

捣蛋鬼的忏悔





🎉 首先，衷心祝贺大家成功完成了这一单元的学习，并与P仔一同揭开了码农圈中捣蛋鬼的神秘面纱！🎉

在这个单元中，我们专注于使用tkinter库构建图形用户界面，为大家打开了Python GUI编程的大门。同时，我们还巧妙地融入了time库、MyQR库和threading库等，使Python作品更加丰富多彩、功能强大。



本节课，我们将一同回顾本单元的精彩作品，深入理解每行代码的奥妙和作用。最后，我们将迎接激动人心的【打怪升级】挑战，将所学知识付诸实践，提升编程技能。

Ready? Let's GO! 🚀



1. 温故知新

同盟集结

任务一 窗体出现

任务二 设置单选题

任务三 创建提交按钮和结果标签

任务四 完善结果标签的内容

```
import tkinter as tk          # 导入tkinter库

# 检验答案, 返回结果
def check_answer():
    answer = var.get()
    if answer == "HTML":
        result_label.config(text="恭喜你获得考验资格, 你敢接受挑战吗? ")
    else:
        result_label.config(text="很遗憾, 下次有缘再见")

# 创建主窗口
root = tk.Tk()               # 创建主窗口对象, 起名为root
root.title("同盟集结")      # 配置窗口标题
root.geometry('400x200')    # 设置窗口大小

# 出题
question_label = tk.Label(root, text="以下哪个选项不是编程语言? ")
question_label.pack()
```



1. 温故知新

```
# 设置单选按钮
var = tk.StringVar()
option_1 = tk.Radiobutton(root, text="Python", variable=var,
                           value="Python")
option_2 = tk.Radiobutton(root, text="HTML", variable=var,
                           value="HTML")
option_1.pack()
option_2.pack()

# 创建提交按钮
tijiao_button = tk.Button(root, text="提交答案", command=check_answer)
tijiao_button.pack()

# 创建结果标签
result_label = tk.Label(root, text="")
result_label.pack()

# 运行主循环，等待用户交互
root.mainloop()
```



1. 温故知新

入会考验

任务一 窗体出现

任务二 创建时钟标签

任务三 时间更新

```
# 导入tkinter、time库
import tkinter as tk
import time
# 创建主窗口
root = tk.Tk()
root.title('大聪明的时钟')
root.geometry('400x80+400+400')
root.resizable(width=False,height=False)
# 创建时间标签
label = tk.Label(text='',font=('微软雅黑',60),fg='blue')
label.pack()
# 定义更新时间的函数
def update_clock():
    now = time.strftime('%H:%M:%S')
    label.config(text=now)
    root.after(1000,update_clock) # 每1秒调用一次update_clock函数
# 更新时间
update_clock()
# 运行主循环,等待用户交互
root.mainloop()
```



1. 温故知新

身份二维码

任务一 导入MyQR库

任务二 生成二维码

任务三 完善作品

```
from MyQR import myqr

myqr.run(
    words="Hello, World!", # 跳转到网址或者字符串 (必须是英文字符串)
    version = 5, # 二维码版本
    level = 'H', # 纠错等级, "H"表示最高等级
    picture = '布丁狗.gif', # 指定背景图片路径
    colored = True, # 是否彩色二维码
    contrast = 1.0, # 对比度, 根据需求调整
    brightness = 1.0, # 亮度, 根据需求调整
    save_name = '身份二维码_小米.gif',
)
```



1. 温故知新

谁的恶作剧

任务一 单个弹窗

任务二 弹窗轰炸

任务三 满屏弹窗

```
import tkinter as tk # 导入tkinter库
import random # 导入随机库
import threading # 导入threading库, 用于创建和管理线程
import time # 导入时间库
def tanchuang():
    root = tk.Tk() # 创建窗口对象
    width = root.winfo_screenwidth() # 获取屏幕的宽度
    height = root.winfo_screenheight() # 获取屏幕的高度
    a = random.randint(0, width) # 随机生成宽度
    b = random.randint(0, height) # 随机生成高度
    root.title('恶作剧') # 定义窗口的标题
    root.geometry("400x100"+"+"+str(a)+"+"+str(b)) # 窗口尺寸位置
    tk.Label(root, text='抓不到我吧, 我就是这么强大!', # 显示文字
            bg='yellow', # 背景颜色
            font=('楷体', 17), # 字体和字体大小
            width=40, height=4 # 标签宽高
            ).pack() # 配置标签位置
    root.mainloop()
threads = [] # 空列表, 存储线程对象
for i in range(100): # 需要的弹框数量, 根据自己需要来修改
    # 创建一个线程对象, 将tanchuang函数作为目标函数
    t = threading.Thread(target=tanchuang)
    # 将新创建的线程对象添加到列表中
    threads.append(t)
    # 使主线程暂停0.1秒, 以延迟新线程的启动。确保窗口出现有一定间隔
    time.sleep(0.1)
    # 启动线程中的第i个线程, 调用tanchuang函数并创建一个新弹出窗口
    threads[i].start()
```



2. 打怪升级

小米制作了一个程序，向大家表示歉意，求得大家的谅解。

需要实现的效果：

1. 弹窗如下：



2. 点击“不原谅”按钮，弹窗会出现在屏幕中其他位置

3. 点击“原谅你吧！”按钮，下方会出现“感谢您的谅解！”，1秒后窗口关闭



提示如下：

1. 导入tkinter库

2. 创建窗口对象

3. 定义窗口的标题为“小米的忏悔”

4. 设置窗口的尺寸

5. 配置标签“最近给大家添麻烦了，对不起！你们能够原谅我吗？”

6. 配置按钮，写着“不原谅”，红色底，白色字，点击按钮窗口移动

7. 配置按钮，写着“原谅你吧！”，绿色底，白色字，点击按钮小米感谢谅解

8. 配置“感谢谅解”的标签，初始文本为空

9. 运行主循环，窗口出现

10. 完善move_window函数，实现窗口移到随机位置的效果

11. 完善yes函数，修改感谢谅解的文字为“感谢您的谅解！”，1秒后关闭窗口



2. 打怪升级

【新知识】

可以使用destroy方法来关闭或退出窗口

格式为：

窗口名.destroy()

3. 术语箱

move 移动

window 窗口

destroy 毁坏，消失