

十秒挑战





1. 探索新知

1.1

窗口搭建

1. 导入tkinter库

```
import tkinter as tk
```

2. 创建主窗口

```
root = tk.Tk()
```

3. 设置窗口尺寸、位置

```
root.geometry('300x400')
```

注意：是英文字母x，
不是乘号

4. 设置窗口标题

```
root.title('10秒挑战')
```

5. 设置背景图片

```
bg_image = tk.PhotoImage(file = '十秒挑战.png')  
bg_label = tk.Label(root, image = bg_image)  
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
```

6. 启动事件循环（放在所有代码的最后）

```
root.mainloop()
```



1.1

窗口搭建

任务一新增代码：

```
import tkinter as tk
root = tk.Tk()
root.geometry('300x400')
root.title('10秒挑战')
bg_image = tk.PhotoImage(file='十秒挑战.png')
bg_label = tk.Label(root, image=bg_image)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
...
root.mainloop()
```

1.2

组件布局

还记得怎么创建标签和按钮吗？



1.2

组件布局

1. 设置游戏标题

```
root.title('10秒挑战')

# 创建标签
title_label = tk.Label(root, text="10 秒挑战", font=("楷体", 16))
title_label.pack(pady=(72,10))

root.mainloop()
```

2. 设置时间显示标签

```
# 创建时间显示标签
timex_label=tk.Label(root, text="点击按钮开始挑战\n", font=("楷体", 12))
timex_label.pack(pady=(22, 17))
```

3. 设置游戏按钮

问：如何在按钮中导入图片？

答：跟设置背景图片的方法一致，将图片转换为PhotoImage的形式

```
...
def change():
    Pass
# 创建按钮并设置图片
btn_photo = tk.PhotoImage(file = '按钮.png')
btn = tk.Button(root, image=btn_photo, command=change)
btn.pack(pady=20)

root.mainloop()
```

1.2

组件布局

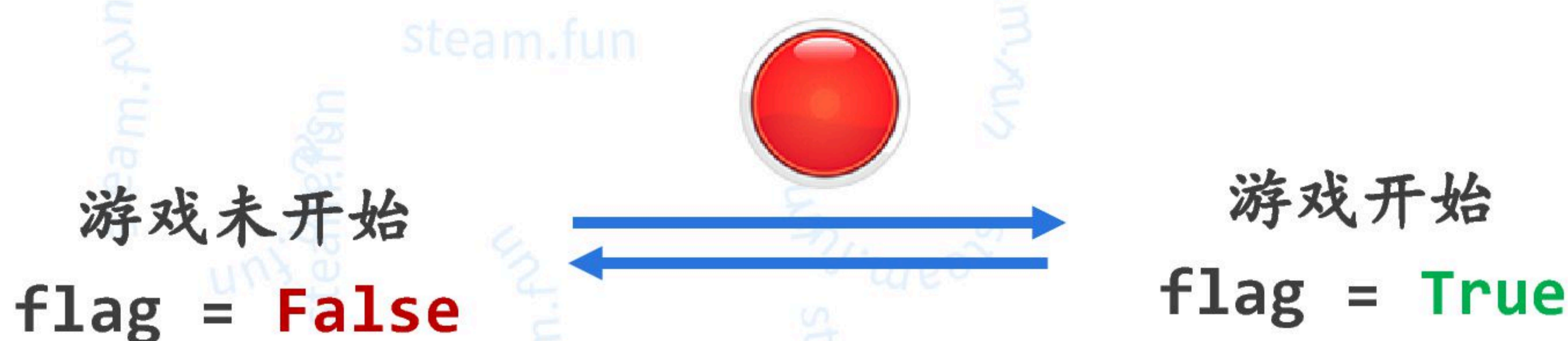
任务二新增代码：

```
...
def change():
    Pass
# 创建标签
title_label = tk.Label(root, text="10 秒挑战", font=("楷体", 16))
title_label.pack(pady=(72,10))
# 创建时间显示标签
timex_label=tk.Label(root,text="点击按钮开始挑战\n",font=("楷体", 12))
timex_label.pack(pady=(22,17))
# 创建按钮并设置图片
btn_photo = tk.PhotoImage(file = '按钮.png')
btn = tk.Button(root, image=btn_photo, command=change)
btn.pack(pady=20)
root.mainloop()
```

1.3

十秒挑战

1. 按钮的开始、停止两个功能转换



```
...
root.title("10秒挑战")
flag = False # 初始化运行状态
...
```

这里可以定义一个函数，
用来切换游戏状态



```
def change():
    global flag # 声明flag为全局变量
    if not flag: # 如果计时尚未开始
        flag = True
        # <计时开始>
    else: # 如果计时已经开始
        flag = False
        # <计时结束，呈现挑战结果>
```

2. 计时开始

游戏考验的是第一次按下按钮起，到第二次按下按钮止，之间的时间差是否为10秒整。

所以，我们需要记录计时开始和计时结束时的时间戳，由此计算出时间差值。

什么是 时间戳？

时间戳是指格林威治时间1970年01月01日00时00分00秒（北京时间1970年01月01日08时00分00秒）起至现在的总秒数

如何获取时间戳？

使用标准库中的 `time` 库。

下方是获取当前时间戳的代码示例：

```
import time
a = time.time() # 获取当前时间的时间戳
print(a)
```

计时开始后，屏幕上会实时更新计时，如何实现这个效果呢？



我们可以定义一个函数 `update_time`，定期更新显示的时间

```
if not flag: # 如果计时尚未开始
    flag = True
    start_time = time.time()
    update_time() # 开始更新时间显示
else: # 如果计时已经开始
    flag = False
# <计时结束，呈现挑战结果>
```

定期更新显示时间

```
def update_time():
    if flag:
        now = time.time()
        timex = now - start_time
        timex_label.config(text=f"时间: {timex:.3f} 秒\n")
        root.after(10, update_time) # 10ms 后更新一次
```

每次更新时间前，都判断是否在计时状态



运行试一下效果吧！

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "D:\py3.8\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "C:/Users/wugedianxia/Desktop/v3_4/源码/十秒挑战 跟做.py", line 28, in change
    update_time() # 开始更新时间显示
  File "C:/Users/wugedianxia/Desktop/v3_4/源码/十秒挑战 跟做.py", line 17, in update_time
    timex = now - start_time
NameError: name 'start_time' is not defined
```

更新时间显示的函数中也会用到 `start_time` 变量，所以需要在获取时间戳的 `change` 函数中声明 `start_time` 为全局变量

1.3

十秒挑战

```
def change():  
    global flag, start_time
```

3. 计时结束，呈现挑战结果

- ① 计时结束时，首先要记录计时结束当下的时间戳
- ② 计算 计时开始 到 计时结束 的实际耗时
- ③ 设置目标耗时（十秒挑战，所以目标耗时为10）
- ④ 计算实际耗时与目标耗时的差值
- ⑤ 更新显示

```
if not flag: # 如果计时尚未开始  
    ...  
else: # 如果计时已经开始  
    flag = False # 停止更新时间显示  
    end_time = time.time() # 获取当前时间戳  
    timex = end_time - start_time # 计算实际耗时  
    goal = 10 # 目标耗时  
    time_difference = timex - goal # 计算与目标差值  
    # <更新显示>
```

保留3位小数

```
# <更新显示>  
timex_label.config(text=f"实际时间: {timex:.3f}秒\n误差: {time_difference:.3f}秒")
```

⑥ 对话框显示结果



思考：各自满足什么条件会弹出右侧的对话框？

1.3

十秒挑战



标签中显示的误差为“0.000”时，弹出左侧对话框，否则弹出右侧对话框

messagebox 是 Tkinter 库中的一个模块，用于在应用程序中显示各种类型的消息对话框，例如信息对话框、警告对话框、错误对话框等。

```
from tkinter import messagebox
```

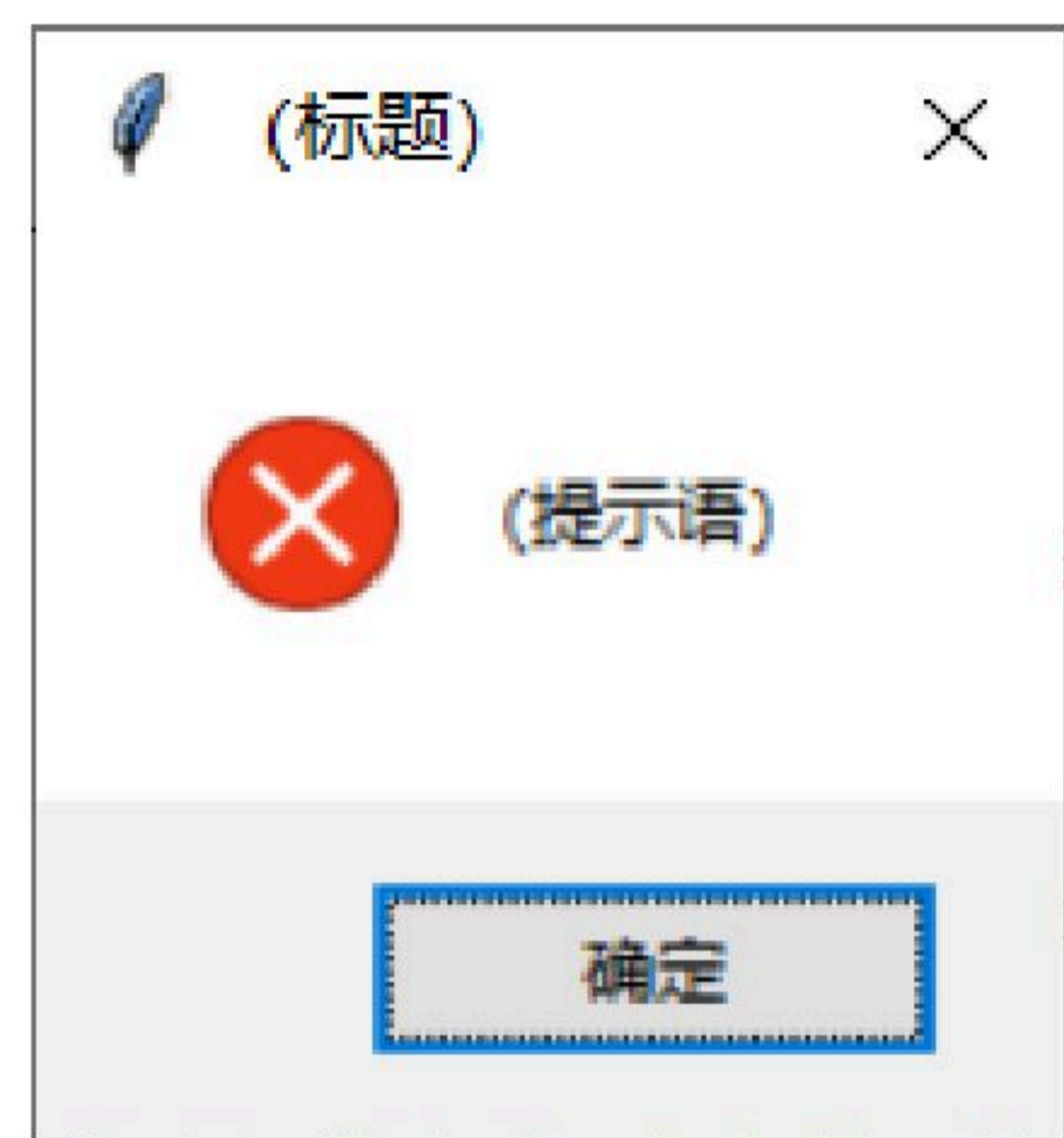
给大家介绍几个messagebox 模块常用的函数，对话框如图所示



信息对话框



警告对话框



错误对话框

messagebox.showinfo(title, message) : 显示信息对话框

messagebox.showwarning(title, message) : 显示警告对话框

messagebox.showerror(title, message) : 显示错误对话框

1.3

十秒挑战

⑥ 对话框显示结果

```
from tkinter import messagebox
...
# 显示结果
if f'{time_difference:.3f}' == '0.000':
    messagebox.showinfo("结果", "难以置信! 你就是掌控时间的神!")
else:
    messagebox.showinfo("结果", "还差一点点, 再接再厉吧!")
```

任务三新增代码:

```
import time
from tkinter import messagebox
...
flag = False # 初始化运行状态
# 定期更新显示时间
def update_time():
    if flag:
        now = time.time()
        timex = now - start_time
        timex_label.config(text=f"时间: {timex:.3f} 秒\n")
        root.after(10, update_time) # 10ms 后更新一次
# 根据当前状态开始或停止计时
def change():
    global flag, start_time
    if not flag: # 如果计时尚未开始
        flag = True
        start_time = time.time() # 获取当前时间戳
        update_time() # 开始更新时间显示
    else: # 如果计时已经开始
        flag = False # 停止更新时间显示
        end_time = time.time() # 获取当前时间戳
        timex = end_time - start_time # 计算实际耗时
        goal = 10 # 目标耗时
        # 计算时间差
        time_difference = timex - goal
        # 更新显示
        timex_label.config(text=f"实际时间: {timex:.3f}秒\n误差:
{time_difference:.3f}秒")
        # 显示结果
        if f'{time_difference:.3f}' == '0.000':
            messagebox.showinfo("结果", "难以置信! 你就是掌控时间的神!")
        else:
            messagebox.showinfo("结果", "还差一点点, 再接再厉吧!")
```

完整代码

```
import tkinter as tk
import time
from tkinter import messagebox

# 创建主窗口
root = tk.Tk()
root.geometry("300x400") # 设置窗口大小
root.title("10秒挑战")

# 初始化运行状态
flag = False

# 定期更新显示时间
def update_time():
    if flag:
        now = time.time()
        timex = now - start_time
        timex_label.config(text=f"时间: {timex:.3f} 秒\n")
        root.after(10, update_time) # 10ms 后更新一次

# 根据当前状态开始或停止计时
def change():
    global flag, start_time

    if not flag: # 如果计时尚未开始
        flag = True
        start_time = time.time() # 获取当前时间戳
        update_time() # 开始更新时间显示
    else: # 如果计时已经开始
        flag = False # 停止更新时间显示
        end_time = time.time() # 获取当前时间戳
        timex = end_time - start_time # 计算实际耗时
        goal = 10 # 目标耗时

        # 计算时间差
        time_difference = timex - goal

        # 更新显示
        timex_label.config(text=f"实际时间: {timex:.3f}秒\n误差:
{time_difference:.3f}秒")
```

完整代码

```
# 显示结果
if f'{time_difference:.3f}' == '0.000':
    messagebox.showinfo("结果", "难以置信! 你就是掌控时间的神!")
else:
    messagebox.showinfo("结果", "还差一点点, 再接再厉吧!")

# 创建标签
title_label = tk.Label(root, text="10 秒挑战", font=("楷体", 16))
title_label.pack(pady=(72,10))
# 创建时间显示标签
timex_label=tk.Label(root, text="点击按钮开始挑战\n", font=("楷体", 12))
timex_label.pack(pady=(22,17))
# 创建按钮并设置图片
btn_photo = tk.PhotoImage(file = '按钮.png')
btn = tk.Button(root, image=btn_photo, command=change)
btn.pack(pady=20)

# 运行Tkinter主循环
root.mainloop()
```



2. 强化练习

1. 作品中，函数 `update_time()` 的主要功能是什么？（ ）

- A. 结束计时
- B. 更新时间显示
- C. 开始计时
- D. 重置窗口大小

2. 全局变量 `flag` 的作用是什么？（ ）

- A. 存储当前时间
- B. 记录按钮点击次数
- C. 标记计时状态
- D. 储存图像文件名

3. 以下哪行代码获取当前时间戳？（ ）

- A. `timex_label.config(text="点击按钮开始挑战")`
- B. `start_time = time.time()`
- C. `btn.pack(pady=20)`
- D. `title_label.pack(pady=20)`



2. 强化练习

4. 在 Tkinter 中哪一个函数用于显示信息对话框? ()

- A. messagebox.showinfo
- B. tkinter.message_box
- C. message.show
- D. messagebox.info

5. 执行下列语句, 得到的结果是? ()

- A. 0.3
- B. 0.33
- C. 0.333
- D. 00.333

```
print('{0:.3f}'.format(0.333333333333))
```

3. 术语箱

messagebox	对话框
change	改变, 更换
update	更新
goal	目标

4. 课后挑战

点击切换背景颜色

要求:

1. 窗口内有标签、按钮如图所示
2. 窗口初始背景为蓝色
3. 蓝色背景时点击按钮, 背景变红, 弹出对话框“变红了”; 红色背景时点击按钮, 背景变蓝, 弹出对话框“变蓝了”

注意: 窗口名.config(bg='颜色')
可以设置、更改窗口背景颜色

