Networkx的作图参数：

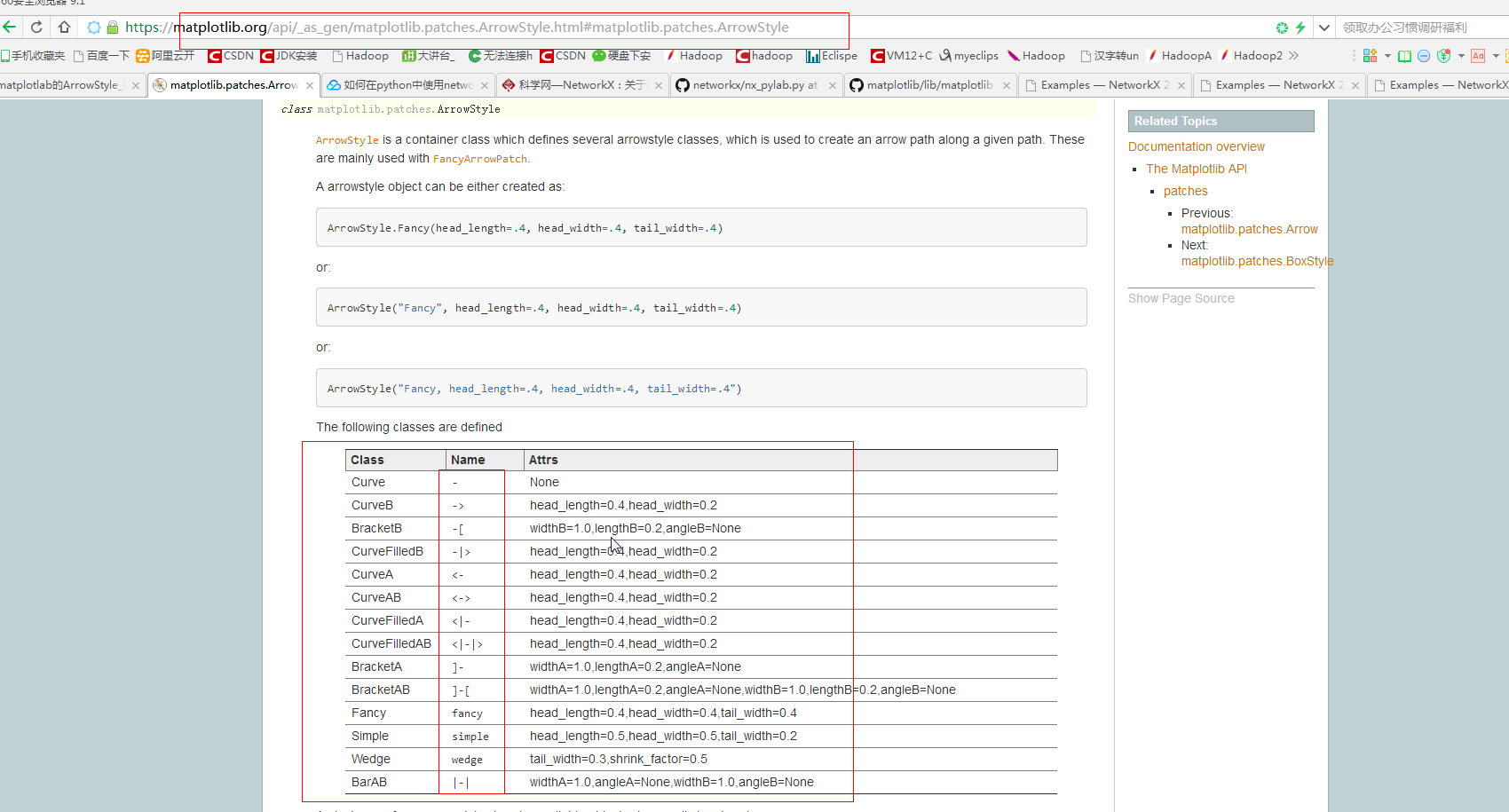
见源码：

https://github.com/networkx/networkx/blob/master/networkx/drawing/nx\_pylab.py

其中ArrowStyle参数，是基于Matplotlab的matplot.patches.Arrowstyle类，有固定的几种形状，如下：

地址，matplotlab官网：

https://matplotlib.org/api/\_as\_gen/matplotlib.patches.ArrowStyle.html#matplotlib.patches.ArrowStyle



Pos：

https://github.com/networkx/networkx/blob/master/networkx/drawing/layout.py

计算顶点位置的参数，默认使用spring\_layout方式，

**在**[**networkx**](https://github.com/networkx/networkx)/[networkx](https://github.com/networkx/networkx/tree/master/networkx)/[drawing](https://github.com/networkx/networkx/tree/master/networkx/drawing)/**layout.py里**

有：

random\_layout

circular\_layout

shell\_layout

bipartite\_layout

fruchterman\_reingold\_layout

kamada\_kawai\_layout

spectral\_layout

rescale\_layout

arrows:

布尔值——默认为true, 有向图会加箭头

arrowstyle——上面已经介绍了，应该是配合arrows使用

arrowsize——默认为10

见：matplotlib.patches.FancyArrowPatch类里的属性mutation\_scale

with\_labels——布尔值，默认true.

点上标识label

ax——matplotlib的axes类：

画轴线的（可能是别的图）

nodelist————一个python的list容器，存储了点的信息（默认是G.nodes（）），只画nodelist里的点，G中的其他点不画

edgelist——————一个python的list容器，存储边的列表（默认是G.edges()），也只画该list里的边

node\_size————一个scalar或者array?(可能是numpy里的)，默认长度为300， 点的大小，需要和nodelist的大小一致。

node\_color————节点的颜色，值可为：str类型，用颜色表示，如‘r’表示红色，默认为红色；

也可为：float类型，会将float转成RGB通道的颜色

另：值除了是单个str或者float

也可以是:Sequence[str]或者Sequence[float], 比如一个颜色列表的list，numpy.array，应该是复合Sequence接口的都行

注意：node\_color的值为一个Sequence序列时，长度要和node\_list一致。（表示点的着色）

node\_shape————节点形状，值可为：str

是matplotlab中的scatter类，其提供如下几种：

'so^>v<dph8'中的一种

alpha————点或边的“透明度”，默认值为1.0， 浮点型

cmap————matplotlab的colormap， 默认为None,颜色映射给节点

vmin,vmax————默认为None,对node的color map的scaling的最大值最小值，不懂！？？？？

linewidths————symbol border的宽——3种值：（None,scalar,Sequence）

widths————边的宽度，默认1.0

edge\_color——————边的宽度，值和node\_color一样：

单个边，值： 格式化的str，如‘r’表示red；或者float值，会被转化成颜色

多个边，值：一个Sequence序列的str或者float, 必须：和edge\_list长度一致

edge\_cmap——类似于cmap

edge\_vmin,edge\_vmax——类似于vmin,vmax

style——————————线的风格，默认为“solid”,实心，edge线的几种如下：solid、dashed、dotted、dashdot

labels————————dic类型，节点的标签， nodes的标签（应该就是别名）——会显示在图上

默认为None

font\_size——————————节点标签的文本字体大小，默认12

font\_color————————str类型，默认为“k” ，上面labels的字体颜色

font\_weight————————字体的权重。。。。默认为“normal”

font\_family———字体家族，默认为“sans-serif”

label——————不同于labels，这个是图例legend的标签

另外，networkx作图的部分，在这里面：

<https://github.com/networkx/networkx/blob/master/networkx/drawing/nx_pylab.py>

也即：networkx.drawing包里面

上面只是说了图的一些属性、构建。

具体作图方法：有几类：

画点、画边、 画点的labels、画边的labels、画整张图的lagend之类

具体如下：

1. 画点：

def draw\_networkx\_nodes(G, pos,

nodelist=None,

node\_size=300,

node\_color='r',

node\_shape='o',

alpha=1.0,

cmap=None,

vmin=None,

vmax=None,

ax=None,

linewidths=None,

edgecolors=None,

label=None,

\*\*kwds):

1. 画边：

def draw\_networkx\_edges(G, pos,

edgelist=None,

width=1.0,

edge\_color='k',

style='solid',

alpha=1.0,

arrowstyle='-|>',

arrowsize=10,

edge\_cmap=None,

edge\_vmin=None,

edge\_vmax=None,

ax=None,

arrows=True,

label=None,

node\_size=300,

nodelist=None,

node\_shape="o",

\*\*kwds):

1. 画点的labels:

def draw\_networkx\_labels(G, pos,

labels=None,

font\_size=12,

font\_color='k',

font\_family='sans-serif',

font\_weight='normal',

alpha=1.0,

bbox=None,

ax=None,

\*\*kwds):

4.画边的labels:

def draw\_networkx\_edge\_labels(G, pos,

edge\_labels=None,

label\_pos=0.5,

font\_size=10,

font\_color='k',

font\_family='sans-serif',

font\_weight='normal',

alpha=1.0,

bbox=None,

ax=None,

rotate=True,

\*\*kwds):

5.画各种风格的图（各种style）

如下几种：

def draw\_circular(G, \*\*kwargs):

def draw\_kamada\_kawai(G, \*\*kwargs):

def draw\_random(G, \*\*kwargs):

def draw\_spectral(G, \*\*kwargs):

def draw\_spring(G, \*\*kwargs):

def draw\_shell(G, \*\*kwargs):

6．设置透明度：

def apply\_alpha(colors, alpha, elem\_list, cmap=None, vmin=None, vmax=None):