quickplots Documentation

Release 2.1.0

Sam Ireland

February 05, 2017

Contents

quickplots is a lightweight, object-oriented plotting library for Python. It currently supports line charts.

CHAPTER 1

Example

```
>>> import quickplots
>>> chart = quickplots.line((0, 0), (1, 1), (2, 4), (3, 9), name="squares")
>>> chart.get_series_by_name("squares").color("#FF0000")
>>> chart.create().save("chart.svg")
```

CHAPTER 2

Table of Contents

2.1 Installing

2.1.1 pip

quickplots can be installed using pip:

```
$ pip install quickplots
```

quickplots is written for Python 3. If the above installation fails, it may be that your system uses `pip` for the Python 2 version - if so, try:

```
$ pip3 install quickplots
```

2.1.2 Requirements

quickplots relies on OmniCanvas for its graphics capabilities. If you install quickplots using pip this library will be installed automatically. It also requires numerus.

Otherwise quickplots has no external dependencies, and is pure Python.

2.2 Overview

2.2.1 Creating Charts

The easiest way to create a chart is with the quick-add functions. For example, the following code would create a line chart of the sine function:

```
>>> import quickplots
>>> from math import sin, radians
>>> data = [(x, sin(radians(x))) for x in range(360)]
>>> chart = quickplots.line(*data, color="#0000FF", title="sin(x)")
>>> chart.x_label("angle")
>>> chart.y_label("sine of angle")
```

A few things to note here. The `line()` function takes (x, y) data points as its positional arguments - these can be lists or tuples. All arguments that are not data must be given as keyword arguments.

As with all other quickplots functions that accept data in this way, you can also provide the data in the form of two lists (or tuples) - one of all the x values and one of all the y values:

```
>>> chart = quickplots.line([x for x in range(360)], [sin(radians(x)) for x in range(360)])
```

Line charts

Use the `line()` function as above to create line charts. You can pass in hex colors to the `color` function and line styles (see the full documentation for a full list of styles) to the `linestyle` argument.

Charts themselves also have a `line()` method for adding new line series. To add the cosine function to the above chart, you would do the following:

```
>>> cosine_data = [(x, cos(radians(x))) for x in range(360)]
>>> chart.line(*cosine_data, color="#00FF00")
```

Scatter charts

`scatter()` will create a scatter chart. You can set the size of the points with the `size` argument, as well as their `color` and `linewidth` (the width of the points' border).

Charts themselves also have a `scatter()` method for adding new line series. To add the cosine function to the above chart, you would do the following:

```
>>> cosine_data = [(x, cos(radians(x))) for x in range(360)]
>>> chart.scatter(*cosine_data, color="#00FF00")
```

2.2.2 Modifying Charts

Charts have a title, an x axis label, and a y axis label, which can be modified like so:

```
>>> chart.title()
'sin(x)'
>>> chart.title("A new title")
>>> chart.title()
'A new title'
>>> chart.x_label("A new x-axis label")
>>> chart.y_label("A new y-axis label")
```

Ticks will be automatically generated, but if you want to specify your own you can specify your own:

```
>>> chart.x_ticks(0, 90, 180, 270, 360)
>>> chart.x_ticks()
(0, 90, 180, 270, 360)
```

Charts can have one or more `Series` objects. The `series` property will return the first series, and the `all_series` will return all the series on the chart.

See the documentation for `Chart` and `AxisChart` for more information.

2.2.3 Outputting Charts

All charts have a `create()` method which will create an OmniCanvas **‘canvas <https://omnicanvas.readthedocs.io/en/latest/api/canvas.htm\ l#omnicanvas.canvas.Canvas>‘_** with the chart painted to it. These can be saved or rendered as SVG text.

```
>>> chart.create()
<Canvas 700×500 (7 Graphics)>
>>> chart.create().save("Charts.svg")
```

2.3 Full API

2.3.1 `quickplots.series` (Data Series)

2.3.2 `quickplots.charts` (Charts)

2.3.3 `quickplots.quick` (Quick-add functions)

2.4 Changelog

2.4.1 Release 2.1.0

5 February 2017

- Added Scatter series for scatter plots.
- Charts now have ticks and grid lines.
- Added new colour palette and colour generation.
- Line charts can now set the width of their line.

2.4.2 Release 2.0.0

9 January 2017

- Remade quickplots as dependent on OmniCanvas for its graphics rendering.
- Added LineSeries and AxisCharts.
- Added quick-add function for easy creation of line charts.