控件之间横向和纵向的像素间隔

# 二维码文本输入框

text = wx.StaticText(panel, -1, "文本：")

sizer.Add(text, (0, 0))

self.tc = wx.TextCtrl(panel, -1, "", style=wx.TE\_MULTILINE|wx.TE\_PROCESS\_ENTER)

sizer.Add(self.tc, (1, 0), (1, 2), flag=wx.EXPAND)

# 二维码大小滑块

text = wx.StaticText(panel, -1, "大小：")

sizer.Add(text, (2, 0), flag=wx.TOP, border=5)

self.size = wx.Slider(panel, -1, 1, 1, 5, style=wx.SL\_HORIZONTAL|wx.SL\_AUTOTICKS)

sizer.Add(self.size, (3, 0), flag=wx.EXPAND)

# 二维码容错标准

text = wx.StaticText(panel, -1, "容错：")

sizer.Add(text, (2, 1), flag=wx.TOP, border=5)

choices = [

"低：约修正7%的错误",

"中：约修正15%的错误",

"高：约修正30%的错误"

]

self.tolerance = wx.Choice(panel, -1, choices=choices)

self.tolerance.SetSelection(1) # 默认选中“中”

sizer.Add(self.tolerance, (3, 1), flag=wx.EXPAND)

# 二维码中间logo

text = wx.StaticText(panel, -1, "logo：")

sizer.Add(text, (4, 0), flag=wx.TOP, border=5)

self.logo = filebrowse.FileBrowseButton(panel, -1, labelText="", buttonText="浏览", toolTip="请选择二维码Logo",

dialogTitle="二维码Logo", fileMask=WILDCARD,

changeCallback=self.ShowQrCode)

sizer.Add(self.logo, (5, 0), (1, 2), flag=wx.EXPAND)

# 二维码颜色

staticBox = wx.StaticBox(panel, -1, "颜色：")

sbSizer = wx.StaticBoxSizer(staticBox)

innerSizer = wx.BoxSizer() # 为了实现空白间隔，再套一层BoxSizer

text = wx.StaticText(panel, -1, "前景：")

innerSizer.Add(text, 0)

self.fgColor = csel.ColourSelect(panel, -1, colour=wx.BLACK)

innerSizer.Add(self.fgColor, 1, wx.RIGHT, 5)

text = wx.StaticText(panel, -1, "背景：")

innerSizer.Add(text, 0, flag=wx.LEFT, border=5)

self.bgColor = csel.ColourSelect(panel, -1, colour=wx.WHITE)

innerSizer.Add(self.bgColor, 1)

sbSizer.Add(innerSizer, 1, wx.EXPAND|wx.ALL, 10)

sizer.Add(sbSizer, (6, 0), (1, 2), flag=wx.EXPAND)

# 总成

sizer.AddGrowableCol(0)

sizer.AddGrowableCol(1)

sizer.AddGrowableRow(1)

panel.SetSizer(sizer)

panel.Layout()

# 绑定事件

self.tc.Bind(wx.EVT\_TEXT, self.ShowQrCode)

self.size.Bind(wx.EVT\_SLIDER, self.ShowQrCode)

self.tolerance.Bind(wx.EVT\_CHOICE, self.ShowQrCode)

self.fgColor.Bind(csel.EVT\_COLOURSELECT, self.ShowQrCode)

self.bgColor.Bind(csel.EVT\_COLOURSELECT, self.ShowQrCode)

# 颜色选择控件需要响应一下大小变化事件，才能有较好的显示效果

self.fgColor.Bind(wx.EVT\_SIZE, self.OnFgColorSizeChange)

self.bgColor.Bind(wx.EVT\_SIZE, self.OnBgColorSizeChange)

def OnFgColorSizeChange(self, evt):

"""前景色选择控制大小事件处理"""

self.fgColor.SetColour(wx.BLACK)

def OnBgColorSizeChange(self, evt):

"""背景色选择控制大小事件处理"""

self.bgColor.SetColour(wx.WHITE)

def InitRightPanel(self, panel):

"""初始化右侧窗口"""

sizer = wx.GridBagSizer(10, 10)

# 静态图片

bitWindow = wx.ScrolledWindow(panel, -1, style=wx.BORDER\_SUNKEN)

bitWindow.SetBackgroundColour(wx.WHITE)

bitWindow.SetScrollRate(20, 20)

self.bitmap = wx.StaticBitmap(bitWindow, -1)

self.bitmap.SetBackgroundColour(wx.WHITE)

self.bitmap.CenterOnParent()

sizer.Add(bitWindow, (0, 0), (1, 2), wx.EXPAND)

# 保存到文件

fileBtn = wx.Button(panel, -1, "保存到文件")

sizer.Add(fileBtn, (1, 0), flag=wx.EXPAND)

# 复制到剪切板

clipboardBtn = wx.Button(panel, -1, "复制到剪切板")

sizer.Add(clipboardBtn, (1, 1), flag=wx.EXPAND)

# 总成

sizer.AddGrowableRow(0)

sizer.AddGrowableCol(0)

sizer.AddGrowableCol(1)

panel.SetSizer(sizer)

# 绑定事件

fileBtn.Bind(wx.EVT\_BUTTON, self.OnSave)

clipboardBtn.Bind(wx.EVT\_BUTTON, self.OnClipBoard)

def OnSave(self, evt):

"""保存二维码到文件"""

if len(self.tc.GetValue()) > 0:

dlg = wx.FileDialog(

self, message="保存二维码文件", defaultDir=os.getcwd(),

defaultFile="qrcode.png", wildcard=WILDCARD, style=wx.FD\_SAVE|wx.FD\_OVERWRITE\_PROMPT

)

if dlg.ShowModal() == wx.ID\_OK:

path = dlg.GetPath()

if os.path.splitext(path)[1].lower() == ".png":

imgType = wx.BITMAP\_TYPE\_PNG

else:

imgType = wx.BITMAP\_TYPE\_JPEG

self.bitmap.GetBitmap().SaveFile(path, imgType)

def OnClipBoard(self, evt):

"""复制二维码到剪切板事件处理"""

if len(self.tc.GetValue()) > 0:

if wx.TheClipboard.Open():

wx.TheClipboard.SetData(wx.BitmapDataObject(self.bitmap.GetBitmap()))

wx.TheClipboard.Close()

wx.MessageDialog(self, "成功复制到剪切板。", APP\_TITLE, wx.OK|wx.ICON\_INFORMATION).ShowModal()

else:

wx.MessageDialog(self, "访问剪切板失败。", APP\_TITLE, wx.OK|wx.ICON\_ERROR).ShowModal()

def ShowQrCode(self, evt):

"""显示二维码"""

text = self.tc.GetValue()

if len(text) > 0:

# 初始化qrcode

ec = [qrcode.constants.ERROR\_CORRECT\_L, qrcode.constants.ERROR\_CORRECT\_M, qrcode.constants.ERROR\_CORRECT\_H]

qr = qrcode.QRCode(

version=self.size.GetValue(),

error\_correction=ec[self.tolerance.GetSelection()]

)

# 设置二维码文本

qr.add\_data(text)

# 生成二维码图片

img = qr.make\_image(fill\_color=self.fgColor.GetValue().GetAsString(),

back\_color=self.bgColor.GetValue().GetAsString())

# 设置二维码Logo

logoPath = self.logo.GetValue()

if len(logoPath) > 0 and os.path.isfile(logoPath):

img = img.convert("RGB")

icon = Image.open(logoPath)

# 获取图片的宽高

img\_w, img\_h = img.size

# 参数设置logo的大小

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

# 重新设置logo的尺寸

icon = icon.resize((icon\_w, icon\_h), Image.ANTIALIAS)

# 得到画图的x，y坐标，居中显示

w = int((img\_w - icon\_w) / 2)

h = int((img\_h - icon\_h) / 2)

# 黏贴logo照

img.paste(icon, (w, h), mask=None)

# 显示二维码

self.bitmap.SetBitmap(wx.Bitmap(self.\_PIL2Wx(img)))

self.bitmap.GetParent().SetVirtualSize(self.bitmap.GetSize())

self.bitmap.CenterOnParent()

@staticmethod

def \_PIL2Wx(pilimg):

"""将PIL Image对象转换为wx.Image"""

image = wx.Image(pilimg.size[0], pilimg.size[1])

image.SetData(pilimg.convert("RGB").tobytes())

return image

class MainApp(wx.App):

"""主应用程序"""

def OnInit(self):

"""主应用程序初始化回调函数"""

self.SetAppName(APP\_TITLE)

self.frame = MainFrame(None)

self.frame.Show()

return True

if \_\_name\_\_ == "\_\_main\_\_":

app = MainApp()

app.MainLoop()

**3. 将图标文件隐藏到代码中**

很多同学都遇到过这样的问题：将源码打包成单一文件的可执行程序时，无法处理图标文件。不仅是图标文件，还有图像、音视频等资源文件，也存在同样的问题，除非打包成文件夹。下面，我来一步步演示如何将图标文件隐藏到代码中。

**3.1 将图标文件转为图标模块（.py文件）**

将图标文件转为图标模块（.py文件）并没有现成的工具，下面的代码可以将图标等二进制文件的数据保存为.py文件，并附带了读取数据的方法。详情请参考拙作[《Python小技巧：将图像音视频等资源文件隐藏在代码中》](https://xufive.blog.csdn.net/article/details/104332523)

**pybin.py**

# -\*- coding: utf-8 -\*-

import os

import base64

from io import BytesIO

def bin2module(bin\_file, py\_file=None):

"""二进制文件转存为python模块

bin\_file - 二进制文件名

py\_file - 生成的模块文件名，默认使用二进制文件名，仅更改后缀名

"""

fpath, fname = os.path.split(bin\_file)

fn, ext = os.path.splitext(fname)

if not py\_file:

py\_file = os.path.join(fpath, '%s.py'%fn)

with open(bin\_file, 'rb') as fp:

content = fp.read()

content = base64.b64encode(content)

content = content.decode('utf8')

with open(py\_file, 'w') as fp:

fp.write('# -\*- coding: utf-8 -\*-\n\n')

fp.write('import base64\n')

fp.write('from io import BytesIO\n\n')

fp.write('content = """%s"""\n\n'%content)

fp.write('def get\_fp():\n')

fp.write(' return BytesIO(base64.b64decode(content.encode("utf8")))\n\n')

fp.write('def save(file\_name):\n')

fp.write(' with open(file\_name, "wb") as fp:\n')

fp.write(' fp.write(base64.b64decode(content.encode("utf8")))\n')

if \_\_name\_\_ == '\_\_main\_\_':

# 将图标文件QrCreator.ico转存为icon.py

bin2module('QrCreator.ico', 'icon.py')

https://csdnimg.cn/release/blogv2/dist/pc/img/newCodeMoreWhite.png

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38

将这段代码复制下来，保存在和图标文件QrCreator.ico同级的目录下，运行，就会在同级目录下生成icon.py文件。

**3.2 在程序中导入图标模块**

现在，我们有了一个新的模块，叫做icon，里面保存了图标文件QrCreator.ico的数据，还有两个方法，一个是get\_fp()，用于返回二进制的IO对象（类文件对象），另一个是save()，可以将数据重新保存为本地文件。我们只需要在主程序QrCreator.py的模块导入部分增加下面这一行，就可以不再依赖图标文件QrCreator.ico运行了。

from icon import get\_fp

* 1

**3.3 将icon图标模块的数据转换为wx的图标对象**

借助于pillow模块（主程序QrCreator.py已经导入了该模块，无需重复导入），我们可以很轻松地将icon模块中的数据读成一个PIL图像对象，再转为wx.Bitmap对象，最终转为wx.Icon对象。

im = Image.open(get\_fp()) # 读图标的二进制数据，转为PIL对象

bmp = wx.Bitmap.FromBufferRGBA(im.size[0], im.size[1], im.tobytes()) #转为wx.Bitmap对象

icon = wx.Icon() # 创建空图标

icon.CopyFromBitmap(bmp) # wx.Bitmap数据复制到wx.Icon对象

* 1
* 2
* 3
* 4

用上面的4行代码替换主程序QrCreator.py中的这一行，即算大功告成。

icon = wx.Icon('QrCreator.ico')

* 1

**4. 使用pyinstaller打包成可执行程序**

在主程序QrCreator.py的同级目录下，打开一个命令行窗口，运行：

pyinstaller -F QrCreator.py -i QrCreator.ico -w

-F参数表示打包成单个文件，-W参数表示以窗口模式运行，-i则是指定图标路径。看到这里，你也许会问：图标不是已经隐藏在代码中了吗？为什么还要指定图标文件呢？原来，pyinstaller 的-i参数指定的图标，是用来生成exe文件图标的，而我们隐藏的图标文件，仅在运行程序时显示在标题栏上。如果不使用-i参数，生成的exe文件图标将会使用系统默认图标，而不是我们自己的图标。

运气好的话，将会在主程序QrCreator.py的同级目录下生成两个文件夹：build 和 dist，打包生成的.exe文件就存放在dist目录中。  
  
但是，别高兴得太早。运行该程序的话，也许会出现这样的错误提示：

Failed to execute script pyi\_rth\_pkgres.py

如果真的出现了这个错误，也别灰心，这是pyinstaller的一个bug，请先删除pyinstaller：

[python](https://so.csdn.net/so/search?q=python&spm=1001.2101.3001.7020) -m pip uninstall pyinstaller

然后再安装这个开发版本（要是网速慢，也可以下载到本地安装）：

python -m pip install https://github.com/pyinstaller/pyinstaller/archive/develop.zip

重新打包，就应该不会有问题了。祝你好运！

[[](https://xufive.blog.csdn.net/)**天元浪子**](https://xufive.blog.csdn.net/)

[关注](javascript:;)

* https://csdnimg.cn/release/blogv2/dist/pc/img/newHeart2021Black.png121
* https://csdnimg.cn/release/blogv2/dist/pc/img/newUnHeart2021Black.png
* [[https://csdnimg.cn/release/blogv2/dist/pc/img/newCollectBlack.png](javascript:;)505](javascript:;)
* [打赏](javascript:;)
* [[https://csdnimg.cn/release/blogv2/dist/pc/img/newComment2021Black.png](https://xufive.blog.csdn.net/article/details/104753578#commentBox)17](https://xufive.blog.csdn.net/article/details/104753578#commentBox)
* [https://csdnimg.cn/release/blogv2/dist/pc/img/newShareBlack.png](javascript:;)

专栏目录

* [关于我们](https://www.csdn.net/company/index.html" \l "about" \t "_blank)
* [招贤纳士](https://www.csdn.net/company/index.html#recruit)
* [商务合作](https://marketing.csdn.net/questions/Q2202181741262323995)
* [寻求报道](https://marketing.csdn.net/questions/Q2202181748074189855)
* https://g.csdnimg.cn/common/csdn-footer/images/tel.png400-660-0108
* https://g.csdnimg.cn/common/csdn-footer/images/email.png[kefu@csdn.net](mailto:webmaster@csdn.net)
* https://g.csdnimg.cn/common/csdn-footer/images/cs.png[在线客服](https://csdn.s2.udesk.cn/im_client/?web_plugin_id=29181)
* 工作时间 8:30-22:00
* [公安备案号11010502030143](http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010502030143)
* [京ICP备19004658号](http://beian.miit.gov.cn/publish/query/indexFirst.action)
* [京网文〔2020〕1039-165号](https://csdnimg.cn/release/live_fe/culture_license.png)
* [经营性网站备案信息](https://csdnimg.cn/cdn/content-toolbar/csdn-ICP.png)
* [北京互联网违法和不良信息举报中心](http://www.bjjubao.org/)
* [家长监护](https://download.csdn.net/tutelage/home)
* [网络110报警服务](http://www.cyberpolice.cn/)
* [中国互联网举报中心](http://www.12377.cn/)
* [Chrome商店下载](https://chrome.google.com/webstore/detail/csdn%E5%BC%80%E5%8F%91%E8%80%85%E5%8A%A9%E6%89%8B/kfkdboecolemdjodhmhmcibjocfopejo?hl=zh-CN)
* [账号管理规范](https://blog.csdn.net/blogdevteam/article/details/126135357)
* [版权与免责声明](https://www.csdn.net/company/index.html#statement)
* [版权申诉](https://blog.csdn.net/blogdevteam/article/details/90369522)
* [出版物许可证](https://img-home.csdnimg.cn/images/20220705052819.png)
* [营业执照](https://img-home.csdnimg.cn/images/20210414021142.jpg)
* ©1999-2024北京创新乐知网络技术有限公司

[](https://xufive.blog.csdn.net/)

[天元浪子](https://xufive.blog.csdn.net/)https://csdnimg.cn/release/blogv2/dist/mobile/img/expertNew.png

码龄9年[[https://img-home.csdnimg.cn/images/20210412060958.png](https://i.csdn.net/#/uc/profile?utm_source=14998968) Python领域优质创作者](https://i.csdn.net/#/uc/profile?utm_source=14998968)

[189](https://xufive.blog.csdn.net/)

[原创](https://xufive.blog.csdn.net/)

[5451](https://blog.csdn.net/rank/list/weekly" \t "_blank)

[周排名](https://blog.csdn.net/rank/list/weekly" \t "_blank)

[152万+](https://blog.csdn.net/rank/list/total" \t "_blank)

[总排名](https://blog.csdn.net/rank/list/total" \t "_blank)

349万+

访问

[https://csdnimg.cn/identity/blog8.png](https://blog.csdn.net/blogdevteam/article/details/103478461)

等级

3万+

积分

3万+

粉丝

1万+

获赞

3082

评论

5万+

收藏













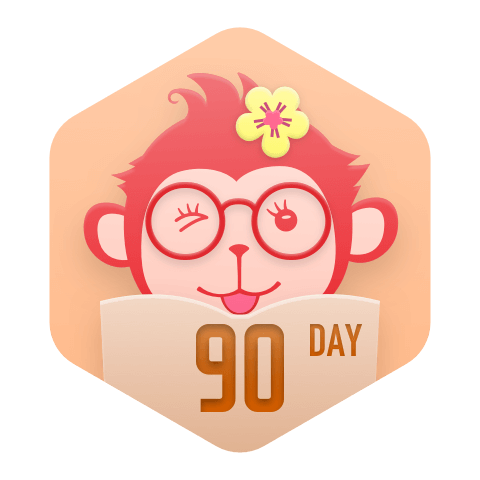






















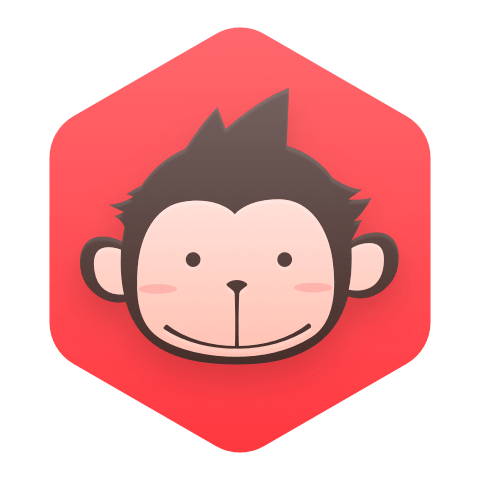












[私信](https://im.csdn.net/chat/xufive)

关注

[](https://mp.csdn.net/)



**热门文章**

* [致 Python 初学者 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://xufive.blog.csdn.net/article/details/102993570) 372524](https://xufive.blog.csdn.net/article/details/102993570)
* [Python十大装B语法 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://xufive.blog.csdn.net/article/details/102856921) 364544](https://xufive.blog.csdn.net/article/details/102856921)
* [wxPython：python首选的GUI库 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://xufive.blog.csdn.net/article/details/82665460) 129192](https://xufive.blog.csdn.net/article/details/82665460)
* [python 程序员进阶之路：从新手到高手的100个模块 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://xufive.blog.csdn.net/article/details/102676755) 122211](https://xufive.blog.csdn.net/article/details/102676755)
* [这么漂亮的图画，竟然是用NumPy画出来的？请跟我来，10行代码玩转NumPy！ [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://xufive.blog.csdn.net/article/details/118018606) 102701](https://xufive.blog.csdn.net/article/details/118018606)

**分类专栏**

[[](https://blog.csdn.net/xufive/category_9506563.html)跟“风云卫星”数据工程师学Python付费](https://blog.csdn.net/xufive/category_9506563.html)27篇

[[](https://blog.csdn.net/xufive/category_12357757.html)Rust学习笔记](https://blog.csdn.net/xufive/category_12357757.html)1篇

[[](https://blog.csdn.net/xufive/category_6727443.html)python论道](https://blog.csdn.net/xufive/category_6727443.html)146篇

[[](https://blog.csdn.net/xufive/category_10884398.html)IT动态](https://blog.csdn.net/xufive/category_10884398.html)1篇

[[](https://blog.csdn.net/xufive/category_9418730.html)MSN三剑客](https://blog.csdn.net/xufive/category_9418730.html)5篇

[[](https://blog.csdn.net/xufive/category_9418763.html)wxPython](https://blog.csdn.net/xufive/category_9418763.html)6篇

[[](https://blog.csdn.net/xufive/category_9418768.html)OpenGL学习笔记](https://blog.csdn.net/xufive/category_9418768.html)15篇

[[](https://blog.csdn.net/xufive/category_9414167.html)OpenCV学习笔记](https://blog.csdn.net/xufive/category_9414167.html)2篇

https://csdnimg.cn/release/blogv2/dist/pc/img/arrowDownWhite.png

**最新评论**

* [不足20行 python 代码，高效实现 k-means 均值聚类算法](https://xufive.blog.csdn.net/article/details/101448969#comments_31339042)

[Char\_Gill:](https://blog.csdn.net/Char_Gill)请问这个跟直接调用sklearn.cluster中的KMeans有什么不同吗？速度上也会有优势吗？

* [C/C++/Java/Go/Rust，Python喊你来打擂：3秒钟内统计出小于1亿的素数个数](https://xufive.blog.csdn.net/article/details/102892049#comments_31192476)

[hq872036709:](https://blog.csdn.net/hq872036709)再改进一下，筛一亿24毫秒

* [C/C++/Java/Go/Rust，Python喊你来打擂：3秒钟内统计出小于1亿的素数个数](https://xufive.blog.csdn.net/article/details/102892049#comments_31148296)

[hq872036709:](https://blog.csdn.net/hq872036709)改进一下我的算法筛一亿25毫秒

* [C/C++/Java/Go/Rust，Python喊你来打擂：3秒钟内统计出小于1亿的素数个数](https://xufive.blog.csdn.net/article/details/102892049#comments_31148228)

[hq872036709:](https://blog.csdn.net/hq872036709)改进一下，现在筛一亿只要25毫秒

* [C/C++/Java/Go/Rust，Python喊你来打擂：3秒钟内统计出小于1亿的素数个数](https://xufive.blog.csdn.net/article/details/102892049#comments_31127619)

[hq872036709:](https://blog.csdn.net/hq872036709)我再改进一下算法，现在筛一亿只要27毫秒

**您愿意向朋友推荐“博客详情页”吗？**

* https://csdnimg.cn/release/blogv2/dist/pc/img/npsFeel1.png

强烈不推荐

* https://csdnimg.cn/release/blogv2/dist/pc/img/npsFeel2.png

不推荐

* https://csdnimg.cn/release/blogv2/dist/pc/img/npsFeel3.png

一般般

* https://csdnimg.cn/release/blogv2/dist/pc/img/npsFeel4.png

推荐

* https://csdnimg.cn/release/blogv2/dist/pc/img/npsFeel5.png

强烈推荐

**最新文章**

* [Slint语言文档（机翻中文版）](https://xufive.blog.csdn.net/article/details/131066003)
* [WxGL应用实例：绘制带有塑料质感的分子模型](https://xufive.blog.csdn.net/article/details/130510891)
* [WxGL应用实例：绘制点云](https://xufive.blog.csdn.net/article/details/130489670)

[2023年7篇](https://xufive.blog.csdn.net/?type=blog&year=2023&month=06)

[2022年26篇](https://xufive.blog.csdn.net/?type=blog&year=2022&month=12)

[2021年38篇](https://xufive.blog.csdn.net/?type=blog&year=2021&month=11)

[2020年44篇](https://xufive.blog.csdn.net/?type=blog&year=2020&month=12)

[2019年55篇](https://xufive.blog.csdn.net/?type=blog&year=2019&month=12)

[2018年15篇](https://xufive.blog.csdn.net/?type=blog&year=2018&month=12)

[2017年5篇](https://xufive.blog.csdn.net/?type=blog&year=2017&month=10)

[2016年1篇](https://xufive.blog.csdn.net/?type=blog&year=2016&month=09)

**目录**

1. [文章目录](https://xufive.blog.csdn.net/article/details/104753578#t0)
2. [1. 前言](https://xufive.blog.csdn.net/article/details/104753578#t1)
3. [2. 二维码生成器代码](https://xufive.blog.csdn.net/article/details/104753578#t2)
4. [3. 将图标文件隐藏到代码中](https://xufive.blog.csdn.net/article/details/104753578#t3)
   1. [3.1 将图标文件转为图标模块（.py文件）](https://xufive.blog.csdn.net/article/details/104753578#t4)
   2. [3.2 在程序中导入图标模块](https://xufive.blog.csdn.net/article/details/104753578#t5)
   3. [3.3 将icon图标模块的数据转换为wx的图标对象](https://xufive.blog.csdn.net/article/details/104753578#t6)
5. [4. 使用pyinstaller打包成可执行程序](https://xufive.blog.csdn.net/article/details/104753578#t7)

[](https://mp.csdn.net/mp_blog/manage/creative)

https://g.csdnimg.cn/side-toolbar/3.4/images/guide.pnghttps://g.csdnimg.cn/side-toolbar/3.4/images/kefu.png举报https://g.csdnimg.cn/side-toolbar/3.4/images/fanhuidingbucopy.png