

05. Presentation Layouts



Bokeh Tutorial

(<http://bokeh.pydata.org/>)

```
In [1]: from bokeh.io import output_notebook, show
        from bokeh.plotting import figure

        output_notebook()
```

(<http://bokeh.pydata.org/>) BokehJS 0.12.14 successfully loaded.

In the previous chapters we started to learn how to create single plots using different kinds of data. But we often want to plot more than one thing. Bokeh plots can be individually embedded in HTML documents, but it's often easier to combine multiple plots in one of Bokeh's built-in layouts. We will learn how to do that in this chapter.

The cell below defines a few data variables we will use in examples.

```
In [2]: x = list(range(11))
        y0, y1, y2 = x, [10-i for i in x], [abs(i-5) for i in x]
```

Rows and Columns

The `bokeh.layouts` module provides the `row` and `column` functions to arrange plot objects in vertical or horizontal layouts. Below is an example of three plots arranged in a row.

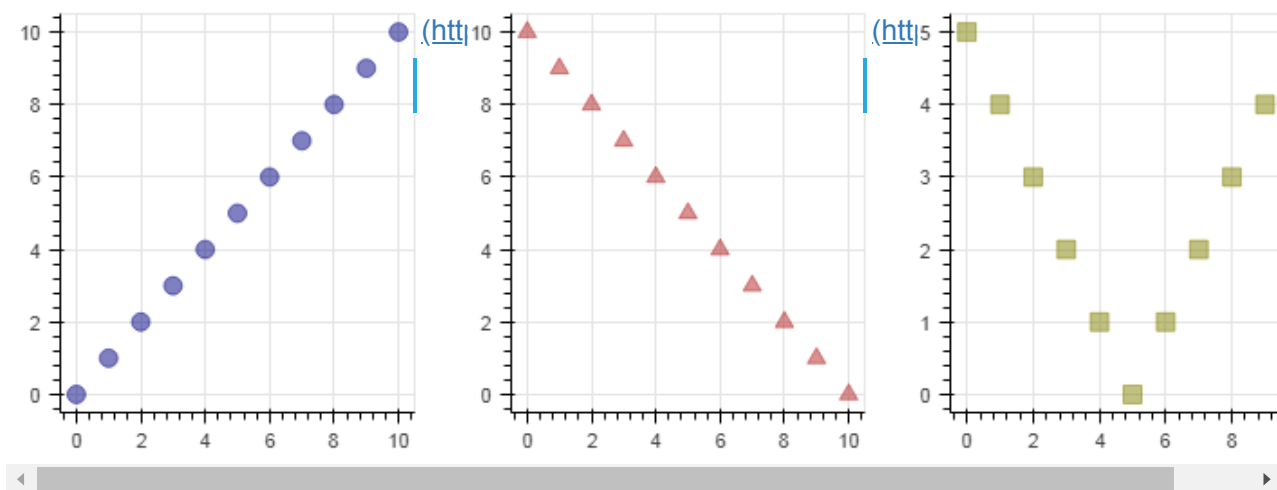
```
In [3]: from bokeh.layouts import row

# create a new plot
s1 = figure(width=250, plot_height=250)
s1.circle(x, y0, size=10, color="navy", alpha=0.5)

# create another one
s2 = figure(width=250, height=250)
s2.triangle(x, y1, size=10, color="firebrick", alpha=0.5)

# create and another
s3 = figure(width=250, height=250)
s3.square(x, y2, size=10, color="olive", alpha=0.5)

# show the results in a row
show(row(s1, s2, s3))
```



```
In [4]: # EXERCISE: use column to arrange a few plots vertically
```

Grid plots

Bokeh also provides a `gridplot` layout in `bokeh.layouts` for arranging plots in a grid, as show in the example below.

```
In [4]: from bokeh.layouts import gridplot

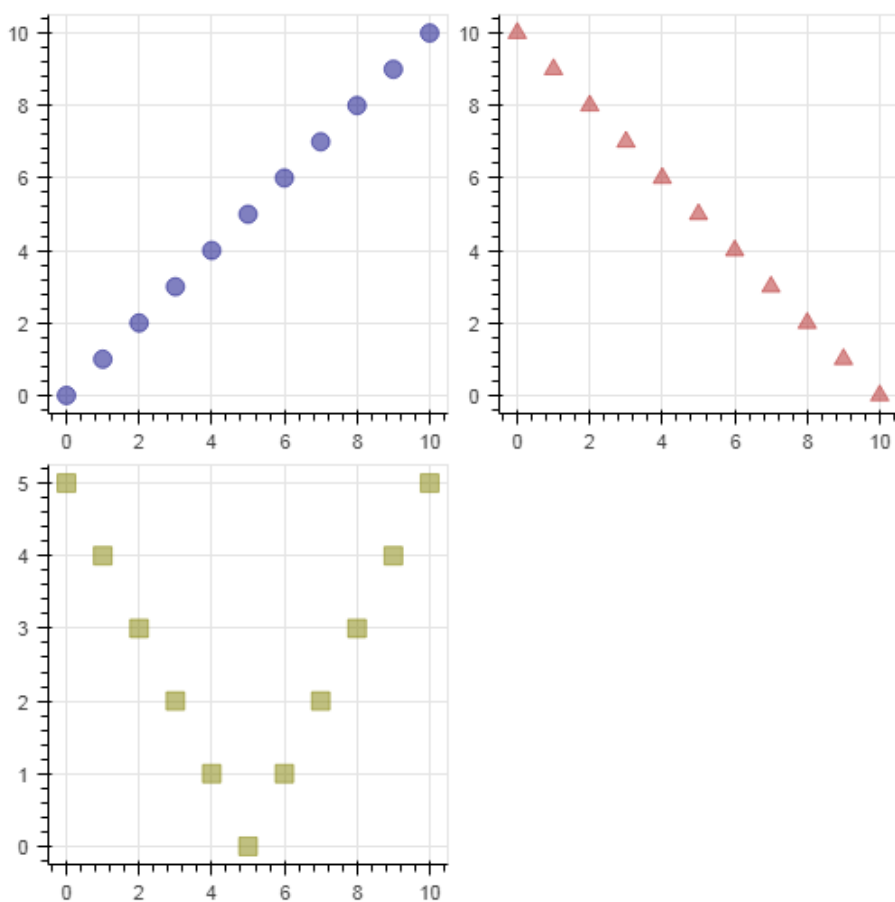
# create a new plot
s1 = figure(width=250, plot_height=250)
s1.circle(x, y0, size=10, color="navy", alpha=0.5)

# create another one
s2 = figure(width=250, height=250)
s2.triangle(x, y1, size=10, color="firebrick", alpha=0.5)

# create and another
s3 = figure(width=250, height=250)
s3.square(x, y2, size=10, color="olive", alpha=0.5)

# put all the plots in a gridplot
p = gridplot([[s1, s2], [s3, None]], toolbar_location=None)

# show the results
show(p)
```



```
In [6]: # EXERCISE: create a gridplot of your own
```

```
In [ ]:
```