

同盟集结





1. 探索新知

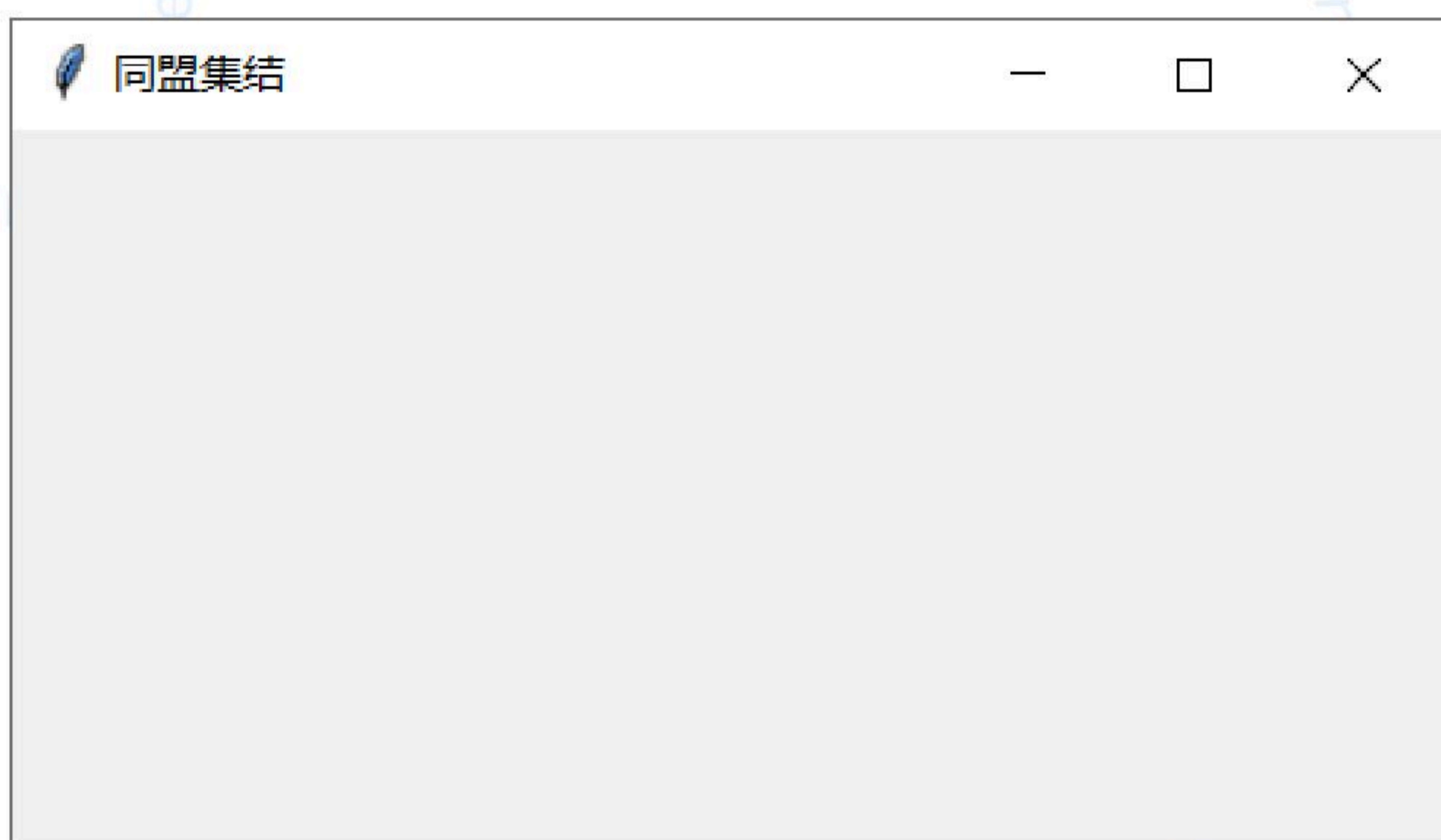
1.1

窗体出现

1. 导入 tkinter 库

```
import tkinter as tk
```

导入tkinter库，并给库起个小名为“tk”，方便后续使用



如何创建窗口？

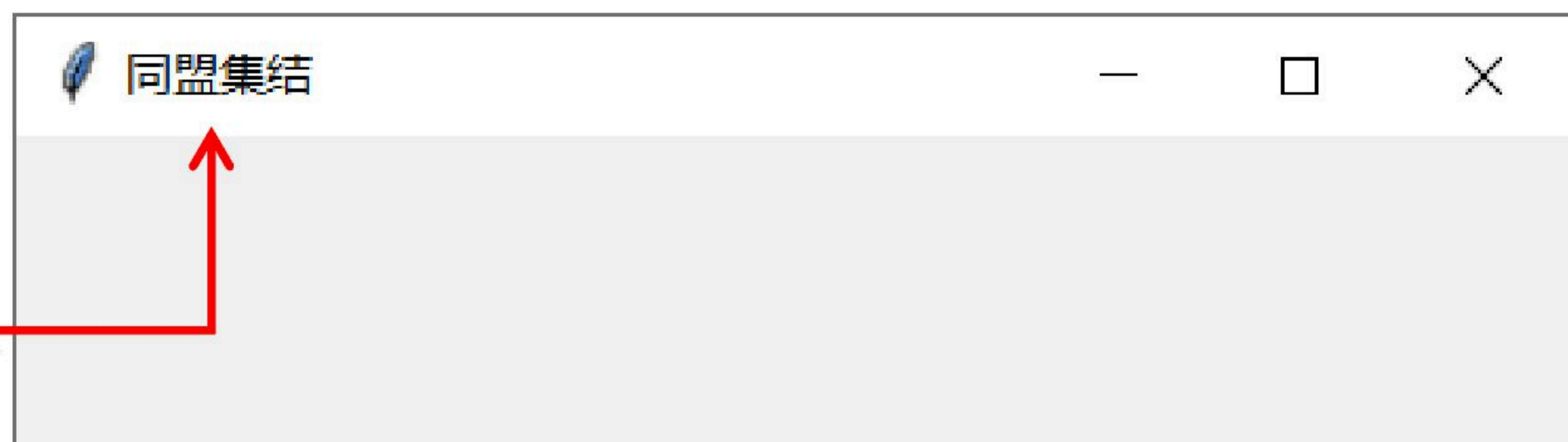
2. 创建主窗口对象，并起名为root

【root】：
根，根源

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

3. 配置窗口标题

```
root.title("同盟集结")
```



1.1

窗体出现

4. 设置窗口大小

【geometry】：
几何

```
root.geometry('400x200')
```

指令手册

这里输入英文小写字母"x"即可

窗口.geometry("widthxheight+x+y")

width: 窗口的宽度
height: 窗口的高度
x: 窗口左上角距离屏幕左边缘的距离
y: 窗口左上角距离屏幕上边缘的距离

注意: "x"、"+" 两边一定不能加空格, 否则会报错

◆ ◆ ◆



运行看看效果, 窗体出现了吗?

5. 让窗口活起来

```
root.mainloop()
```

指令手册

窗口.mainloop()

- mainloop()方法通常在Tkinter应用程序的末尾调用
- 调用mainloop()之后, Tkinter应用程序会进入运行状态, 并等待用户交互

◆ ◆ ◆

1.1

窗体出现

任务一新增代码如下：

```
import tkinter as tk      # 导入tkinter库

# 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
```

1.2

设置单选题



1. 问题标签

【label】：
标签

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

【pack】：
包装、打包

指令手册

tkinter.Label(master, option=value, ...)

- Tkinter 库中的一个部件，用于在窗口中显示文本或图像
- 常用的参数有：

master-确定父窗口部件、**text**-标签上显示的文本、
font-指定文本的字体字号、**fg**-文本颜色、**bg**-标签颜色、
width-标签宽度、**height**-标签高度

下面是一个例子：

```
question = tkinter.Label(root, text="请选择你喜欢的动物",
                          font=('黑体',60), fg='red',
                          bg='black', width=10, height=1)
```

部件.pack()

- 使用 pack 几何管理器将部件添加到窗口中
- 自动调整部件的大小和位置，以便在父窗口中合适地显示它们

2. 单选选项

【StringVar】：
字符串变量

【variable】：
变量

```
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】：
选项

【Radiobutton】：
单选按钮

指令手册

tkinter.StringVar()

- 能够创建一个 StringVar 对象
- StringVar 是 Tkinter 中用于存储字符串值的特殊变量类型
- 与 IntVar(整型变量) 类似, 这种类型的变量通常与单选按钮 (Radiobutton) 或复选框 (Checkbutton) 等部件一起使用, 用于存储用户的选择

tkinter.Radiobutton(master, option=value, ...)

- Tkinter 库中的一个部件, 用于创建单选按钮
- 常用的参数有:

master-确定父窗口部件、**text**-设置选项内容、

variable-与 Radiobutton 部件关联的 Tkinter 变量 (通常是 IntVar 或 StringVar), 用于存储用户选择的值、

value-设置当用户选择该单选按钮时 variable 的值

下面是一个例子:

```
option1 = tkinter.Radiobutton(root, text="小猫",
                               variable=var, value="小猫")
```



一起来分析一下这段代码:

```
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")
```

- 第一行代码创建了一个 StringVar 对象, 并将其赋值给变量 var

- 第二行代码创建了一个单选按钮部件，并将其赋值给变量 `option_1`。
`root`：这里的`root`是一个窗口部件，表示这个单选按钮被添加到了`root`窗口中
`text="Python"`：这个单选按钮显示的是文本"Python"
`variable=var`：与之前创建的 `StringVar` 对象 `var` 相关联
`value="Python"`：当用户选择这个单选按钮时，`var` 的值将被设置为"Python"
- 第三行代码与第二行类似。创建了一个单选按钮部件，并将其赋值给变量 `option_2`。
`root`：这里的`root`是一个窗口部件，表示这个单选按钮被添加到了`root`窗口中
`text="HTML"`：这个单选按钮显示的是文本"HTML"
`variable=var`：与之前创建的 `StringVar` 对象 `var` 相关联
`value="HTML"`：当用户选择这个单选按钮时，`var` 的值将被设置为"HTML"



我发现运行作品后，
两个选项都没有出现，
怎么回事嘞？



我想起来了！忘记使用 `pack` 几何管理器将按钮添加到窗口啦！怎么做呢？猜猜试试吧！

```
option_1.pack()
option_2.pack()
```

1.2

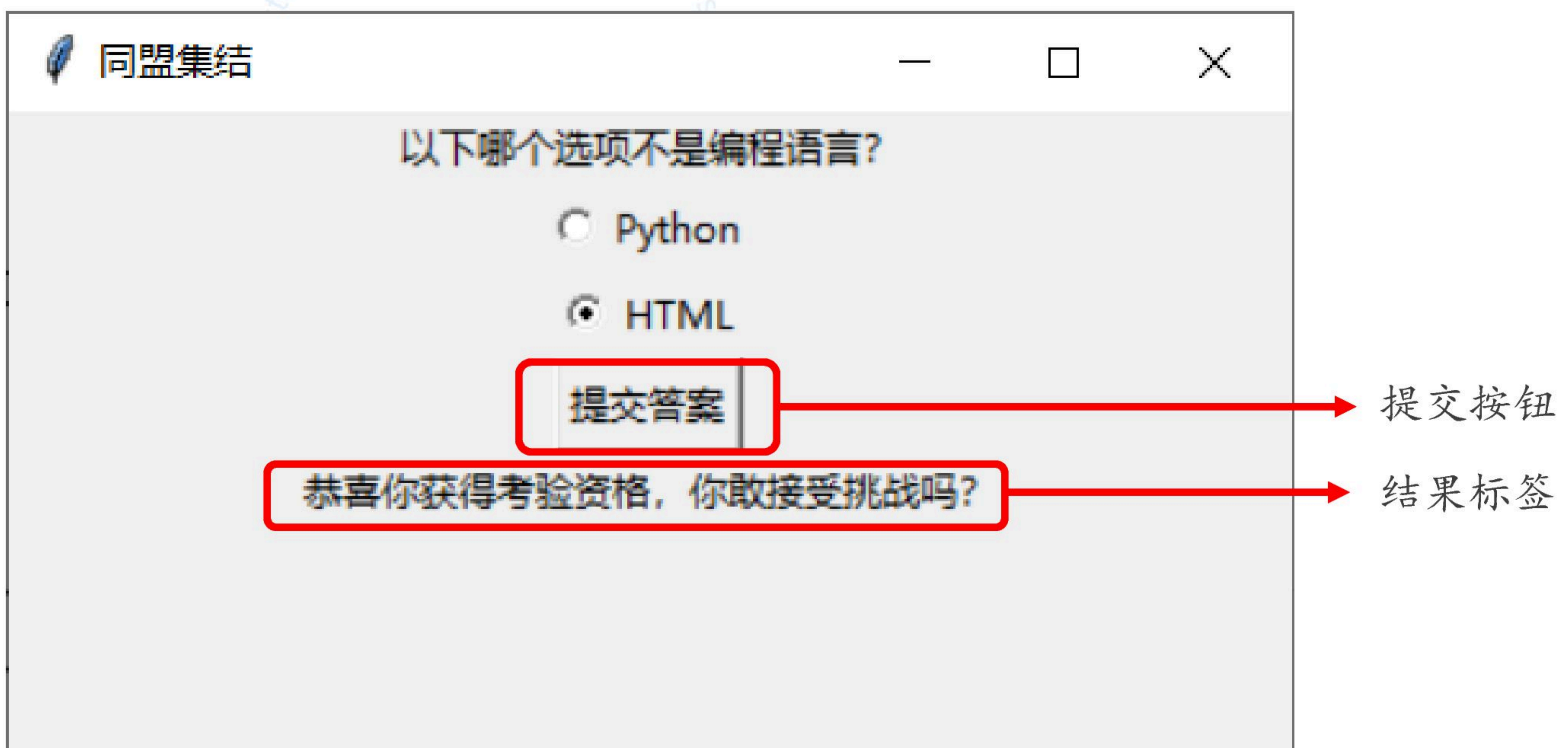
设置单选题

任务二新增代码如下：

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

1.3

创建提交按钮和结果标签



1. 提交按钮

【button】：
按钮

【command】：
命令、指挥

【check】：
检查

```
tijiao_button = tk.Button(root, text="提交答案", command=check_answer)
tijiao_button.pack()
```

指令手册

tkinter.Button(master, option=value, ...)

- Tkinter 库中用于创建按钮部件的函数
- 常用的参数有：

master : 确定父窗口部件、**text** : 按钮上显示的文本、

command : 按钮被点击时执行的函数或方法

下面是一个例子：

```
option1 = tkinter.Button(root, text="提交",
                          command=check_answer)
```

一起来分析一下这段代码：

- 第一行代码创建了一个新的按钮部件，并将其赋值给变量 `tijiao_button`。
root : 这里的 `root` 是一个窗口部件，表示这个提交按钮被添加到了 `root` 窗口中
text="提交答案" : 这个参数设置了按钮上显示的文本，即 "提交答案"
command=check_answer : 这个参数指定了当按钮被点击时应该调用的函数或方法。在这里，当按钮被点击时，`check_answer` 函数将被执行。
需要注意的是，`check_answer` 函数需要在这段代码之前被定义，否则会引发错误
- 第二行代码使用 `pack` 几何管理器将按钮添加到窗口



调用函数前，我们需要先将函数定义好，防止报错

```
def check_answer():
    pass
```

- 先创建一个空函数`check_answer()`，这样程序运行到的时候就不会报错
- 在编写代码时，有时我们可能知道需要某个函数，但还没有决定如何实现它。在这种情况下，可以编写一个空函数作为**占位符**，以便以后回来填写具体的实现代码

2. 结果标签

【result】：

结果

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

- **第一行代码创建了一个新的标签部件**，并将其赋值给变量 `result_label`。
root：这里的 `root` 是一个窗口部件，表示这个标签被添加到了 `root` 窗口中
text=""：这个参数设置了标签中要显示的文本内容，此时这里是一个空字符串
- **第二行代码负责将刚刚创建的标签部件 `result_label` 添加到窗口上**

1.3

创建提交按钮和结果标签

任务三新增代码如下：

```
# 定义空函数占位
def check_answer():
    pass
# 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()
```

1.4

完善结果标签的内容



根据选择答案的正确与否，会给出不同的结果标签，接下来我们完善这部分内容。首先，各位聪明蛋知道题目的正确答案吗？

以下哪个选项不是编程语言？

- A. Python B. HTML

小百科：HTML是一种标记语言，不是一种编程语言。

HTML具有一些语法规则，但它不能执行计算、处理数据或进行逻辑判断等编程操作。因此，HTML不被认为是一种编程语言。

```
def check_answer(): # 定义检查答案的函数
    answer = var.get() # 将选中的答案返回的值放在answer中
    if answer == "HTML": # 如果回答正确, 那么对应修改结果标签文本内容
        result_label.config(text="恭喜你获得考验资格, 你敢接受挑战吗? ")
    else: # 如果回答错误, 那么对应修改结果标签文本内容
        result_label.config(text="很遗憾, 下次有缘再见")
```

单选按钮被点击后返回的值存储在 `var` 中。如何获取 `var` 当前的值?

`var` 是 `StringVar` (字符串变量), 需要通过 `get()` 方法获取 `var` 的当前值。获取到的值放在 `answer` 变量中。

```
answer = var.get()
```

指令手册

部件.config(option=value, ...)

- Tkinter 库中用于配置和修改Tkinter部件的属性, 可以用于各种Tkinter 部件, 如 Button、Label 等等
- 通过调用 部件.config() 方法并传递相应的选项和值, 你可以更改部件的属性, 如文本、颜色、字体、大小等

下面是一个例子:

```
yes = tkinter.Button(root, text="点我", fg="red",
                    bg="blue", font=('微软雅黑', 30))
yes.config(text='别点我', fg="black", bg="yellow",
          font=('黑体', 60))
```



1.4

完善结果标签的内容

```
def check_answer():  
    <检验答案，修改对应结果标签>  
    ...  
  
# 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="") # 结果标签的创建  
result_label.pack()
```

点击提交按钮后，执行 check_answer() 函数

将文本内容为空的结果标签修改为对应内容

任务四修改空函数如下：

```
# 填写check_answer函数具体的实现代码  
def check_answer(): # 定义检查答案的函数  
    answer = var.get() # 将选中的答案返回的值放在answer中  
    if answer == "HTML": # 如果回答正确，那么对应修改结果标签文本内容  
        result_label.config(text="恭喜你获得考验资格，你敢接受挑战吗？")  
    else: # 如果回答错误，那么对应修改结果标签文本内容  
        result_label.config(text="很遗憾，下次有缘再见")
```

完整代码

```
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()
# 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()
```



2. 强化练习

1. 下面哪个函数用于获取用户选择的答案？（ ）

- A. get()
- B. pack()
- C. config()
- D. mainloop()

2. 本节课代码创建的主窗口对象名称是什么？（ ）

- A. tk
- B. root
- C. var
- D. option

3. 在本课代码中，当用户选择正确答案并点击提交按钮时，会显示哪条消息？（ ）

- A. "很遗憾，下次有缘再见"
- B. "恭喜你获得考验资格，你敢接受挑战吗？"
- C. "提交答案"
- D. "以下哪个选项不是编程语言？"

4. 下面哪个函数能用于修改标签部件的文本内容？（ ）

- A. pack()
- B. config()
- C. get()
- D. var()

5. 根据本课代码，下面哪个变量用于存储用户选择的答案？（ ）

- A. root
- B. var
- C. option_1
- D. option_2

3.术语箱

root	根, 根源	variable	变量
label	标签	button	按钮
pack	包装、打包	command	命令、指挥
option	选项	check	检查
StringVar	字符串变量	result	结果
Radiobutton	单选按钮		

4.课后挑战

给题目再添加两个选项，如下：

以下哪个选项不是编程语言？

- A. Python
- B. HTML
- C. Scratch
- D. C++

