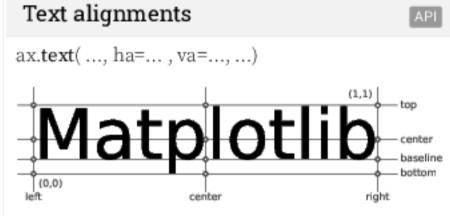


# origin="upper extent=[0,10,0,5] extent=[10,0,0,5] extent=[0,10,0,5] extent=[10,0,0,5]



left	center	right	
Text parar	neters		API
	mily=, size=, ntproperties=)	weight=)	
The quick b		xx-large	(1,73)
The quick bro		x-large	-
The quick brown fo		large medium	(1.20) (1.00)
The quick brown fox			(0.83)
The quick brown fox The quick brown fox		xx-small	

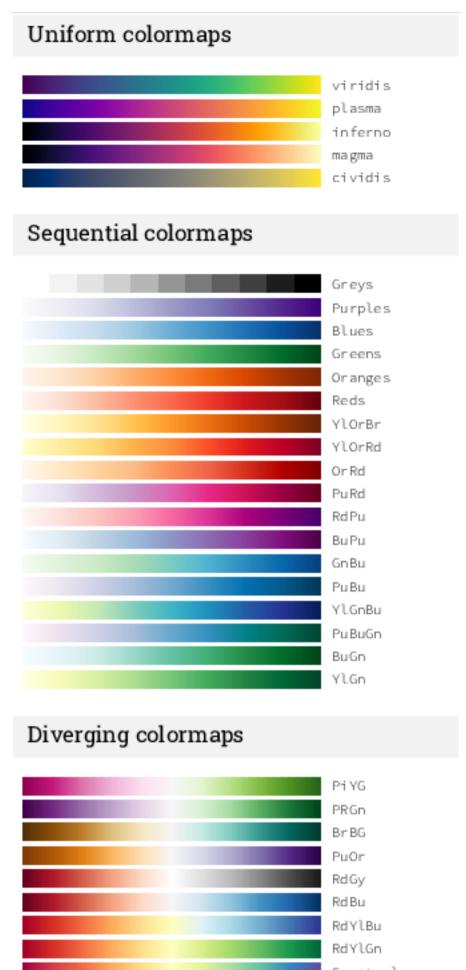
The quick brown fox jumps over the lazy dog	semibold (600)	
The quick brown fox jumps over the lazy dog	normal (400)	
The quick brown fox jumps over the lazy dog	ultralight (100)	
The quick brown fox jumps over the	lazy dog monospace	
The quick brown fox jumps over the lazy dog	serif	
The quick brown fox jumps over the lazy dog	sans	
The quick brown fox jumps over the lazy dog	cursive	
The quick brown fox jumps over the lazy dog	italic	
The quick brown fox jumps over the lazy dog	normal	
The quick brown fox jumps over the LAZY dog The quick brown fox jumps over the lazy dog	small-caps normal	

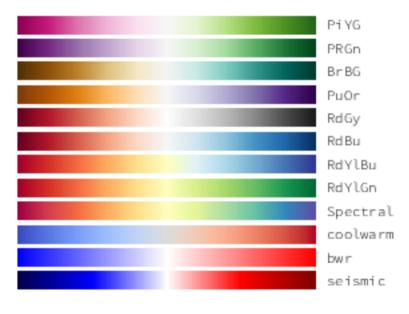
black (900)

bold (700)

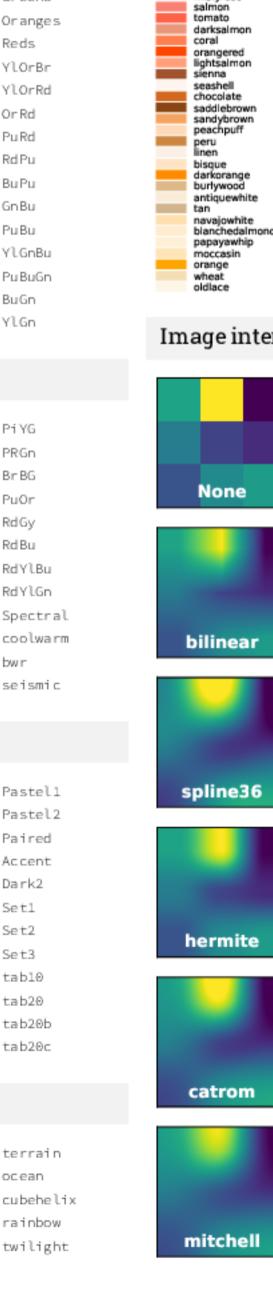
The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog









Color names

floralwhite darkgoldenrod goldenrod

gold lemonchiffon

palegoldenro darkkhaki

beige lightyellow

yellow olivedrab yellowgreen darkolivegreen

lawngreen honeydew

imegreen

darkgreen

seagreen mediumseagreen

mediumspringgreen

mediumaguamarine

springgreen

aguamarine

green lime

darkseagreer palegreen lightgreen forestgreer

olive

cornsilk

khaki

darkturquoise

powderblue

lightskyblue

lightslategray lightslateg

steelblue

aliceblue

slategray

rovalblue ghostwhite

avender

navy darkblue

midnightblue

mediumblue

siateblue darkslateblue

mediumslateblue

mediumpurple

mediumorchid

blueviolet

indigo darkorchid

darkviolet

thistle

plum violet

slategrey lightsteelblue

comflowerblue

cadetblue

skyblue

black k dimgray

dimgrey

gray grey darkgray darkgrey

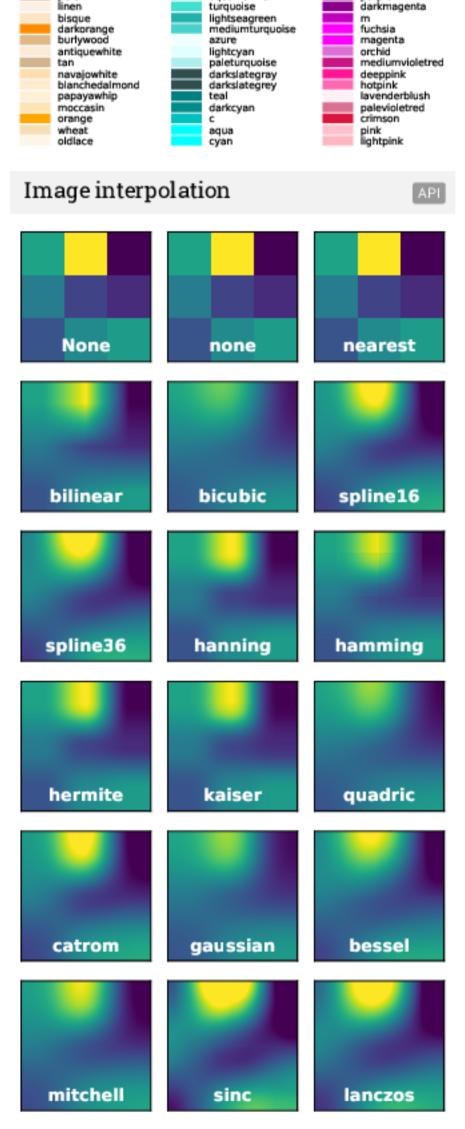
snow

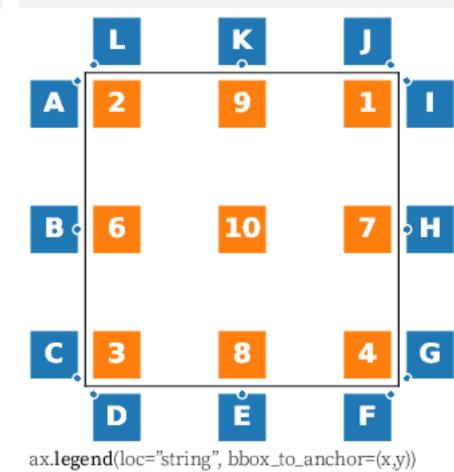
rosybrow lightcoral

indianred

firebrick

marcon





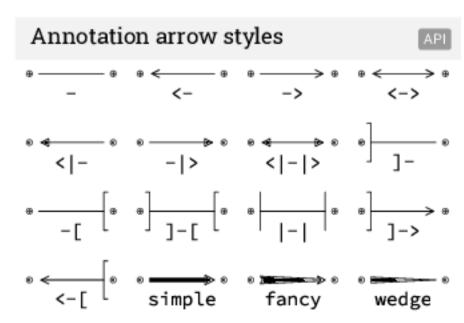
2: upper left 9: upper center 1: upper right 6: center left 10: center 7: center right lower left 8: lower center 4: lower right

A: upper right / (-0.1,0.9) C: lower right / (-0.1,0.1) E: upper center / (0.5, -0.1) G: lowerleft/(1.1,0.1) I: upper left / (1.1, 0.9) K: lower center / (0.5,1.1)

Legend placement

B: centerright / (-0.1,0.5) D: upper left / (0.1, -0.1) F: upper right / (0.9, -0.1) H: center left / (1.1,0.5) J: lower right / (0.9,1.1) L: lower left / (0.1,1.1)

## Annotation connection styles arc3, rad-0 arc3, rad-0.3 angle3, angleA-0, angleB-90 angle, angleA--90, angleB-180, rad-6 angle, angleA--98, angleB-188, rad-25 arc, angleA--90, angleB-0, armA-0, armB-40, rad-0 bar, fraction=0.3 bar, fraction--0.3 bar, angle=180, fraction==6.2



#### How do I ... ... resize a figure? → fig.set\_size\_inches(w, h) ... save a figure?

→ fig.savefig("figure.pdf") ... save a transparent figure?

→ fig.savefig("figure.pdf", transparent=True) ... clear a figure/an axes?

 $\rightarrow$  fig.clear()  $\rightarrow$  ax.clear() ... close all figures?

→ plt.close("all") ... remove ticks?

 $\rightarrow \text{ax.set}\_[\text{xy}] \text{ticks}([])$ ... remove tick labels?

→ ax.set\_[xy]ticklabels([])

... rotate tick labels?

→ ax.set\_[xy]ticks(rotation=90)

... hide top spine?

→ ax.spines['top'].set\_visible(False)

... hide legend border?

→ ax.legend(frameon=False)

... show error as shaded region?

→ ax.fill\_between(X, Y+error, Y-error) ... draw a rectangle?

→ ax.add\_patch(plt.Rectangle((0, 0), 1, 1)

... draw a vertical line?  $\rightarrow$  ax.axvline(x=0.5)

... draw outside frame?

→ ax.plot(..., clip\_on=False)

... use transparency?

 $\rightarrow$  ax.plot(..., alpha=0.25)

... convert an RGB image into a gray image?  $\rightarrow$  gray = 0.2989\*R + 0.5870\*G + 0.1140\*B

... set figure background color?

→ fig.patch.set\_facecolor("grey")

... get a reversed colormap?

→ plt.get\_cmap("viridis\_r")

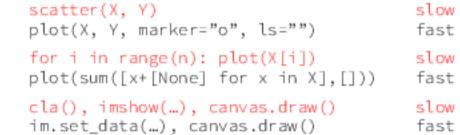
... get a discrete colormap?

→ plt.get\_cmap("viridis", 10)

... show a figure for one second?

→ fig.show(block=False), time.sleep(1)

### Performance tips



#### Beyond Matplotlib

Seaborn: Statistical Data Visualization Cartopy: Geospatial Data Processing vt: Volumetric data Visualization mpld3: Bringing Matplotlib to the browser Datashader: Large data processing pipeline plotnine: A Grammar of Graphics for Python

Matplotlib Cheatsheets Copyright (c) 2021 Matplotlib Development Team Released under a CC-BY 4.0 International License

