

Напомним, что ранее мы создали подобную программу (она будет пока нашим шаблоном).

```
from tkinter import * # подключаем модуль tkinter
tk = Tk() # создаем объект главного окна
tk.title('Название команды') # задаем заголовок окна
tk.geometry("1024x640+0+0") # задаем размеры и смещение окна
tk.resizable(width=False, height=False) # запрещаем менять размеры окна
```

**ЗДЕСЬ МЫ БУДЕМ РАЗМЕЩАТЬ ОСНОВНОЙ ТЕКСТ ПРОГРАММЫ**

```
tk.mainloop()
```

## Виджет *Canvas* (Холст). Метод *pack*

*Canvas* — это объект библиотеки *tkinter*, представляющий собой прямоугольную область. В этой области можно размещать другие объекты: графику, текст и другие виджеты. Отображенные на холсте объекты можно изменять и перемещать.

При создании объекта-экземпляра *Canvas* можно указать его родительский объект, ширину, высоту и цвет фона.

```
c = Canvas(tk, width=1024, height=640, bg='#ffff00')
c = Canvas(tk, width=1024, height=640)
```

В нашем случае родительским объектом будет служить созданный нами ранее объект окна *tk*. Ширина и высота холста может совпадать с указанными нами ранее размерами окна. Цвет фона холста указывать необязательно.

Чтобы отобразить любой созданный объект в другом объекте-контейнере, необходимо вызвать один из трех методов — они называются *менеджеры геометрии*. Это методы *pack*, *place* и *grid*. Если к элементу интерфейса не применить какой-либо из этих методов, то он не отобразится в окне программы. При этом в одном окне (или любом другом родительском виджете) нельзя комбинировать разные менеджеры геометрии.

Мы воспользуемся методом *pack* (упаковать).

```
c.pack()
```

Попробуйте задавать в своей программе различные размеры холста (меньше размера окна и больше размера окна) и различный цвет фона.

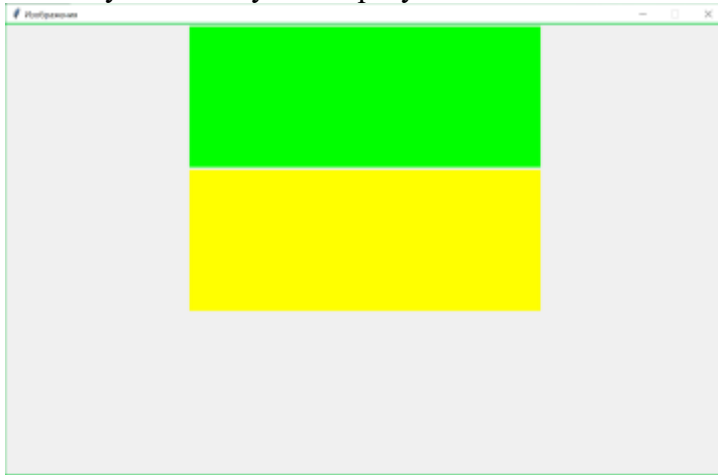
Если в *pack* не передавать аргументы, то виджеты будут располагаться вертикально друг над другом. Тот объект, который первым вызовет *pack*, будет вверху. Вызвавший вторым — под первым, и т. д.

Давайте посмотрим, как это работает. Добавьте следующий код:

```
c1 = Canvas(tk, width=500, height=200, bg='#00ff00')
c2 = Canvas(tk, width=500, height=200, bg='#ffff00')
c1.pack()
c2.pack()
```

Здесь мы создаем два холста одинакового размера, но разного цвета: *c1* — зеленый, *c2* — желтый.

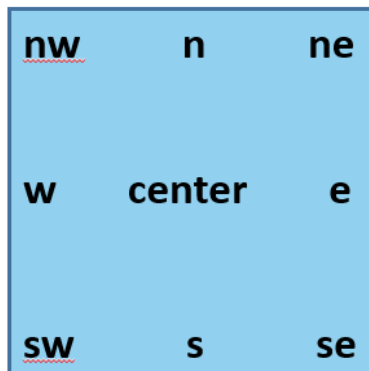
Вы получите следующий результат:



Поменяйте местами две последние строки кода. Теперь желтый холст будет над зеленым.

У метода *pack* (и у многих других методов) есть **параметр точки привязки *anchor***. В привязке указывается, где указанный объект будет находиться по отношению к другому, если этот второй больше первого – то есть помещает виджет в определенной части контейнера. Например, положение холста внутри окна или положение изображения внутри холста. Эти константы задаются как точки компаса, где север находится вверху, а запад – слева.

Варианты  
значения  
параметра  
*anchor*



Давайте поэкспериментируем на примере нашего холста. Замените методы *pack* без параметров на следующий код:

```
c1.pack(anchor=SE)
c2.pack(anchor=NW)
```

Обратите внимание, что для наших экземпляров холста работает только расположение по горизонтали – все равно зеленый холст будет находиться выше желтого.

Если вы оставите один экземпляр холста:

```
c = Canvas(tk, width=500, height=200, bg='#00ff00')
```

и будете менять для него разные точки привязки, вы увидите, что тоже работает только расположение по горизонтали.

Дело в том, что у метода *pack* есть еще **параметр *side* (сторона)**, который принимает одно из четырех значений – *TOP*, *BOTTOM*, *LEFT*, *RIGHT* (верх, низ, лево, право). По умолчанию, когда в *pack* не указывается параметр *side*, его значение равняется *TOP*. Из-за этого наши виджеты располагаются вертикально друг за другом.

Если вы укажете следующие параметры для наших двух холстов:

```
c1.pack(side=BOTTOM)
```

```
c2.pack(side=BOTTOM)
```

то обнаружите, что желтый окажется выше зеленого. Поменяйте местами эти две строки – теперь зеленый выше желтого. То есть показываются объекты в порядке вызова метода *pack*, но относительно стороны привязки.

Оставьте один холст и поэкспериментируйте с ним, задавая разные значения параметра *side*. Потом сделайте это с двумя холстами.

**Параметр *expand*** используется для растяжения виджета и может принимать значение *FALSE* (по умолчанию) или *TRUE* (можно задать и числовые значения 0 и 1). При отсутствии других параметров позиционирования значение ***expand=TRUE*** позволяет поместить виджет по центру контейнера.

Задайте только этот параметр для одного созданного холста в окне с меняющимися размерами и попробуйте изменять при работе программы размеры окна – вы увидите, что холст неизменно остается в центре окна.

**Параметр *fill*** определяет, будет ли виджет растягиваться, чтобы заполнить свободное пространство вокруг. Этот параметр может принимать следующие значения: *NONE* (по умолчанию, элемент не растягивается), *X* (элемент растягивается только по горизонтали), *Y* (элемент растягивается только по вертикали) и *BOTH* (элемент растягивается по вертикали и горизонтали). Поэкспериментируйте с разными значениями параметра.

### ВНИМАНИЕ!!!

Синтаксис *Питона* позволяет задавать значения подобных параметров в двух вариантах: значение в кавычках передается в нижнем регистре, без кавычек – в верхнем регистре. Например:

```
side=BOTTOM
```

```
side='bottom'
```

При размещении на холсте геометрических и других объектов указываются их координаты на холсте. Точкой отсчета является верхний левый угол холста.

