**NOTE: This is an official document by Indexademics. Unless otherwise stated, this document may not be accredited to individuals or groups other than the club IDX, nor should this document be distributed, sold, or modified for personal use in any way.**

**IDX G9 Computer Science STUDY GUIDE ISSUE 6**

**By Iris Lan and Gavin Yu**

**Matplotlib**

*Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.*

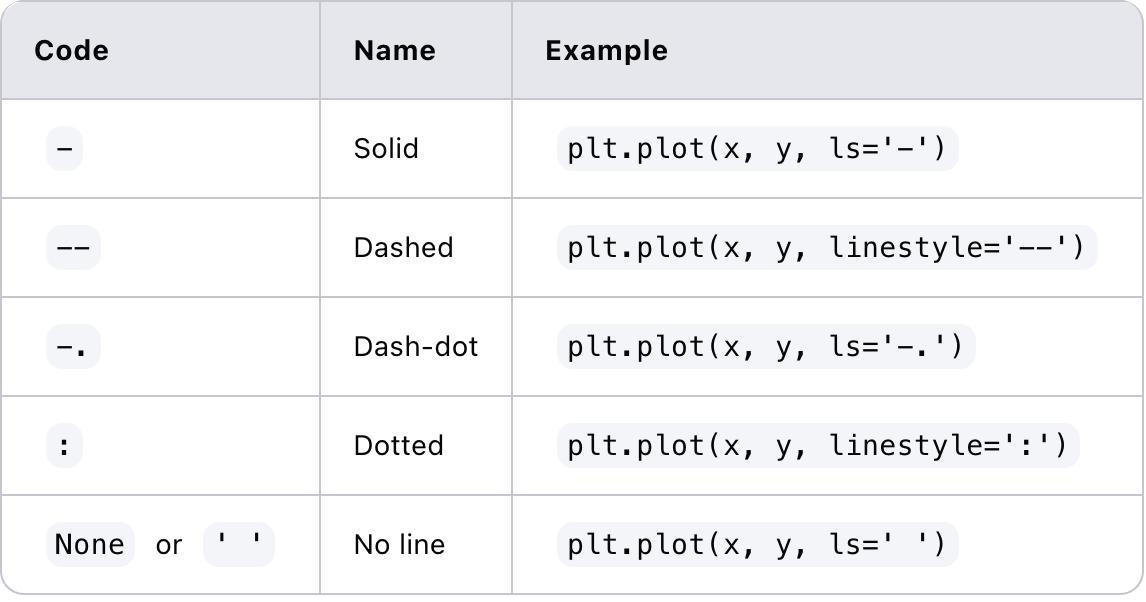
**Installation**

* Install - python –m pip install matplotlib
* import - import matplotlib.pyplot as plt

**Basic Grammar**

fig = plt.figure() 🡪 creating an empty figure

**Plotting different graphs**

* Line plot – plt.plot(x, y, color = ‘’, linestyle = ‘’, mfc = ‘’, marker = ‘’, label = ‘’)
  + x, y – the values of x and y axes in lists
  + color – color of the line or bar or sections
  + label – label the name of the line
  + linestyle – the style of the line in the line plot, default setting is solid
  + mfc – color of the dot
  + marker – shape used to display the points in the line plot
* Scatter plot – plt.scatter(x, y, color = ‘’, marker = ‘’)
* Bar chart – plt.bar(x, y, color = ‘’, width = )
  + Width – width of the bar
* Horizontal Bar chart – plt.barh(x, y, color = ‘’, width = )
* Multiple bar chart
  + Can use Series to customize the location of bars quickly
    - x = pd.Series(list(range(1,len(df)+1)))

x1, x2 = x - w/2, x + w/2 # 2 bars

x1, x2, x3 = x - w, x, x + w # 3 bars

* Pie chart – plt.pie(values, color = ‘’, labels = [], autopct = ‘’)
  + Autopct = ‘%.1f%%’
  + Labels – label each section of the pie chart
  + Starangle is used to set the starting angle of the first slice
  + Explode represents the fraction of the radius of a specific slice
* Subplots – plt.subplot(rows, cols, index)
  + GridSpec customizes the grid structure

**Examples**

**1. Line Plot**

plt.plot([1, 2, 3, 4], [1, 4, 9, 16])

**2. Scatter Plot**

plt.scatter([1, 2, 3, 4], [1, 4, 9, 16])

**3. Bar Plot**

plt.bar(['A', 'B', 'C', 'D'], [20, 35, 30, 25])

**4. Horizontal Bar Plot**

plt.barh(['A', 'B', 'C', 'D'], [20, 35, 30, 25])

**5. Histogram**

plt.hist(np.random.randn(1000), bins = 20)

**6. Pie Chart**

plt.pie([15, 30, 45, 10], labels=[A, B, C, D])

**rcParams**

* plt.rcParams['font.family'] = ‘Times New Roman’
* plt.rcParams['font.size'] = 14
* plt.rcParams['lines.linewidth'] = 2

**Labels**

* plt.legend(loc = ‘position’)
  + loc = 'best', 'upper right', 'upper left', 'lower left', 'lower right', 'right', 'center left', 'center right', 'lower center', 'upper center', 'center'
* plt.xlabel(‘name of x axis’), plt.ylabel(‘name of y axis’)
* plt.title(‘name of the graph’)

**Axis Customization**

* plt.xticks(), plt.yticks(),
* plt.xlim(), and plt.ylim()
* plt.grid(axis = ‘name of axis’, linestyle = ‘linestyle’)
  + axis = ‘x’ or ‘y’

**Others**

* customize size of the plot
  + plt.figure(figsize=(width, height))
* plt.twinx()
  + directly after using this function, write the code for the other twin graph

**Zip() function**

* combines the data series, colors, and labels into tuple
* for a, b in zip(...) iterates those tuples
* Example:
  + list1 = [1, 2, 3]
  + list2 = ['a', 'b', 'c']
  + zipped = zip(list1, list2)
* Output: [(1, 'a'), (2, 'b'), (3, 'c')]