

Позиционирование Pack

Последнее обновление: 11.09.2022

<https://metanit.com/python/tkinter/2.4.php>

Для позиционирования виджетов в контейнере применяются различные способы. Один из них представляет вызов у виджета метода `pack()`. Этот метод принимает следующие параметры:

`expand`: если равно `True`, то виджет заполняет все пространство контейнера.

`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).

`anchor`: помещает виджет в определенной части контейнера. Может принимать значения `n`, `e`, `s`, `w`, `ne`, `nw`, `se`, `sw`, `c`, которые являются сокращениями от `Noth`(север - вверх), `South` (юг - низ), `East` (восток - правая сторона), `West` (запад - левая сторона) и `Center` (по центру). Например, значение `nw` указывает на верхний левый угол

`side`: выравнивает виджет по одной из сторон контейнера. Может принимать значения: `TOP` (по умолчанию, выравнивается по верхней стороне контейнера), `BOTTOM` (выравнивание по нижней стороне), `LEFT` (выравнивание по левой стороне), `RIGHT` (выравнивание по правой стороне).

`ipadx`: устанавливает отступ содержимого виджета от его границы по горизонтали.

`ipady`: устанавливают отступ содержимого виджета от его границы по вертикали.

`padx`: устанавливает отступ виджета от границ контейнера по горизонтали.

`pady`: устанавливает отступ виджета от границ контейнера по вертикали.

Растяжение виджета

Для растяжения виджета применяется параметру `expand` передается значение `True` (или соответствующее число). Причем при отсутствии других параметров позиционирования значение `expand=True` позволяет поместить виджет по центру:

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")

btn = ttk.Button(text="Click me")
btn.pack(expand=True)

root.mainloop()
```

Anchor

Параметр `anchor` помещает виджет в определенной части контейнера. Может принимать следующие значения:

n: положение вверху по центру

e: положение в правой части контейнера по центру

s: положение внизу по центру

w: положение в левой части контейнера по центру

nw: положение в верхнем левом углу

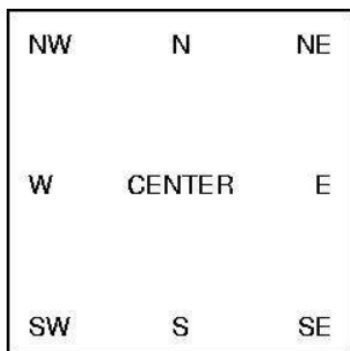
ne: положение в верхнем правом углу

se: положение в нижнем правом углу

sw: положение в нижнем левом углу

center: положение центру

Схематически это выглядит следующим образом:



Стоит отметить, что значение в кавычках для параметра anchor передается в нижнем регистре, без кавычек - в верхнем регистре

```
btn.pack(anchor="nw")
btn.pack(anchor=NW)
```

Также стоит отметить, что для некоторых сценариев (например, помещение в нижней части контейнера) может потребоваться указать для параметра expand значение True. Например, поместим кнопку в верхнем левом углу:

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")

btn = ttk.Button(text="Click me")
btn.pack(anchor="nw")

root.mainloop()
```

Заполнение контейнера

Параметр `fill` позволяет заполнить пространство контейнер по вертикали (значение `X`), по вертикали (значение `Y`) или по обеим сторонам (значение `BOTH`). По умолчанию значение `NONE`, при котором заполнение контейнера отсутствует. Например, заполним все пространство контейнера по горизонтали

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")

btn = ttk.Button(text="Click me")
btn.pack(fill=X)

root.mainloop()
```

Для заполнения контейнера по всем сторонам также требуется установить параметр `expand = True`

```
btn.pack(fill=BOTH, expand=True)
```

Отступы

Параметры `padx` и `pady` позволяют указать отступы виджета от границ контейнера:

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")

btn = ttk.Button(text="Click me")
btn.pack(anchor="nw", padx=20, pady=30)

root.mainloop()
```

Здесь кнопка смещена относительно верхнего левого угла на 20 единиц вправо и на 30 единиц вниз

Выше устанавливался общий отступ от левой и правой стороны и общий отступ от верхней и нижней кромки контейнера. Поскольку кнопка позиционировалась в верхнем левом углу и имела небольшие размеры, отступ от нижней и правой кромки контейнера нас не особо интересовали. Однако при желании мы можем задать отдельно два отступа от правой и левой границы и отдельно два отступа от верхней и нижней границ:

```
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")
```

```
btn = ttk.Button(text="Click me")
btn.pack(fill=X, padx=[20, 60], pady=30)
```

```
root.mainloop()
```

В данном случае отступ слева - 20 единиц, а справа - 60 единиц

Параметры `ipadx` и `ipady` позволяют указать отступы содержимого виджета от границ виджета по горизонтали и вертикали соответственно:

```
from tkinter import *
from tkinter import ttk
```

```
root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")
```

```
btn = ttk.Button(text="Click me")
btn.pack(expand=True, ipadx=10, ipady=10)
```

```
root.mainloop()
```

Позиционирование по стороне

Используем параметр `side`:

```
from tkinter import *
from tkinter import ttk
```

```
root = Tk()
root.title("METANIT.COM")
root.geometry("250x200")
```

```
btn1 = ttk.Button(text="BOTTOM")
btn1.pack(side=BOTTOM)
```

```
btn2 = ttk.Button(text="RIGHT")
btn2.pack(side=RIGHT)
```

```
btn3 = ttk.Button(text="LEFT")
btn3.pack(side=LEFT)
```

```
btn4 = ttk.Button(text="TOP")
btn4.pack(side=TOP)
```

```
root.mainloop()
```

Комбинируя параметры `side` и `fill`, можно растянуть элемент по вертикали:

```
btn1 = ttk.Button(text="CLICK ME")
btn1.pack(side=LEFT, fill=Y)
```