Начнем традиционно — поздороваемся с миром. Для этого сделаем простое окно, у которого будет только заголовок.

Создайте файл my\_app.py и напишите в нем:

from tkinter import \* # Импортируем библиотеку

window = Tk() # Создаем новое окно

window.title("Hello, World!") # Добавляем заголовок окна

window.mainloop() # Запускаем бесконечный цикл окна

Обратите внимание на функцию **mainloop()**. Она вызывает бесконечный цикл. Окно будет ждать любого действия от пользователя до тех пор, пока пользователь его не закроет. Без mainloop() окно не отобразится.

**Базовые виджеты**

Виджеты — это основа библиотеки Tkinter. Через них пользователи взаимодействуют с программой. Каждый виджет определен классом. Вот основные семь:

|  |  |  |
| --- | --- | --- |
| Класс виджета | Что делает | Пример использования |
| Frame | Помогает организовать пользовательский интерфейс как визуально, так и на уровне кода. Отображается как простой прямоугольник. | Разделение интерфейса на блоки. |
| Label | Виджет для отображения текста или изображения. Пользователи могут его просматривать, но не могут взаимодействовать с ним. | Заголовки, подписи, иконки в интерфейсе. |
| Button | Это элемент интерфейса, с которым пользователи могут взаимодействовать. На кнопку нажимают, чтобы выполнить действие. Button может отображать текст и изображение, как и Label, но также может принимать дополнительные параметры для изменения поведения. | Кнопки в интерфейсе: для перехода, сохранения, выхода из программы. |
| Checkbutton | Разновидность кнопки, которая содержит двоичное значение. При нажатии переключатель переворачивается, затем происходит обратный вызов. | Включение и выключение опций с помощью галочек. |
| Radiobutton | Кнопка, которая позволяет выбрать один из нескольких взаимоисключающих вариантов. | Список с опциями, например, для выбора языка интерфейса. |
| Entry | Виджет для ввода одной строки текста. | Указание имени, пароля, города и других данных пользователей. |
| Combobox | Объединяет Entry со списком опций. Пользователи могут выбирать из предложенных вариантов или указывать свои. | Выпадающий список. |

Это лишь базовые виджеты. Посмотреть их подробное описание можно в официальной документации Tk. Там же есть и другие стандартные виджеты — например, скроллбар, шкала прогресса, список.

**Управление геометрией**

Для управления геометрией используются менеджеры. Первый — **pack()**. Он размещает виджеты в определенном порядке. Это происходит в два этапа:

Расчет участка — прямоугольной области, в которую будет помещен виджет. После размещения виджета на участке оставшееся пространство заполняется пустотой. Центрирование виджета в участке, если не указаны иные параметры.

По умолчанию pack() размещает виджеты друг над другом и центрирует.

Самый популярный — **grid()**. Он превращает окно или фрейм в сетку, разделяя пространство на строки и столбцы. Чтобы указать место виджета, достаточно передать индексы row и column. Индексы начинаются с 0.

В grid() также можно гибко управлять размерами виджетов при расширении и меньшении окна.

У pack() тоже есть недостаток. Так как расположение виджетов зависит от порядка вызова метода, приложения сложно модифицировать без полного понимания кода. В grid()эти проблемы решены, поэтому он сейчас используется чаще, чем другие менеджеры геометрии.

Примеры использования виджетов

Чтобы лучше понять концепцию создания и размещения виджетов, давайте разберем несколько примеров. Здесь мы добавляем текст с помощью виджета Label и кнопку с помощью виджета Button:

from tkinter import \*

window = Tk()

window.title("Highload")

lbl = Label(window, text="Первая программа", font=("Roboto Bold", 20))

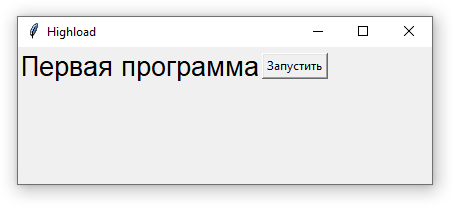
lbl.grid(column=0, row=0)

btn = Button(window, text="Запустить")

btn.grid(column=0, row=2)

window.mainloop()

Получаем вот такой результат:

[](https://highload.today/wp-content/uploads/2021/11/image6-6.png)

Пока наша кнопка ничего не делает, давайте это исправим. Добавим функцию, которая меняет текст Label, и навесим ее на кнопку:

from tkinter import \*

def clicked():

lbl.configure(text="Запустили!")

window = Tk()

window.title("Highload")

lbl = Label(window, text="Первая программа", font=("Roboto Bold", 20))

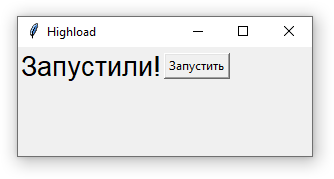
lbl.grid(column=0, row=0)

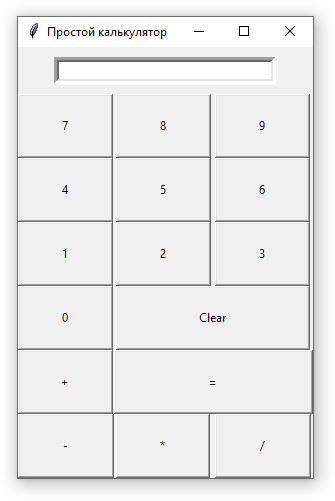
btn = Button(window, text="Запустить", command=clicked)

btn.grid(column=1, row=0)

window.mainloop()

Теперь после нажатия на кнопку текст Label будет меняться на тот, который мы указали внутри функции:

[](https://highload.today/wp-content/uploads/2021/11/image2-9.png)

[](https://highload.today/wp-content/uploads/2021/11/image4-7.png)

Это очень простой калькулятор. Но он уже работает так, как нужно. Его можно усовершенствовать — добавить дополнительные функции, изменить внешний вид, настроить ввод с клавиатуры.

Полный код приложения:

from tkinter import \*

root = Tk()

root.title("Простой калькулятор")

e = Entry(root, width=35, borderwidth=5)

e.grid(row=0, column=0, columnspan=3, padx=10, pady=10)

def button\_click(number):

#e.delete(0, END)

current = e.get()

e.delete(0, END)

e.insert(0, str(current) + str(number))

def button\_clear():

e.delete(0, END)

def button\_add():

first\_number = e.get()

global f\_num

global math

math = "addition"

f\_num = int(first\_number)

e.delete(0, END)

def button\_equal():

second\_number = e.get()

e.delete(0, END)

if math == "addition":

e.insert(0, f\_num + int(second\_number))

if math == "subtraction":

e.insert(0, f\_num - int(second\_number))

if math == "multiplication":

e.insert(0, f\_num \* int(second\_number))

if math == "division":

e.insert(0, f\_num / int(second\_number))

def button\_subtract():

first\_number = e.get()

global f\_num

global math

math = "subtraction"

f\_num = int(first\_number)

e.delete(0, END)

def button\_multiply():

first\_number = e.get()

global f\_num

global math

math = "multiplication"

f\_num = int(first\_number)

e.delete(0, END)

def button\_divide():

first\_number = e.get()

global f\_num

global math

math = "division"

f\_num = int(first\_number)

e.delete(0, END)

# Определяем кнопки

button\_1 = Button(root, text="1", padx=40, pady=20, command=lambda: button\_click(1))

button\_2 = Button(root, text="2", padx=40, pady=20, command=lambda: button\_click(2))

button\_3 = Button(root, text="3", padx=40, pady=20, command=lambda: button\_click(3))

button\_4 = Button(root, text="4", padx=40, pady=20, command=lambda: button\_click(4))

button\_5 = Button(root, text="5", padx=40, pady=20, command=lambda: button\_click(5))

button\_6 = Button(root, text="6", padx=40, pady=20, command=lambda: button\_click(6))

button\_7 = Button(root, text="7", padx=40, pady=20, command=lambda: button\_click(7))

button\_8 = Button(root, text="8", padx=40, pady=20, command=lambda: button\_click(8))

button\_9 = Button(root, text="9", padx=40, pady=20, command=lambda: button\_click(9))

button\_0 = Button(root, text="0", padx=40, pady=20, command=lambda: button\_click(0))

button\_add = Button(root, text="+", padx=39, pady=20, command=button\_add)

button\_equal = Button(root, text="=", padx=91, pady=20, command=button\_equal)

button\_clear = Button(root, text="Clear", padx=79, pady=20, command=button\_clear)

button\_subtract = Button(root, text="-", padx=41, pady=20, command=button\_subtract)

button\_multiply = Button(root, text="\*", padx=40, pady=20, command=button\_multiply)

button\_divide = Button(root, text="/", padx=41, pady=20, command=button\_divide)

# Располагаем кнопки на экране

button\_1.grid(row=3, column=0)

button\_2.grid(row=3, column=1)

button\_3.grid(row=3, column=2)

button\_4.grid(row=2, column=0)

button\_5.grid(row=2, column=1)

button\_6.grid(row=2, column=2)

button\_7.grid(row=1, column=0)

button\_8.grid(row=1, column=1)

button\_9.grid(row=1, column=2)

button\_0.grid(row=4, column=0)

button\_clear.grid(row=4, column=1, columnspan=2)

button\_add.grid(row=5, column=0)

button\_equal.grid(row=5, column=1, columnspan=2)

button\_subtract.grid(row=6, column=0)

button\_multiply.grid(row=6, column=1)

button\_divide.grid(row=6, column=2)

root.mainloop()

Заключение

Чтобы потренироваться с Tkinter, создайте еще несколько простых приложений. В качестве руководства можно использовать это видео от freeCodeCamp. Это мини-курс по Tkinter, в рамках которого вы создадите программу для просмотра картинок. Это будет минималистическое приложение для погоды.