

Сборка проекта

В этой миссии мы будем собирать все созданные нами слайды в один проект. Для размещения слайдов мы будем использовать виджет *Notebook* и виджет *Frame*. Оба эти виджета мы возьмем из пакета *tkinter.ttk*. Оба пакета (*tk* и *ttk*) содержат практически одни и те же виджеты (однако *Notebook* в пакете *tk* отсутствует). Считается, что *ttk* более современен и предоставляет чуть больше функций по настройке виджетов.

Виджет *Notebook* представляет набор вкладок. Среди параметров виджета есть ширина и высота, но мы будем делать так:

```
nb = ttk.Notebook()  
nb.pack(expand=True, fill=BOTH)
```

Напомним, что параметр менеджера геометрии *pack expand=True* позволит виджету заполнить все пространство контейнера, а параметр *fill=BOTH* растянет виджет и по горизонтали и по вертикали.

Для добавления вкладки в виджет применяется метод *add()*. Главный его параметр – это ссылка на добавляемый виджет (мы будем использовать виджет *Frame* – своеобразный контейнер, который будет содержать все элементы слайда).

Для ярлыка вкладки вы можете задать только текст:

```
nb.add(добавляемый_виджет, text="Текст")
```

или только изображение:

```
nb.add(добавляемый_виджет, image=f_6)
```

или задать и текст, и изображение на ярлыке – в этом случае высота ярлыка будет рассчитываться по высоте изображения. Напомним, что параметр *compound* задает расположение картинки относительно текста.

```
nb.add(добавляемый_виджет, text="Лого", image=f_1, compound=LEFT)
```



Но перед тем как воспользоваться этим методом, надо создать все добавляемые виджеты – у нас это будут несколько экземпляров виджета *Frame*.

Ранее мы размещали элементы, используя в роли контейнера само главное окно.

Например:

```
tk = Tk()  
c = Canvas(tk, width=500, height=450)  
lbl = Label(tk, image=f)
```

Теперь вместо окна в роли родительского контейнера будет выступать экземпляр виджета *Frame*.

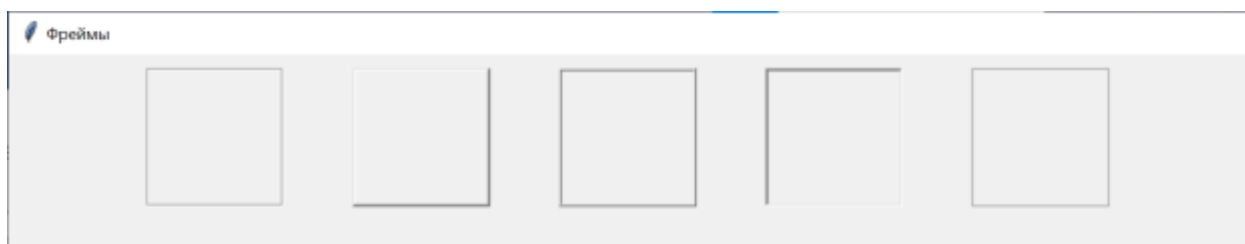
```
fr = ttk.Frame(nb)  
c = Canvas(fr, width=500, height=450)  
lbl = Label(fr, image=f)
```

Фреймы можно использовать не только для вкладок, но и в одном окне для наглядной группировки разных объектов. В этом случае для самого фрейма родительским

контейнером будет являться окно (а не *Notebook* как в примере выше), а для элементов внутри фрейма родительским контейнером будет являться этот фрейм.

Приведем в качестве примера, как выглядят пустые фреймы (квадраты со стороной 100 px) с разными значениями параметра *relief*

```
fr1=ttk.Frame(tk,borderwidth=10,relief='groove',width=100,
height=100)
fr2=ttk.Frame(tk,borderwidth=10,relief='raised',width=100,
height=100)
fr3=ttk.Frame(tk,borderwidth=10,relief='ridge',width=100,
height=100)
fr4=ttk.Frame(tk,borderwidth=10,relief='sunken',width=100,
height=100)
fr5=ttk.Frame(tk,borderwidth=10,relief='solid',width=100,
height=100)
```



Для фрейма, занимающего все пространство вкладки, использовать этот параметр нет особого смысла – разница визуально почти не различима (но в полном примере проекта мы все-таки покажем, как выглядят вкладки с разными значениями этого параметра).

Обратите особое внимание на имена – имена внутри программы должны быть уникальны, поэтому вы не можете создавать метку *lbl* во фрейме *fr1* и метку *lbl* во фрейме *fr2* – имя *lbl* должно быть уникально. В таком случае можно подчеркнуть принадлежность элемента фрейму – например, взять префиксом имя фрейма: *fr1_lbl* – или просто давать уникальные имена по смыслу или используя порядковый номер.

Это же правило относится в том числе и к холсту (*Canvas*) – вкладка, использующая холст, создает свой собственный экземпляр холста.

При создании проекта вы вольны поступить по-разному: можно все изображения, тексты, настройки шрифта и т.п. загрузить в начале программы, а потом при создании фреймов использовать нужные; а можно создавать эти объекты по мере надобности – этот способ позволит не запутать в том, какой объект в каком фрейме используется (естественно, общепотребительные элементы удобнее создавать в начале программы).

У виджета *Notebook* есть еще один интересный метод – *select()*. Он позволяет указать, какая вкладка будет активна. Например, эта команда сделает активной не первую вкладку (по умолчанию именно она активна), а сразу переведет пользователя на шестую вкладку: **`nb.select(fr6)`**

Приведем полный текст примера работы с виджетом – на вкладках все пройденные ранее темы:

```
from tkinter import *
from tkinter import ttk # для создания вкладок и фреймов
from tkinter import font # для создания шрифтового объекта
from PIL import Image, ImageTk # для импорта файлов формата JPG
from tkinter.scrolledtext import ScrolledText # для текста с
```

```

прокруткой
import pygame as pg # для звука

tk = Tk()
tk.title("Проект")
tk.geometry("1024x640+0+0")
tk.resizable(width=False, height=False)

# создаем набор вкладок
nb = ttk.Notebook()
nb.pack(expand=True)

# создаем набор фреймов
fr1 = ttk.Frame(nb, borderwidth=5, relief='groove')
fr2 = ttk.Frame(nb, borderwidth=5, relief='raised')
fr3 = ttk.Frame(nb, borderwidth=5, relief='ridge')
fr4 = ttk.Frame(nb, borderwidth=5, relief='sunken')
fr5 = ttk.Frame(nb, borderwidth=5, relief='solid')
fr6 = ttk.Frame(nb, borderwidth=5, relief='solid')

fr1.pack(fill=BOTH, expand=True)
fr2.pack(fill=BOTH, expand=True)
fr3.pack(fill=BOTH, expand=True)
fr4.pack(fill=BOTH, expand=True)
fr5.pack(fill=BOTH, expand=True)
fr6.pack(fill=BOTH, expand=True)

# добавляем фреймы в качестве вкладок
f_1 = PhotoImage(file="f1.png") # изображение 32x32
f_6 = PhotoImage(file="f6.png")

nb.add(fr1, text="Логотип", image=f_1, compound=LEFT)
nb.add(fr2, text="Фото PNG и GIF")
nb.add(fr3, text="Фото JPG")
nb.add(fr4, text="Текст")
nb.add(fr5, text="Прокрутка")
nb.add(fr6, image=f_6)

# настройки шрифтов
fnt = font.Font(family='Arial', size=12, weight='bold',
slant='roman') # для подписей к фото
fnt_txt = font.Font(family="Times New Roman", size=14) # для
многострочного текста

##### первый фрейм
img_pyт = PhotoImage(file="python_logo.png") # изображение
240x240
fp = font.Font(family="Comic Sans MS", size=72)
lb11 = Label(fr1, image=img_pyт, text="PYTHON", font=fp,
compound="top")

lb11.pack()

```

```

##### второй фрейм
f1 = PhotoImage(file="fire.png") # изображение 350x350
f2 = PhotoImage(file="img1.gif") # изображение 300x500

c1 = Canvas(fr2, width=1024, height=640)
c1.pack()

img = c1.create_image(0, 0, image=f1, anchor="nw")
c1.create_text(0, 370, anchor=NW, text="Метод
create_image\n(файл PNG)", font=fnt)

lbl2 = Label(fr2, image=f2)
lbl2.image = f2
lbl2.place(x=500, y=0)
c1.create_text(500, 520, anchor=NW, text="Виджет Label\n(файл
GIF)", font=fnt)

##### третий фрейм
p = Image.open("img.jpg") # изображение 1000x500
f3 = ImageTk.PhotoImage(p)

c2 = Canvas(fr3, width=1024, height=640)
c2.pack()

lbl3 = Label(fr3, image=f3)
lbl3.image = f3
lbl3.place(x=0, y=60)

c2.create_text(0, 20, anchor=NW, text="Виджет Label - файл JPG",
font=fnt)

# импорт текста для четвертого и пятого фреймов
with open("text1.txt", "r") as f:
    s1 = "".join(f.readlines())
with open("text2.txt", "r") as f:
    s2 = "".join(f.readlines())

##### четвертый фрейм
lbl4_1 = Label(fr4, text=s1, font=fnt_txt, fg='#ffffff',
bg='#ff00ff', justify=LEFT, padx=10, pady=20)
lbl4_2 = Label(fr4, text=s2, font=fnt_txt, fg='#ffffff',
bg='#0000ff', justify=LEFT, wraplength=400)

lbl4_1.place(x=0, y=180, anchor=NW)
lbl4_2.place(x=300, y=10, anchor=NW)

##### пятый фрейм
with open("text1.txt", "r") as f:
    s1 = "".join(f.readlines())
with open("text2.txt", "r") as f:
    s2 = "".join(f.readlines())

txt1 = ScrolledText(fr5, font=fnt_txt, width=90, height=5,

```

```

wrap=WORD, padx=40, pady=15)
txt1.insert(1.0, s2)
txt1.configure(state=DISABLED)

txt2 = ScrolledText(fr5, font=fnt_txt, width=50, height=10,
bg="#CCCCCC", fg="#3300FF", wrap=CHAR, relief=SUNKEN)
txt2.insert(1.0, s2)
txt2.configure(state=DISABLED)

txt3 = ScrolledText(fr5, font=fnt_txt, width=30, height=11,
bg="#CCCCCC", fg="#336633", wrap=CHAR, relief=RAISED)
txt3.insert(1.0, s1)
txt3.configure(state=DISABLED)

txt1.place(x=10, y=10, anchor=NW)
txt2.place(x=10, y=180, anchor=NW)
txt3.place(x=600, y=180, anchor=NW)

##### шестой фрейм
def m_play():
    pg.mixer.music.play()
    btn2.config(state="active")
    btn3.config(state="disabled")
    btn4.config(state="active")

def m_pause():
    pg.mixer.music.pause()
    btn2.config(state="disabled")
    btn3.config(state="active")

def m_unpause():
    pg.mixer.music.unpause()
    btn2.config(state="active")
    btn3.config(state="disabled")

def m_stop():
    pg.mixer.music.stop()
    btn2.config(state="disabled")
    btn3.config(state="disabled")
    btn4.config(state="disabled")

p_mus = Image.open("music.jpg") # изображение размером 600x400
f_mus = ImageTk.PhotoImage(p_mus)
lbl6 = Label(fr6, image=f_mus)
lbl6.image = f_mus
lbl6.place(x=200, y=5)

pg.mixer.init()

```

```
pg.mixer.music.load("mauriat_bach.mp3")

img1 = PhotoImage(file="play.png") # размеры кнопок 40x40
img2 = PhotoImage(file="pause.png")
img3 = PhotoImage(file="unpause.png")
img4 = PhotoImage(file="stop.png")

btn1 = Button(fr6, text="Воспроизвести", width=15,
command=m_play)
btn2 = Button(fr6, text="Пауза", width=15, command=m_pause,
state="disabled")
btn3 = Button(fr6, text="Продолжить", width=15,
command=m_unpause, state="disabled")
btn4 = Button(fr6, text="Стоп", width=15, command=m_stop,
state="disabled")
btn1.place(x=220, y=450)
btn2.place(x=370, y=450)
btn3.place(x=520, y=450)
btn4.place(x=670, y=450)

nb.select(fr1) # делаем активной первую вкладку
tk.mainloop()
pg.mixer.music.stop() # не забываем отключить музыку при выходе
```