

Python - Tkinter grid() Method

This geometry manager organizes widgets in a table-like structure in the parent widget.

Syntax

```
widget.grid( grid_options )
```

Here is the list of possible options –

- **column** – The column to put widget in; default 0 (leftmost column).
- **columnspan** – How many columns widget occupies; default 1.
- **ipadx, ipady** – How many pixels to pad widget, horizontally and vertically, inside widget's borders.
- **padx, pady** – How many pixels to pad widget, horizontally and vertically, outside v's borders.
- **row** – The row to put widget in; default the first row that is still empty.
- **rowspan** – How many rows widget occupies; default 1.
- **sticky** – What to do if the cell is larger than widget. By default, with sticky="", widget is centered in its cell. sticky may be the string concatenation of zero or more of N, E, S, W, NE, NW, SE, and SW, compass directions indicating the sides and corners of the cell to which widget sticks.

Example

Try the following example by moving cursor on different buttons –

```
import Tkinter
root = Tkinter.Tk( )
for r in range(3):
    for c in range(4):
        Tkinter.Label(root, text='R%s/C%s'%(r,c),
            borderwidth=1 ).grid(row=r,column=c)
root.mainloop( )
```

This would produce the following result displaying 12 labels arrayed in a 3 × 4 grid –

