

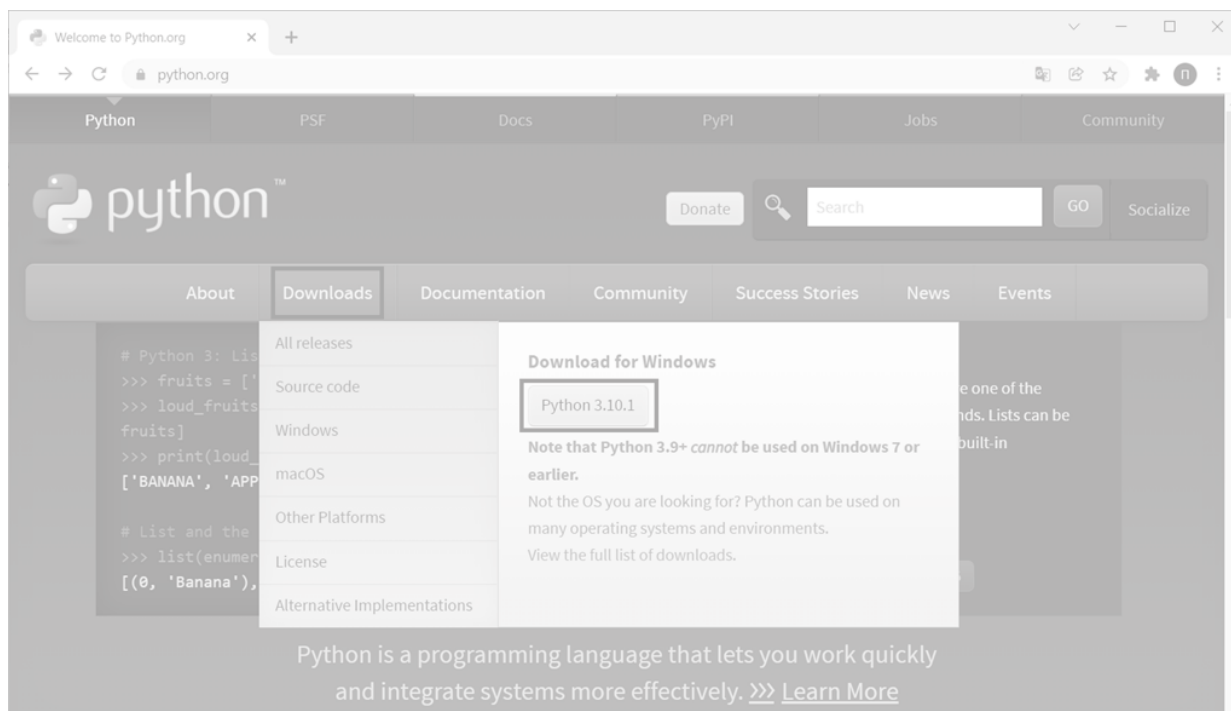
Создание оконных приложений с помощью Python и Tkinter САМОСТОЯТЕЛЬНАЯ РАБОТА

<http://www.kimrt.ru>

Онлайн-курс

[Суперкомпьютерные технологии в задачах моделирования](#)

Установка Python. Загрузите Python с официального сайта (<https://www.python.org/>).



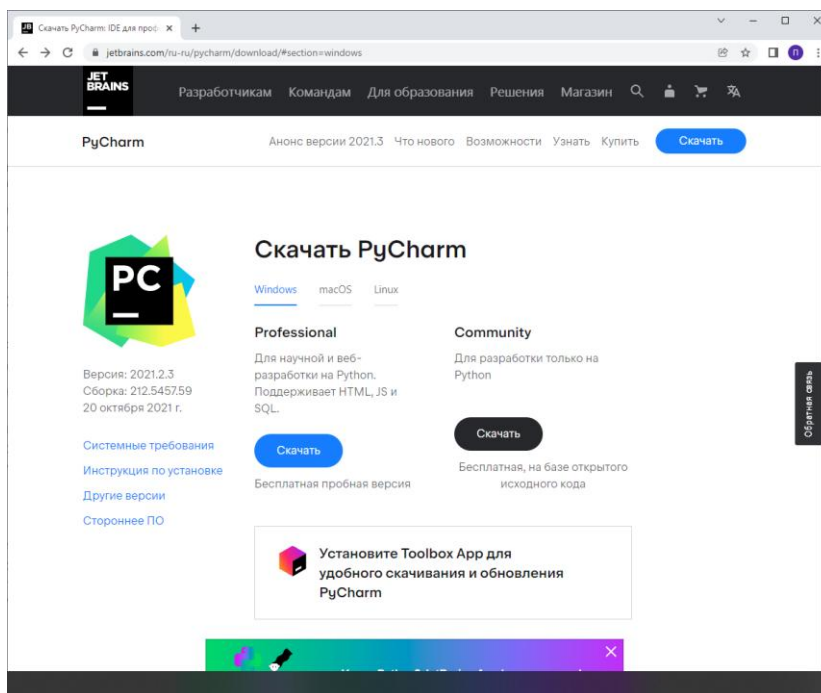
Запустите загруженный файл. Поставьте галочку в пункте «*Add Python to PATH*» и нажмите «*Install Now*». После завершения установки закройте окно.



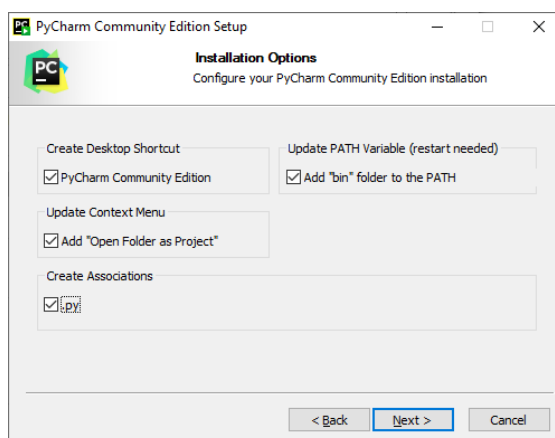
<http://www.kimrt.ru>

Для того, чтобы проверить наличие Python в операционной системе Windows, необходимо открыть консоль (Win+R → cmd) и набрать команду `python -v`. В результате должна появиться надпись с версией Python. Работать можно в консоли после ввода команды `python`, но это крайне неудобно.

Установка интегрированной среды разработки PyCharm. Загрузите PyCharm community с официального сайта (<https://www.jetbrains.com/ru-ru/pycharm/download/#section=windows>).



Запустите загруженный файл и нажмите *Далее*. На следующем этапе можете изменить путь установки, после чего нажмите *Далее*. В следующем окне выберите такие же пункты как на картинке и нажмите *Далее*.



В следующем окне нажмите кнопку *Установить* и дождитесь окончания установки. После завершения установки закройте окно. Откройте PyCharm и

создайте новый проект (File → New project), указав директорию его расположения.

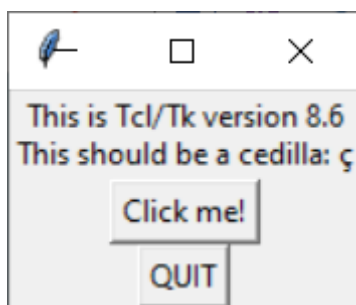
Работа с tkinter. Библиотека *tkinter* по умолчанию устанавливается вместе с *python*, поэтому достаточно просто импортировать ее [1].

```
import tkinter as tk
```

Импортируем библиотеку и назначаем ей псевдоним *tk*. Для проверки работоспособности *tkinter* напишем следующую строчку кода.

```
tkinter._test()
```

При запуске программы у нас должно появиться вот такое окно.



Нажмите *QUIT* для выхода из окна. Удаляем команду *tkinter._test()* из редактора кода. Напишем программу для сложения двух целых чисел и на ее примере рассмотрим некоторые виджеты и менеджеры геометрии. Добавим окно с помощью *Tk()* [2].

```
window = tk.Tk()
```

Также добавим *mainloop()* в конец программы [2].

```
window.mainloop()
```

Метод *mainloop()* необходим для вывода на экран и реагирования на действия пользователя, пока программа не завершится. Этот метод должен располагаться в самом конце программы. Добавим две надписи с помощью *Label()* [2], зададим вывод необходимого сообщения с помощью аргумента *text*. И расположим их с помощью менеджера геометрии *pack()* [3]. Запустим получившуюся программу.

```
import tkinter as tk
```

```
window = tk.Tk()
```

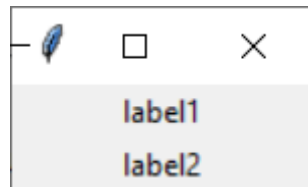
```
label1 = tk.Label(text='label1')
```

```
label2 = tk.Label(text='label2')
```

```
label1.pack()
```

```
label2.pack()
```

```
window.mainloop()
```



Теперь перенесем *mainloop()* на строку выше. В итоге одна из меток не отобразится, потому что код, связанный с графическим интерфейсом, выполняется только до *mainloop()*. Остальной код выполнится только после закрытия окна *tkinter*, и выйдет сообщение об ошибке, связанное с оставшейся строкой *label2.pack()*.



Теперь добавим два поля для ввода с помощью виджета *Entry()*.

```
entry_a = tk.Entry()  
entry_b = tk.Entry()
```

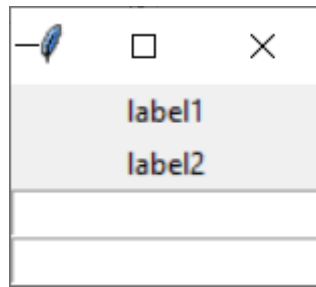
И далее разместим с помощью *pack()*.

```
entry_a.pack()  
entry_b.pack()
```

Получится следующий код.

```
import tkinter as tk  
window = tk.Tk()  
label1 = tk.Label(text='label1')  
label2 = tk.Label(text='label2')  
entry_a = tk.Entry()  
entry_b = tk.Entry()  
label1.pack()  
label2.pack()  
entry_a.pack()  
entry_b.pack()  
window.mainloop()
```

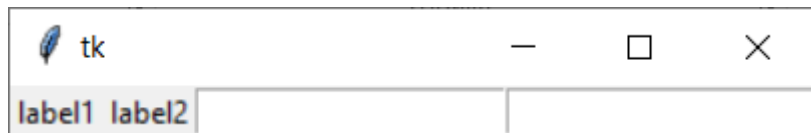
Запустим его и увидим такое окно.



По умолчанию менеджер геометрии *pack()* размещает объект в самую верхнюю незанятую часть окна. С помощью аргумента *side* можно изменить расположения какого-либо объекта. Добавим *side=tk.LEFT* во все методы *pack()*.

```
import tkinter as tk
window = tk.Tk()
label1 = tk.Label(text='label1')
label2 = tk.Label(text='label2')
entry_a = tk.Entry()
entry_b = tk.Entry()
label1.pack(side=tk.LEFT)
label2.pack(side=tk.LEFT)
entry_a.pack(side=tk.LEFT)
entry_b.pack(side=tk.LEFT)
window.mainloop()
```

Запустим программу.



Также *pack()* располагает элементы в такой последовательности, в которой они указаны в коде. Изменим код, переместив *label2.pack()* между *entry_a.pack()* и *entry_b.pack()*.

```
import tkinter as tk
window = tk.Tk()
label1 = tk.Label(text='label1')
label2 = tk.Label(text='label2')
entry_a = tk.Entry()
entry_b = tk.Entry()
label1.pack(side=tk.LEFT)
entry_a.pack(side=tk.LEFT)
label2.pack(side=tk.LEFT)
entry_b.pack(side=tk.LEFT)
```

```
entry_b.pack(side=tk.LEFT)
window.mainloop()
```

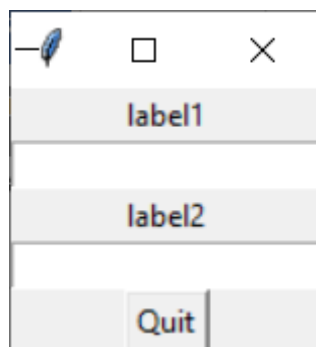


Теперь добавим кнопку выхода с помощью *Button()*.

```
button_quit = tk.Button(text="Quit",
command=window.destroy)
```

Параметр *text* позволяет выводить надпись на кнопке, *command* – закрывать наше окно, созданное командой *window = tk.Tk()*.

```
import tkinter as tk
window = tk.Tk()
label1 = tk.Label(text='label1')
label2 = tk.Label(text='label2')
button_quit = tk.Button(text="Quit",
command=window.destroy)
entry_a = tk.Entry()
entry_b = tk.Entry()
label1.pack()
entry_a.pack()
label2.pack()
entry_b.pack()
button_quit.pack()
window.mainloop()
```



Теперь напишем программу которая складывает два целых числа. Добавим метку, в которой будет выводиться результат и кнопку, которая будет выполнять действие.

```
label_a_plus_b = tk.Label(text='')
```

```
button_a_plus_b = tk.Button(text='=')
```

Напишем функцию, которая будет выполняться при нажатии кнопки. С помощью функции *get()* получим значение из поля *Entry*.

```
def plus(event):
```

```
# получаем значение в переменную a
```

```
    a = int(entry_a.get())
```

```
# получаем значение в переменную b
```

```
    b = int(entry_b.get())
```

```
# Складываем a и b и записываем в аргумент text
```

```
# метки label_a_plus_b.
```

```
    label_a_plus_b["text"] = str(a + b)
```

Присвоим выполнение функции кнопке.

```
button_a_plus_b.bind("<Button-1>", plus)
```

Расположим кнопку и метку с помощью *pack()*.

```
button_a_plus_b.pack()
```

```
label_a_plus_b.pack()
```

Весь код программы выглядит следующим образом

```
import tkinter as tk
```

```
def plus(event):
```

```
    a = int(entry_a.get())
```

```
    b = int(entry_b.get())
```

```
    label_a_plus_b["text"] = str(a + b)
```

```
window = tk.Tk()
```

```
label_a = tk.Label(text='a')
```

```
label_b = tk.Label(text='+\nb')
```

```
label_a_plus_b = tk.Label(text='')
```

```
entry_a = tk.Entry()
```

```
entry_b = tk.Entry()
```

```
button_a_plus_b = tk.Button(text='=')
```

```
button_a_plus_b.bind("<Button-1>", plus)
```

```
button_quit = tk.Button(text="Quit",
```

```
command=window.destroy)
```

```
label_a.pack()
```

```
entry_a.pack()
```

```
label_b.pack()
```

```
entry_b.pack()
```

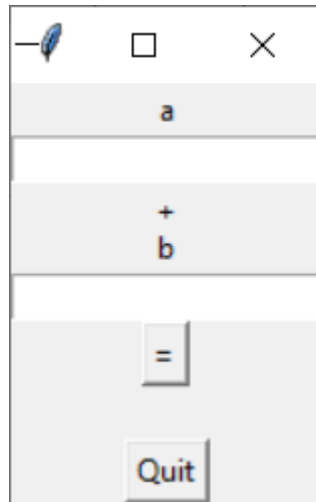
```
button_a_plus_b.pack()
```

```
label_a_plus_b.pack()
```

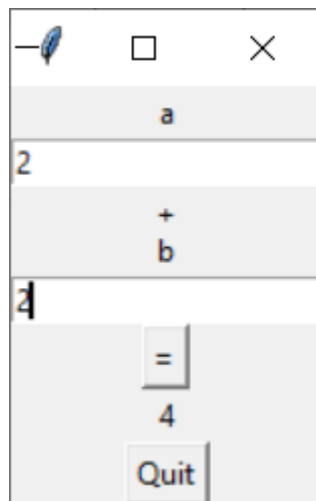
```
button_quit.pack()
```

```
window.mainloop()
```

Результат выполнения программы.



Заполним поля а и b и нажмем кнопку “=”.



http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Tkinter docs. URL: <https://docs.python.org/3/library/tkinter.html>
2. Hello World tkinter program. URL: <https://docs.python.org/3/library/tkinter.html#a-hello-world-program>
3. Tkinter packer. URL: <https://docs.python.org/3/library/tkinter.html#the-packer>