



Fork me on GitHub

Version 3.1.1

[home](#) | [examples](#) | [tutorials](#) | [API](#) | [contents](#) » [API Overview](#) »

[previous](#) | [next](#) | [modules](#) | [index](#)

matplotlib.markers

This module contains functions to handle markers. Used by both the marker functionality of `plot` and `scatter`.

All possible markers are defined here:

marker	symbol	description
"."		point
","		pixel
"o"		circle
"v"		triangle_down
"^"		triangle_up
"<"		triangle_left
">"		triangle_right
"1"		tri_down
"2"		tri_up
"3"		tri_left
"4"		tri_right
"8"		octagon
"s"		square
"p"		pentagon
"P"		plus (filled)
"*"		star
"h"		hexagon1
"H"		hexagon2





















Quick search

Go

Table of Contents

- [matplotlib.markers](#)
- Classes
- Related Topics
- Documentation overview
- API Overview
 - Previous:
[matplotlib.lines.segment_hits](#)
 - Next:
[matplotlib.markers.MarkerStyle](#)

Show Page Source

"+"		plus
"x"		x
"X"		x (filled)
"D"		diamond
"d"		thin_diamond
" "		vline
"_"		hline
0 (TICKLEFT)		tickleft
1 (TICKRIGHT)		tickright
2 (TICKUP)		tickup
3 (TICKDOWN)		tickdown
4 (CARETLEFT)		caretleft
5 (CARETRIGHT)		caretright
6 (CARETUP)		caretup
7 (CARETDOWN)		caretdown
8 (CARETLEFTBASE)		caretleft (centered at base)
9 (CARETRIGHTBASE)		caretright (centered at base)
10 (CARETUPBASE)		caretup (centered at base)
11 (CARETDOWNBASE)		caretdown (centered at base)
"None", " " or ""		nothing
'\$...\$'		Render the string using mathtext. E.g "\$\text{f}\$" for marker showing the letter f.
verts		A list of (x, y) pairs used for Path vertices. The center of the marker is located at (0,0) and the size is normalized, such that the created path is encapsulated inside the unit cell.
path		A Path instance.
(numsides, style, angle)		The marker can also be a tuple (numsides, style, angle), which will create a custom, regular symbol.

`numsides`:
the number of sides

`style`:
the style of the regular symbol:

- 0: a regular polygon
- 1: a star-like symbol
- 2: an asterisk
- 3: a circle (`numsides` and `angle` is ignored); deprecated.

`angle`:
the angle of rotation of the symbol

For backward compatibility, the form `(verts, 0)` is also accepted, but it is deprecated and equivalent to just `verts` for giving a raw set of vertices that define the shape.

`None` is the default which means 'nothing', however this table is referred to from other docs for the valid inputs from marker inputs and in those cases `None` still means 'default'.

Note that special symbols can be defined via the [STIX math font](#), e.g. "\$♠\$". For an overview over the STIX font symbols refer to the [STIX font table](#). Also see the [STIX Fonts Demo](#).

Integer numbers from 0 to 11 create lines and triangles. Those are equally accessible via capitalized variables, like `CARETDOWNBASE`. Hence the following are equivalent:

```
plt.plot([1,2,3], marker=11)
plt.plot([1,2,3], marker=matplotlib.markers.CARETDOWNBAS
```



Examples showing the use of markers:

- [Marker Reference](#)
- [Marker filling-styles](#)
- [Marker Path](#)

Classes

`MarkerStyle`([marker, fillstyle])

Parameters:

© Copyright 2002 - 2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012 - 2018 The Matplotlib development team.

Last updated on Jul 09, 2019. Created using [Sphinx](#) 1.8.5. Doc version v3.1.1-9-g823365338.