

Урок 15. Метод place()

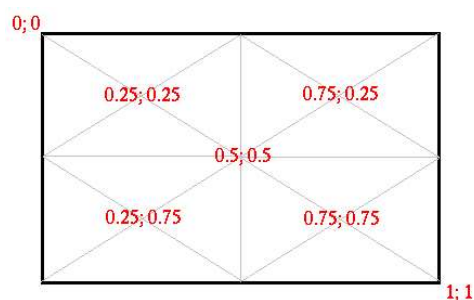
В конце курса рассмотрим третий и последний менеджер геометрии библиотеки tk – Place, который размещает виджеты по координатам. В tkinter использование данного управляющего размещением реализуется через метод place() виджетов.

Методом place() виджету указывается его положение либо в абсолютных значениях (в пикселях), либо в долях родительского окна, то есть относительно. Также абсолютно и относительно можно задавать размер самого виджета.

Основными параметрами place() являются:

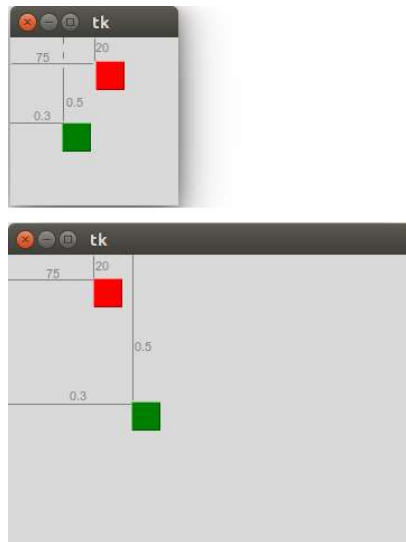
- anchor (якорь) – определяет часть виджета, для которой задаются координаты. Принимает значения N, NE, E, SE, SW, W, NW или CENTER. По умолчанию NW (верхний левый угол).
- relwidth, relheight (относительные ширина и высота) – определяют размер виджета в долях его родителя.
- relx, rely – определяют относительную позицию в родительском виджете. Координата (0; 0) – у левого верхнего угла, (1; 1) – у правого нижнего.
- width, height – абсолютный размер виджета в пикселях. Значения по умолчанию (когда данные опции опущены) приравниваются к естественному размеру виджета, то есть к тому, который определяется при его создании и конфигурировании.
- x, y – абсолютная позиция в пикселях. Значения по умолчанию приравниваются к нулю.

Схема с указанием относительных координат:



Для лучшего понимания разницы между абсолютным и относительным позиционированием рассмотрим пример:

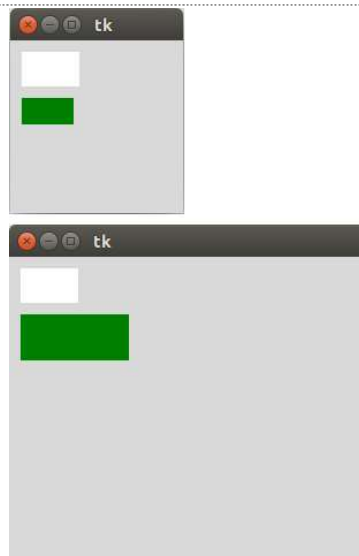
```
from tkinter import *  
  
root = Tk()  
root.geometry('150x150')  
  
Button(bg='red').place(x=75, y=20)  
Button(bg='green').place(relx=0.3, rely=0.5)  
  
root.mainloop()
```



Кнопка, позиция которой была жестко задана, не изменяет своего положения, когда окно раздвигается. Кнопка с относительными координатами смещается. Опции `relx` и `rely` для нее по-прежнему 0.3 и 0.5, но уже относительно нового размера окна.

Тоже самое с размерами виджетов. Если они относительны, то с изменением размера родительского виджета, их размеры также будут изменяться. Если требуется, чтобы виджет всегда сохранял свой размер, последний должен указываться как абсолютный. Или не указываться вовсе, тогда будет приравнен к естественному.

```
...
Label(bg='white').place(x=10, y=10,
                        width=50, height=30)
Label(bg='green').place(x=10, y=50,
                        relwidth=0.3, relheight=0.15)
...
```



Комбинируя различные варианты позиционирования и установки размеров, можно добиться интересных эффектов, а также неожиданных спецэффектов. Поэтому методом `place()` следует пользоваться с осторожностью, хорошо понимая, что вы делаете.

Если размер окна не меняется, а интерфейс сложен, то вероятно управляющий размещением `Place` может быть лучшим выбором, так как с его помощью можно выполнить тонкую настройку и создать наиболее аккуратный интерфейс.

Практическая работа

За рамками данного курса было оставлено несколько классов пакета tkinter. Среди них PhotoImage, позволяющий использовать в программе внешние изображения форматов GIF и PGM. Экземпляры PhotoImage можно размещать на различных виджетах через опцию image.

```
...  
img = PhotoImage(file='smile.gif')  
l=Label(image=img)  
...
```

Напишите программу, состоящую из главного окна и кнопки, на которой изображен смайлик. При клике на кнопку она должна оказываться в новом случайном месте окна. Размер окна может меняться.

