Tkinter

Встроенный модуль.

Импорт:

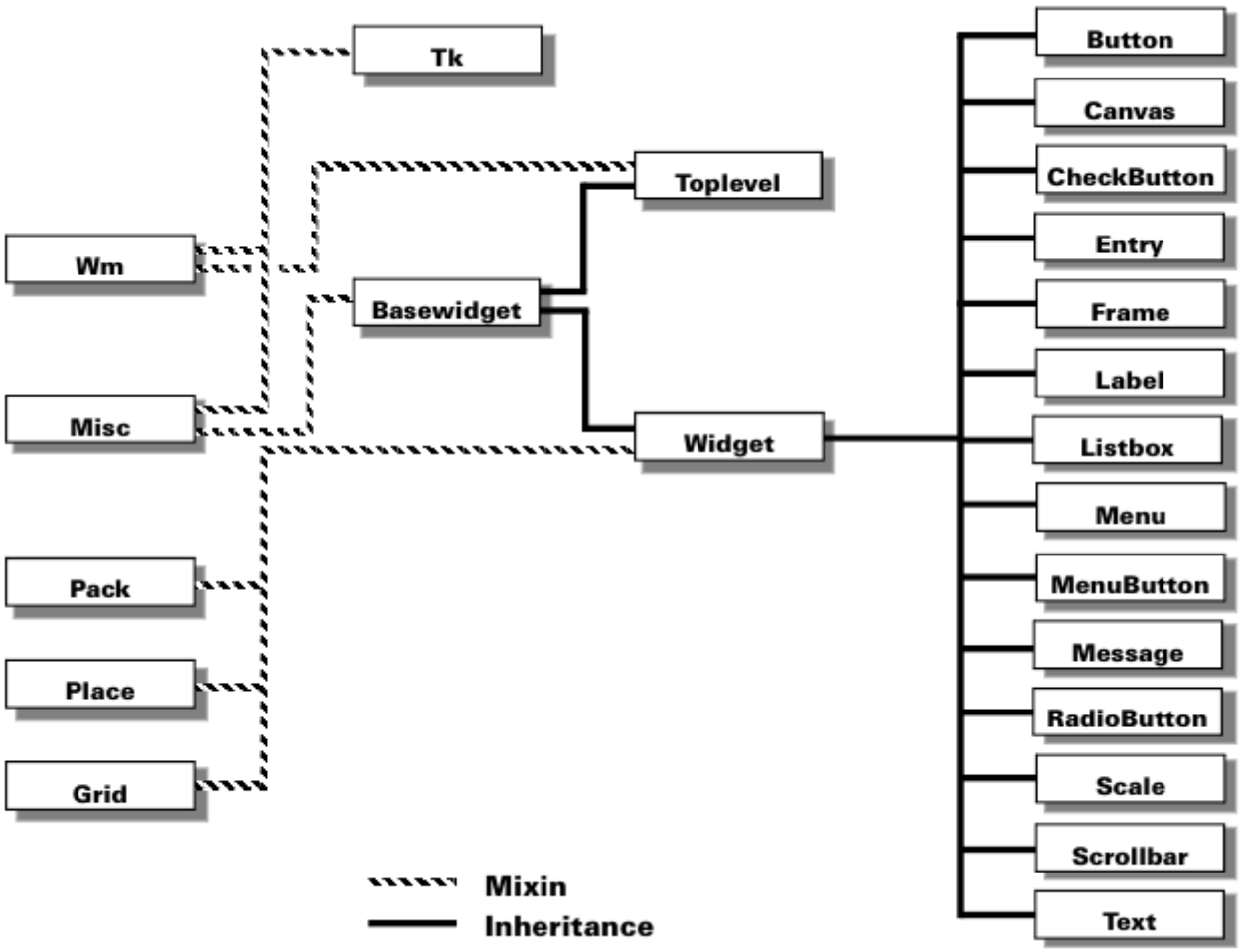
```
import tkinter
from tkinter import *
import tkinter as tk
```

Современный подход сразу использовать ttk

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
# Используйте ttk.Button, ttk.Entry, ttk.Combobox и т.д.
button = ttk.Button(root, text="Современная кнопка")
button.pack()
```

Иерархия виджетов



Общая последовательность действий:

- Создать главное окно.
- Создать виджеты и выполнить конфигурацию их свойств (опций).
- Определить события, то есть то, на что будет реагировать программа.
- Описать обработчики событий, то есть то, как будет реагировать программа.
- Расположить виджеты в главном окне.
- Запустить цикл обработки событий.

Создается главное окно от класса Tk модуля tkinter.

```
root = Tk()
```

Пример окна:

```
from tkinter import *

root = Tk()

ent = Entry(root,width=20)                          #поле ввода
but = Button(root, text="Преобразовать")            #кнопка
```

```
lab = Label(root, width=20, bg='black', fg='white') #метка

def str_to_sort_list(event):
    s = ent.get()
    lab['text'] = s

but.bind('<Button-1>', str_to_sort_list)           #событие левой кнопки мыши
ent.pack()                                         #размещение элементов при помощи менеджера
геометрии
but.pack()
lab.pack()
root.mainloop()                                    #основной цикл
```

Объектно-ориентированный подход

```
from tkinter import *

class Block:
    def __init__(self, master, func):
        self.ent = Entry(master, width=20)
        self.but = Button(master, text="Преобразовать")
        self.lab = Label(master, width=20, bg='black', fg='white')
        self.but['command'] = self.str_to_sort
        #self.but['command'] = getattr(self, func) #вариант с передачей имени функции
        self.ent.pack()
        self.but.pack()
        self.lab.pack()
    def str_to_sort(self):
        s = self.ent.get()
        self.lab['text'] = s

    def str_reverse(self):
        s = self.ent.get()
        self.lab['text'] = s + '1'

root = Tk()
first_block = Block(root)
#first_block = Block(root, 'str_to_sort')
#second_block = Block(root, 'str_reverse')
root.mainloop()
```

События в этом случае определяются по-другому (через свойство).

Есть стилизация через ttk, типа удобнее, узнать.

Если потом работать с элементами не нужно, то без присвоения переменной

```
Label(text="Пункт выдачи").pack()
```

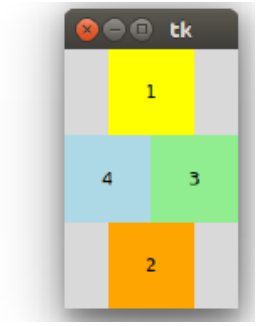
Менеджеры геометрии

Три менеджера геометрии – упаковщик (pack), сетка (grid) и размещение по координатам (place). В одном окне (или родительском виджете) нельзя комбинировать разные менеджеры.

Pack

Объекты применяются последовательно, относительно заданной точки в противоположную сторону. Сторона задается параметром side. Окно создается максимальным размером элементов.

```
l1.pack(side=TOP)
l2.pack(side=BOTTOM)
l3.pack(side=RIGHT)
l4.pack(side=LEFT)
```



По умолчанию side = TOP.

Внутренние (ipadx, ipady) отступы и внешние (padx и pady) отступы.

Изменение размеров окна.

expand (расширение, по умолчанию 0) - равномерное размещение по вертикали.

fill (заполнение) - какое направление экрана заполняем. Может быть NONE, BOTH, X, Y. Без expand не работает.

```
l1.pack(expand=1, fill=Y)
```

anchor (якорь) – может принимать значения N (north – север), S (south – юг), W (west – запад), E (east – восток) и их комбинации.

Вложения блоков

Для вложения используются классы Frame и LabelFrame (с подписью)

```
f_top = LabelFrame(text="Верх")
l1 = Label(f_top, width=7, height=4, bg='yellow', text="1")
```

Т е размещаем Frame, затем размещаем элементы.

Виджеты

Button кнопка

Свойство	Назначение	Пример
text	Надпись на кнопке	b1['text'] = "Изменено"
command	Настройка действия	b1.config(command=change) self.but['command'] = self.str_to_sort (change - настроенная ранее функция)
width и height	Ширина и высота.	По умолчанию ширина и высота текста
bg , fg	Цвет фона и текста	b1['bg'] = '#000000'
activebackground, activeforeground	Цвет фона и текста во время нажатия	
font	Шрифт	b1["font"] = ("Comic Sans MS", 24, "bold")

Label метка. Похожа на кнопку. Нет опции command, связь с событием с помощью bind.

Свойство	Назначение	Пример
bd	Ширина границы вокруг метки	

Entry, text однострочное и многострочное поле ввода

Свойство / метод	Назначение	Пример
get	Получить текст	s = ent.get()
insert(position, text)	Вставить текст. Позиция: 0, END	e1.insert(0, t.strftime('%H:%M:%S '))
delete	Удалить текст	

Свойство / метод	Назначение	Пример
justify	Выравнивание строки CENTER - по центру	

Text:

Свойство / метод	Назначение	Пример
wrap	Правило переноса. WORD - по словам.	text = Text(width=25, height=5, bg="darkgreen", fg='white', wrap=WORD)
	Скролл для текста (и не только для текста) сначала нужно создать	text = Text(width=20, height=7) text.pack(side=LEFT) scroll = Scrollbar(command=text.yview) scroll.pack(side=LEFT, fill=Y) text.config(yscrollcommand=scroll.set)
insert	номер строки и номер столбца Нумерация строк с единицы, а столбцов – с нуля	text.insert(1.0, s)
	Разное форматирование в текстовом поле.	text.tag_add('title', 1.0, '1.end') text.tag_config('title', justify=CENTER, font=("Verdana", 24, 'bold'))

Radiobutton радиокнопки, **Checkbutton** флажки

Listbox списки

Свойство / метод	Назначение	Пример
insert	Добавить элемент Индекс (0-начало, END-конец)	for i in ["1", "2"]: lbox.insert(0,i)

Окно выбора файла

- **askopenfilename**. Открывает диалоговое окно для выбора файла и возвращает путь к выбранному файлу. Если файл не выбран, возвращается пустая строка.
- **askopenfilenames**. Открывает диалоговое окно для выбора файлов и возвращает список путей к выбранным файлам.
- **askdirectory**. Открывает диалоговое окно для выбора каталога (нельзя выбрать несколько каталогов).

- **asksaveasfilename**. Открывает диалоговое окно для сохранения файла, вместо открытия его.

```
from tkinter import filedialog
filename = filedialog.askopenfilename()  # Открываем диалоговое окно для выбора файла
print(filename)
```

PythonMegaWidgets

Сборник виджетов.

Revision #10
Created 17 June 2025 03:35:12 by Admin
Updated 27 January 2026 07:35:02 by Admin