

Урок 12. Метод grid()

Grid является одним из трех менеджеров геометрии в Tkinter (другими являются уже рассмотренный ранее Pack, а также Place). У всех виджетов есть соответствующий данному менеджеру метод grid(). "Grid" с английского переводится как "сетка", однако по смыслу правильнее говорить о таблице.

Табличный способ размещения предпочтителен из-за его гибкости и удобства, когда дело доходит до разработки относительно сложных интерфейсов. Grid позволяет избежать использования множества фреймов, что неизбежно в случае упаковщика Pack.

При размещении виджетов методом grid() родительский контейнер (обычно это окно) условно разделяется на ячейки подобно таблице. Адрес каждой ячейки состоит из номера строки и номера столбца. Нумерация начинается с нуля. Ячейки можно объединять как по вертикали, так и по горизонтали.

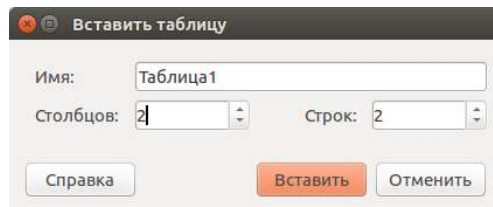
0, 0		
1, 0	1, 1	1, 2
2, 0		2, 2
3, 0		3, 2
4, 0		

На рисунке пунктир обозначает объединение ячеек. Общая ячейка в таком случае обозначается адресом первой.

Никаких предварительных команд по разбиению родительского виджета на ячейки не выполняется. Tkinter делает это сам, исходя из указанных позиций виджетов.

Размещение виджета в той или иной ячейке задается через аргументы row и column, которым присваиваются соответственно номера строки и столбца. Чтобы объединить ячейки по горизонтали, используется атрибут columnspan, которому присваивается количество объединяемых ячеек. Опция rowspan объединяет ячейки по вертикали.

Пусть надо запрограммировать такой GUI:



Представим данный интерфейс в виде таблицы и пронумеруем ячейки, в которых будут располагаться виджеты (подобную разбивку можно произвести в уме):

0, 0 Имя: Таблица 1	0, 1, columnspan=3	
1, 0 Столбцов: 2	1, 1	1, 2 Строк: 2
2, 0 Справка		2, 2 Вставить
		2, 3 Отменить

Теперь пишем код:

```
from tkinter import *
root = Tk()

Label(text="Имя:").grid(row=0, column=0)
table_name = Entry(width=30)
table_name.grid(row=0, column=1, columnspan=3)

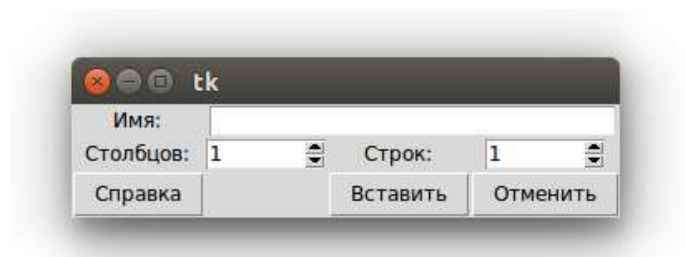
Label(text="Столбцов:").grid(row=1, column=0)
table_column = Spinbox(width=7, from_=1, to=50)
table_column.grid(row=1, column=1)
Label(text="Строк:").grid(row=1, column=2)
table_row = Spinbox(width=7, from_=1, to=100)
table_row.grid(row=1, column=3)

Button(text="Справка").grid(row=2, column=0)
Button(text="Вставить").grid(row=2, column=2)
Button(text="Отменить").grid(row=2, column=3)

root.mainloop()
```

Примечание. В примере используются виджеты класса `Spinbox`, которые не рассматривались в курсе. `Spinbox` похож на `Entry`, но для него задается список принимаемых значений, и имеется подобие скроллера.

Выполнив приведенный выше программный код, получим:



Похоже, но не совсем то, что хотелось. Теперь на помощь должны прийти другие свойства метода `grid()`. У него, также как у `pack()`, имеются атрибуты для задания внешних и внутренних отступов (`padx`, `pady`, `ipadx`, `ipady`).

Кроме этого есть атрибут `sticky` (липкий), который принимает значения направлений сторон света (N, S, W, E, NW, NE, SW, SE). Если, например, указать NW, то виджет прильнет к верхнему левому углу ячейки. Виджеты можно растягивать на весь объем ячейки (`sticky=N+S+W+E`) или только по одной из осей (N+S или W+E). Эффект от "липучки" заметен, только если виджет меньше ячейки.

```
from tkinter import *
root = Tk()

Label(text="Имя:").grid(row=0, column=0, sticky=W, pady=10, padx=10)
table_name = Entry()
table_name.grid(row=0, column=1, columnspan=3, sticky=W+E, padx=10)

Label(text="Столбцов:").grid(row=1, column=0, sticky=W, padx=10, pady=10)
table_column = Spinbox(width=7, from_=1, to=50)
table_column.grid(row=1, column=1, padx=10)
Label(text="Строк:").grid(row=1, column=2, sticky=E)
```

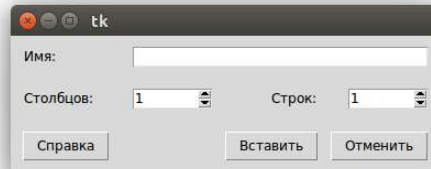
```

table_row = Spinbox(width=7, from_=1, to=100)
table_row.grid(row=1, column=3, sticky=E, padx=10)

Button(text="Справка").grid(row=2, column=0, pady=10, padx=10)
Button(text="Вставить").grid(row=2, column=2)
Button(text="Отменить").grid(row=2, column=3, padx=10)

root.mainloop()

```



С помощью методов `grid_remove()` и `grid_forget()` можно сделать виджет невидимым. Отличие между этими методами лишь в том, что `grid_remove()` запоминает прежнее положение виджета. Поэтому для его отображения в прежней ячейки достаточно применить `grid()` без аргументов. После `grid_forget()` потребуется заново конфигурировать положение виджета.

Скрытие виджетов бывает необходимо в тех случаях, когда, например, от выбора пользователя в одной части интерфейса зависит, какие виджеты появятся в другой.

Практическая работа

Перепрограммируйте второе окно из практической работы предыдущего урока, используя метод `grid()`.

