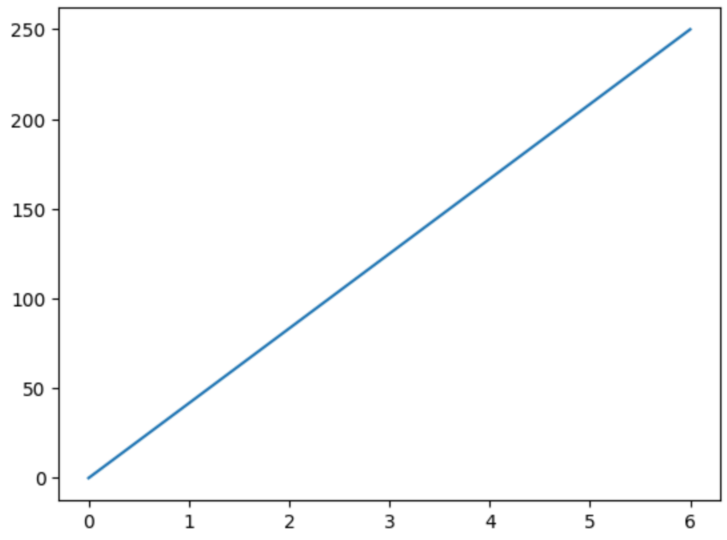
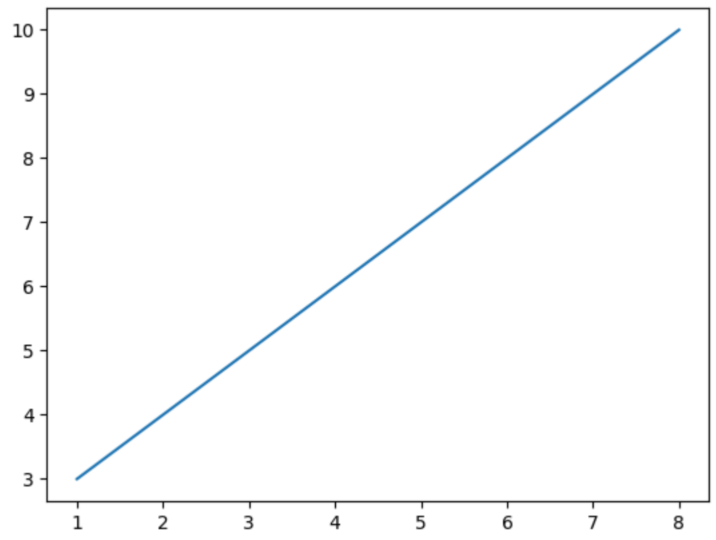
Overview of Data **Visuaization**

1. Draw a line in a diagram from position(0,0) to position(6,250):

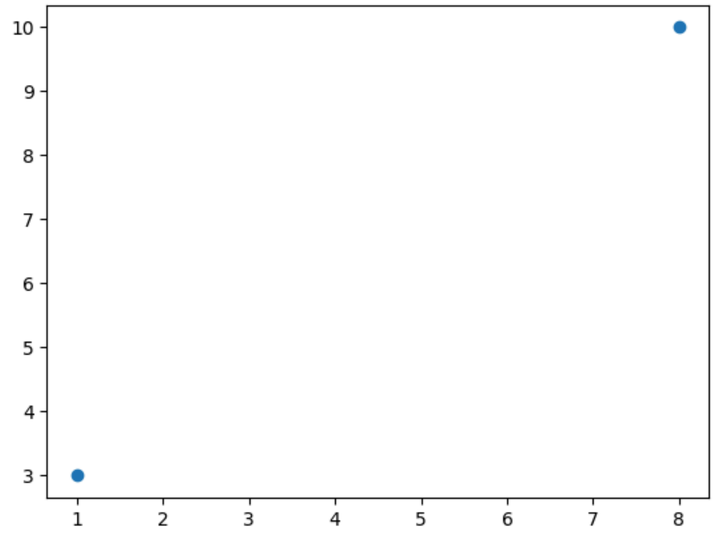
```
import matplotlib.pyplot as plt
import numpy as np
xpoints=np.array([0,6])
ypoints=np.array([0,250])
plt.plot(xpoints,ypoints)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
xpoints=np.array([1,8])
ypoints=np.array([3,10])
plt.plot(xpoints,ypoints)
plt.show()
```



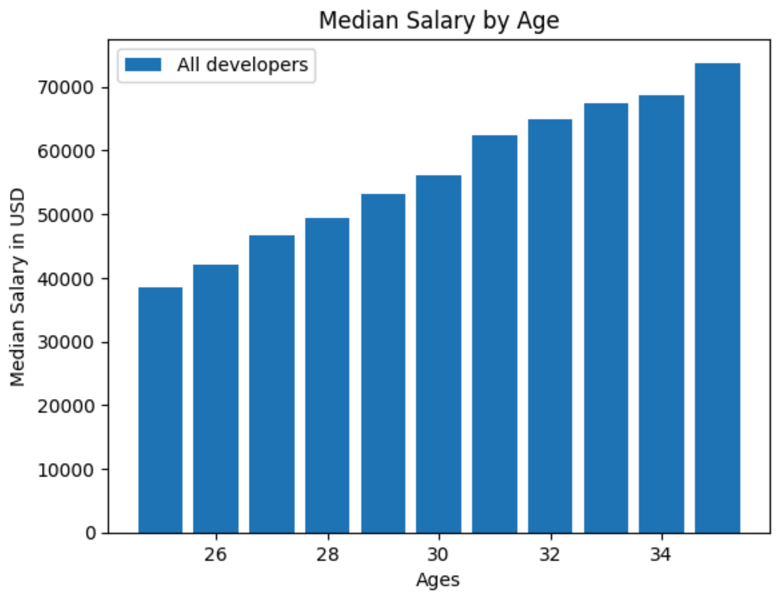
```
import matplotlib.pyplot as plt
import numpy as np
xpoints=np.array([1,8])
ypoints=np.array([3,10])
plt.plot(xpoints,ypoints,'o')
plt.show()
```



```
x = [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
```

```
devs_y = [38496, 42000, 46752, 49320, 53200, 56000, 62316, 64928, 67317, 68748, 73752]
```

```
plt.bar(x, devs_y, label="All developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```

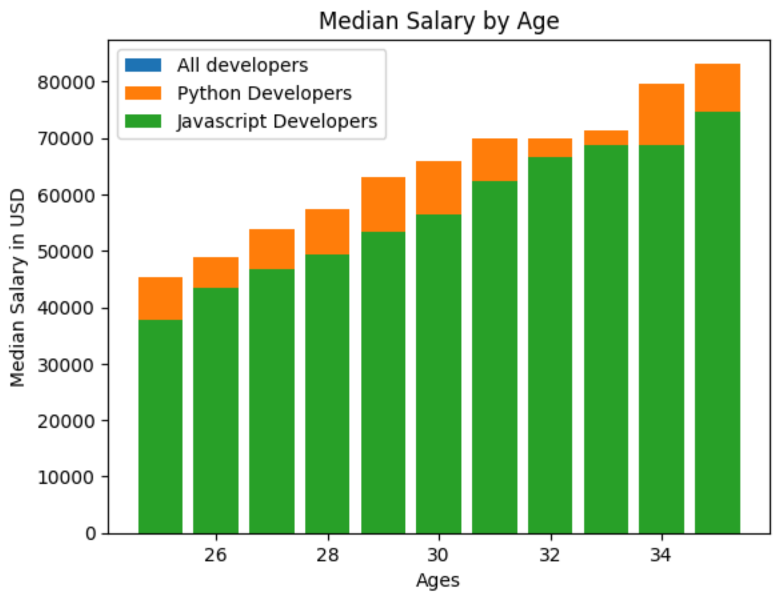


2. Adding more bars to the same plot

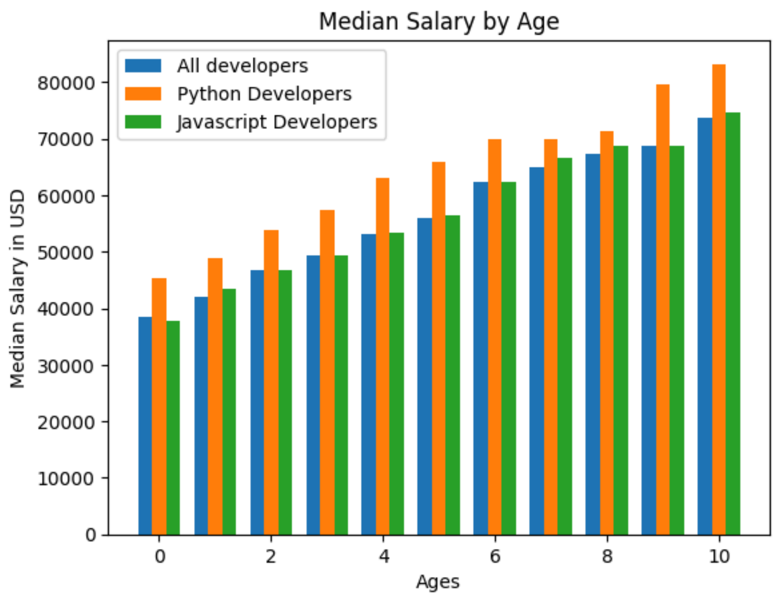
```
py_devs_y = [45372, 48876, 53850, 57287, 63016, 65998, 70003, 70000, 71418, 79674, 83238]
```

```
js_devs_y = [37810, 43515, 46823, 49293, 53437, 56373, 62375, 66674, 68745, 68746, 74583]
```

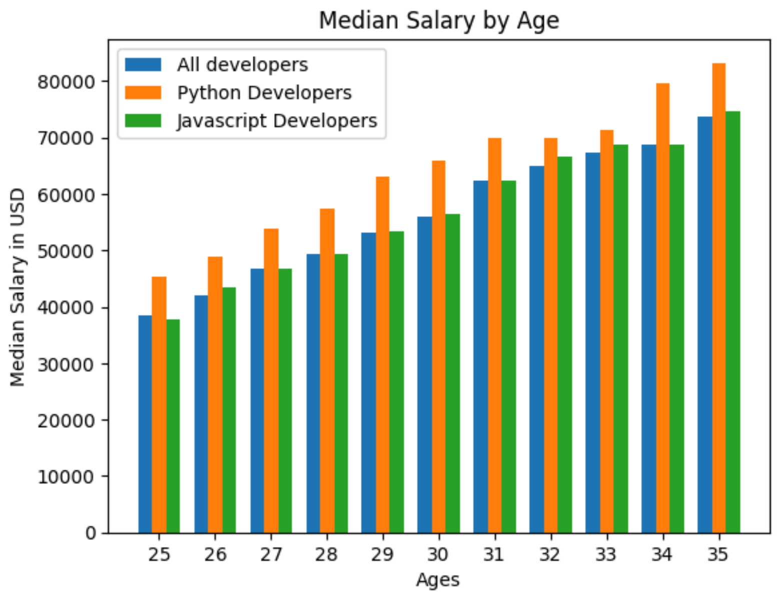
```
plt.bar(x, devs_y, label="All developers")
plt.bar(x, py_devs_y, label="Python Developers")
plt.bar(x, js_devs_y, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```



```
import numpy as np

x_indexes = np.arange(len(x))


width = 0.25


plt.bar(x_indexes - width, devs_y, width=width, label="All developers")
plt.bar(x_indexes, py_devs_y, width = width, label="Python Developers")
plt.bar(x_indexes + width, js_devs_y, width=width, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.legend()
plt.show()
```



```
plt.bar(x_indexes - width, devs_y, width=width, label="All developers")
plt.bar(x_indexes, py_devs_y, width = width, label="Python Developers")
plt.bar(x_indexes + width, js_devs_y, width=width, label="Javascript Developers")
plt.xlabel("Ages")
plt.ylabel("Median Salary in USD")
plt.title("Median Salary by Age")
plt.xticks(ticks=x_indexes, labels=x) #changing the xlabel
plt.legend()
plt.show()
```



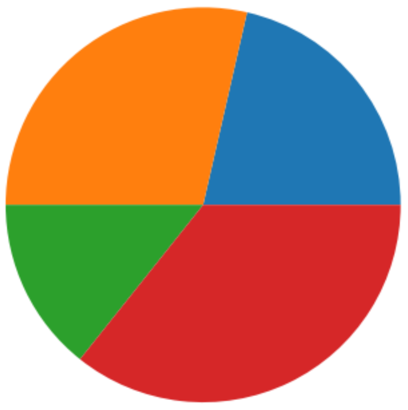
```
from matplotlib import pyplot as plt
```

Recommended to use when we have 5 or less values to plot

1. Plotting the Pie Chart

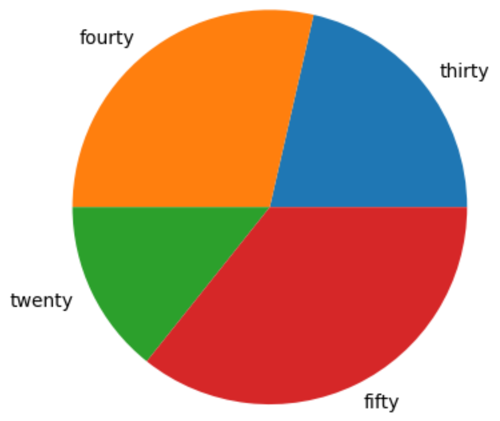
```
slices = [30, 40, 20, 50] #sum needs not be 100
```

```
plt.pie(slices)
plt.show()
```



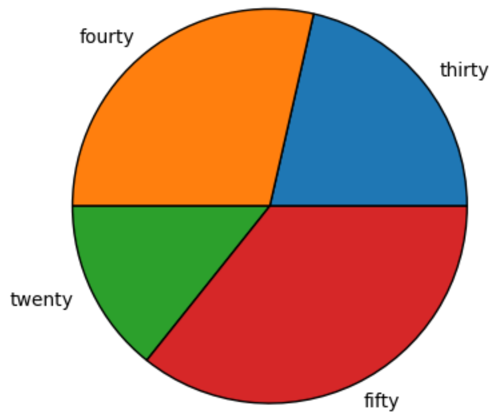
2. Adding labels to the pie chart

```
labels = ['thirty','fourty', 'twenty','fifty']
plt.pie(slices, labels=labels)
plt.show()
```



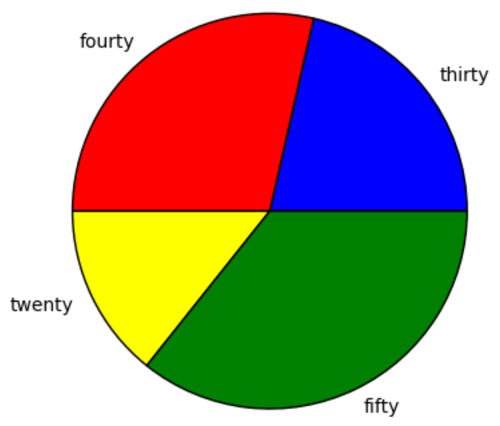
3. setting edge color

```
plt.pie(slices, labels=labels, wedgeprops={'edgecolor':'black'})
plt.show()
```



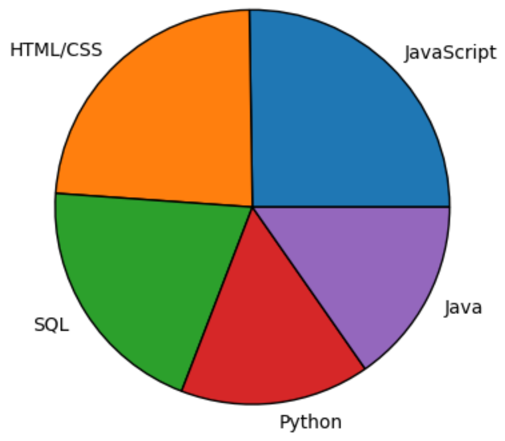
4. setting color of the slices

```
color = ['blue','red','yellow','green']

#hexadecimal color codes can also be used
plt.pie(slices, labels=labels, colors=color, wedgeprops={'edgecolor':'black'})
plt.show()
```



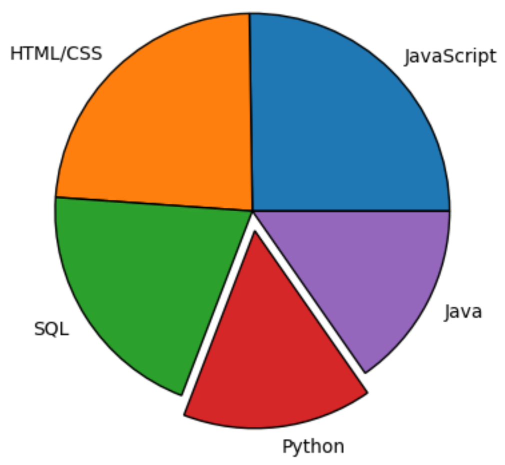
5. plotting real world data

```
labels = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']

slices = [59219, 55466, 47544, 36443, 35917]


plt.pie(slices, labels=labels, wedgeprops={'edgecolor':'black'})
plt.show()
```



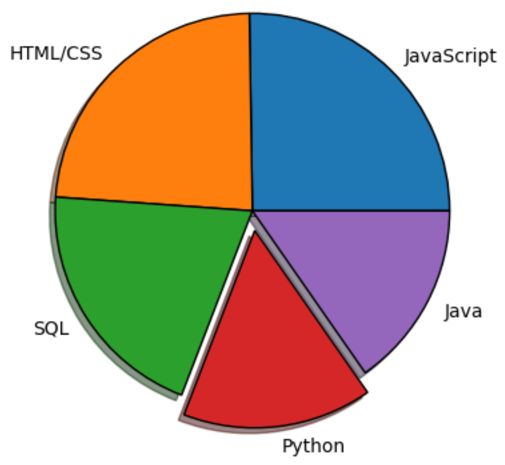
6. Exploding the slice

```
explode = [0, 0, 0, 0.1, 0]

plt.pie(slices, labels=labels, explode=explode, wedgeprops={'edgecolor':'black'})
plt.show()
```



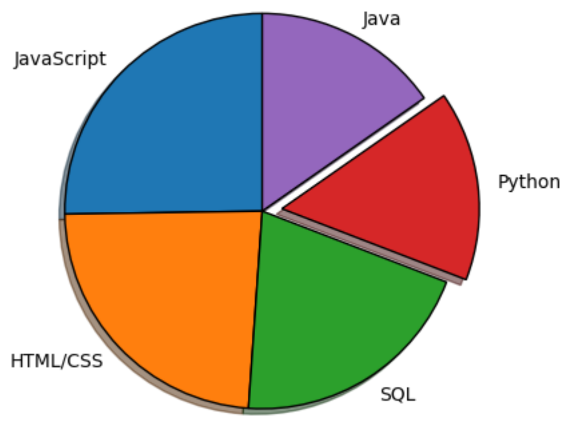
7. adding shadow to the chart

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, wedgeprops={'edgecolor':'black'})
plt.show()
```



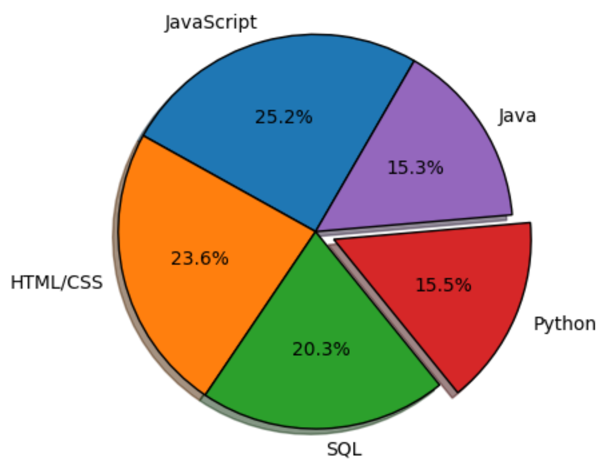
8. setting the starting angle

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, startangle=90, wedgeprops={'edgecolor':'black'})
plt.show()
```



9. displaying percentage of each slices

```
plt.pie(slices, labels=labels, explode=explode, shadow=True, startangle=60, autopct="%0.1f%%", wedgeprops={'edgecolor':'black'})
plt.show()
```



Start coding or generate with AI.

Data Visualization using Matplotlib

1. Line Plots
2. Bar Charts
3. Pie Charts
4. Stack Plots
5. Histograms
6. Scatter Plots
7. Subplots

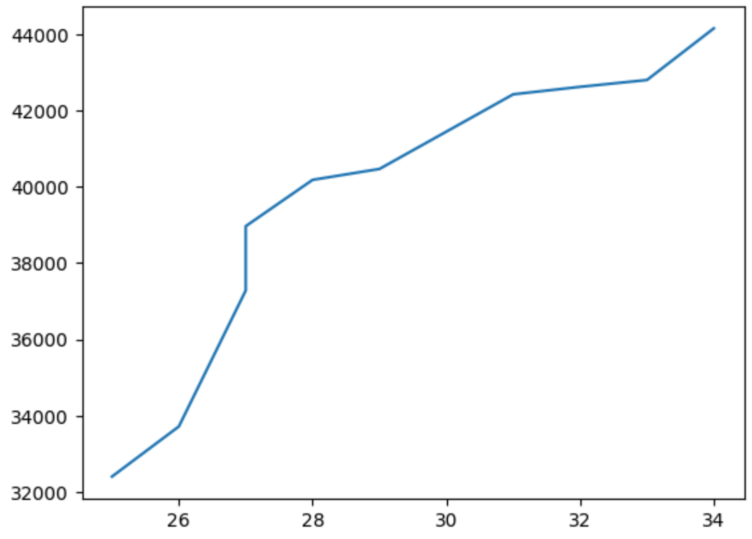
1. Creating Plots

```
import matplotlib.pyplot as plt
import random
```

```
# generating 10 random numbers between 25 to 35
ages = [random.randrange(25,35,1) for ages in range(11)]
ages = sorted(ages, reverse=False)


# generating 10 random numbers between 30k to 45k

devs = [random.randrange(30000,45000,1) for devs in range(11)]
devs = sorted(devs, reverse=False)
```

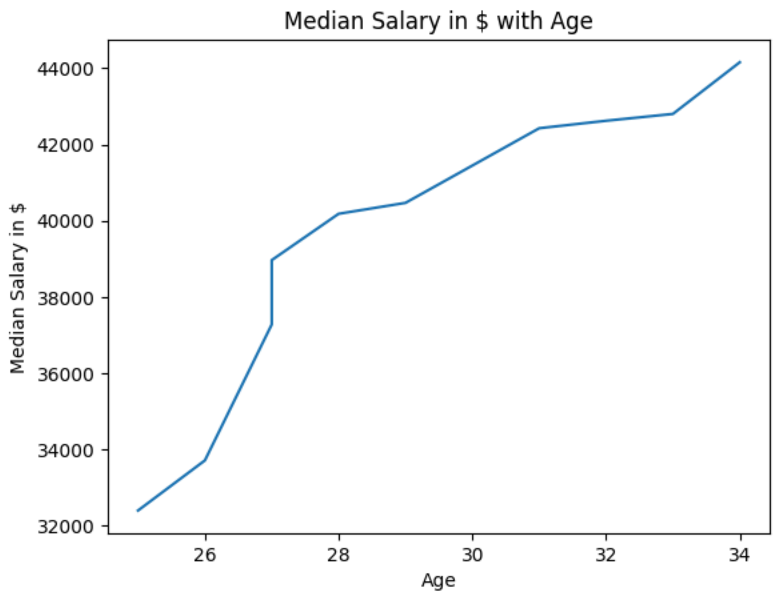
1.1. Plotting Line Plot

```
plt.plot(ages, devs)
plt.show()
```



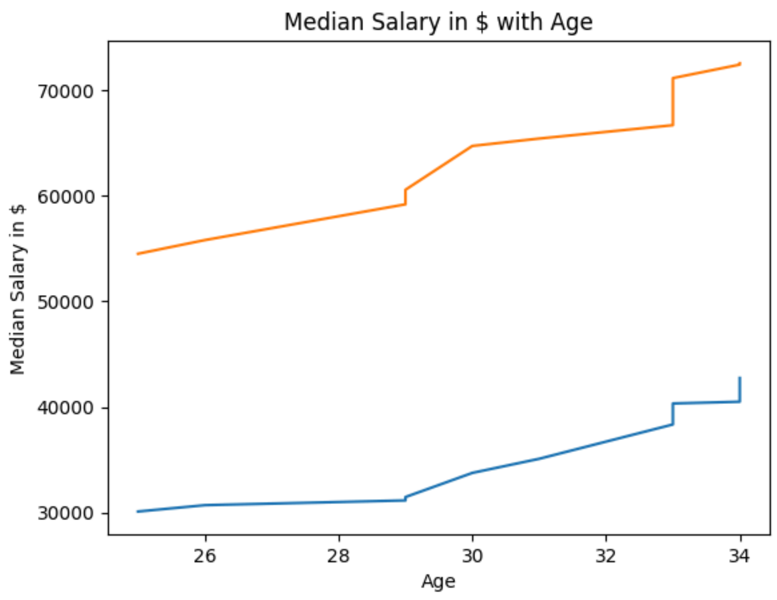
1.2. Adding title, xlabel and ylabel

```
plt.plot(ages, devs)
plt.title("Median Salary in $ with Age") # add the title
plt.xlabel("Age") # add xlabel
plt.ylabel("Median Salary in $") #add ylabel
plt.show()
```



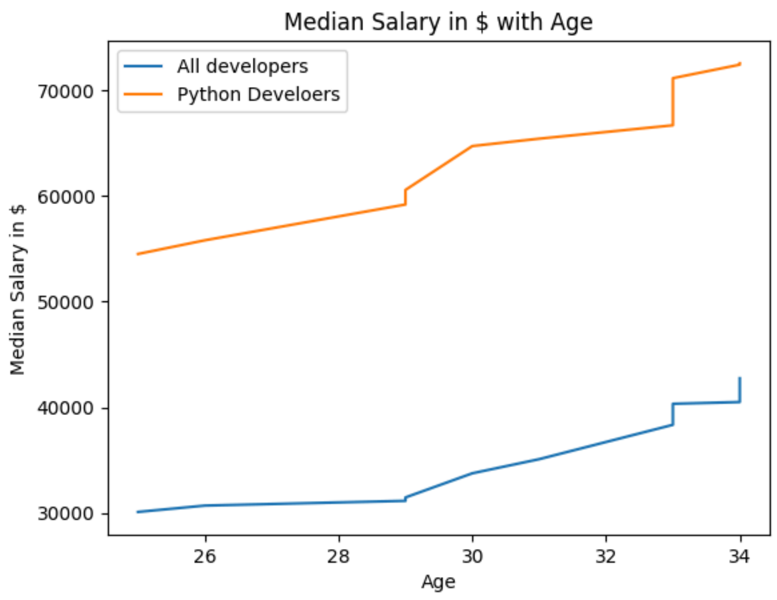
1.3. Adding more plot to the same graph

```
#creating 10 random numbers between 50k to 75k
import random
import matplotlib.pyplot as plt
ages = [random.randrange(25,35,1) for ages in range(11)]
ages = sorted(ages, reverse=False)
devs = [random.randrange(30000,45000,1) for devs in range(11)]
devs = sorted(devs, reverse=False)
py_devs = [random.randrange(50000,75000) for py_devs in range(11)]
py_devs = sorted(py_devs, reverse=False)


plt.plot(ages, devs)
plt.plot(ages, py_devs) # adding other plot to the same figure
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.show()
```



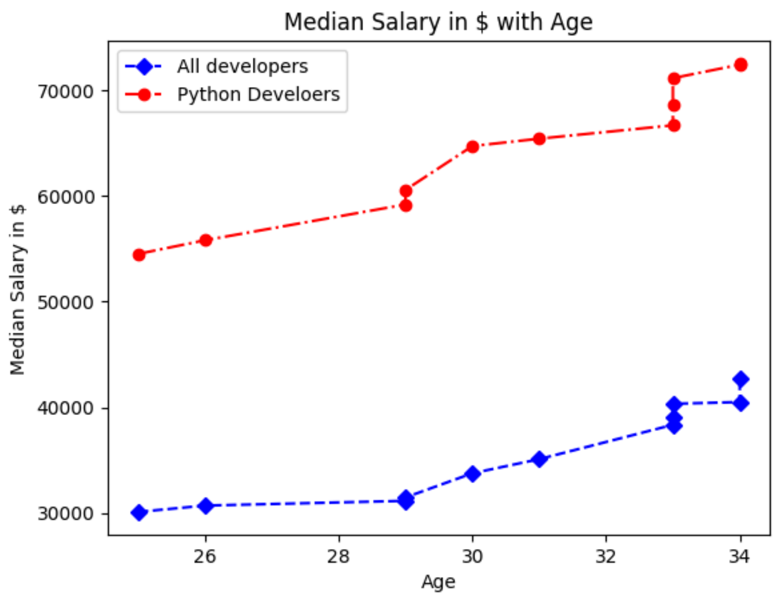
1.4. Adding legend to the plot

```
plt.plot(ages, devs, label = "All developers") # label
plt.plot(ages, py_devs, label = "Python Develoers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend() #plot the legend
plt.show()
```



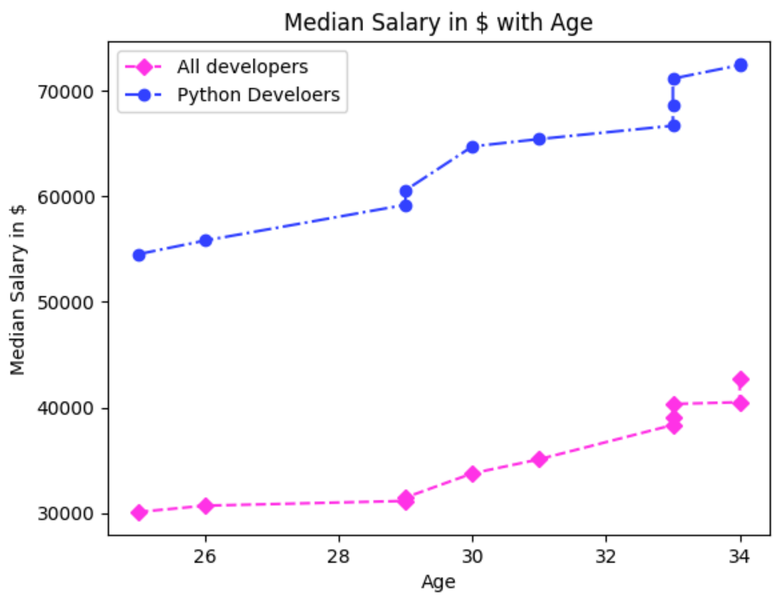
1.5. Setting marker, linestyle and color

```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="blue", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="red", linestyle = "-.", marker = "o", label = "Python Develoers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend()
plt.show()
```



1.6. Hexadecimal code for colors

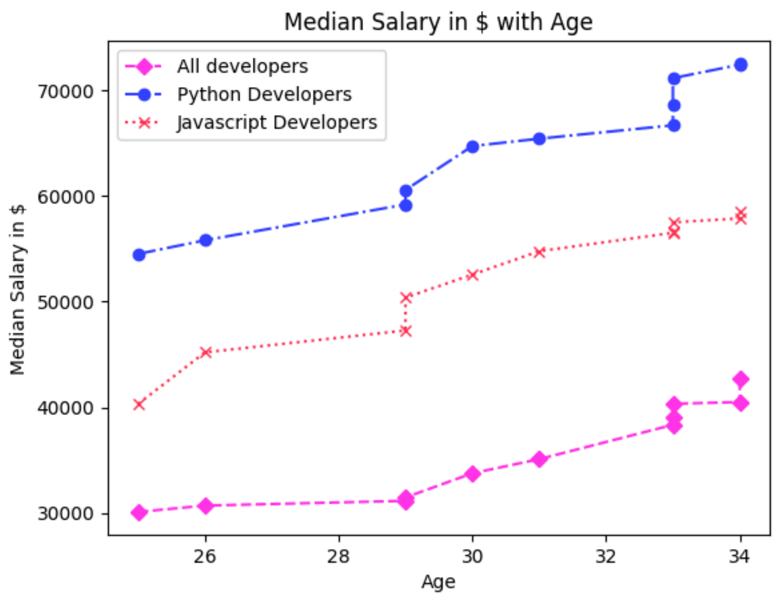
```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="#FF33E9", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="#3344FF", linestyle = "-.", marker = "o", label = "Python Develoers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend()
plt.show()
```



Adding other plot to the same graph

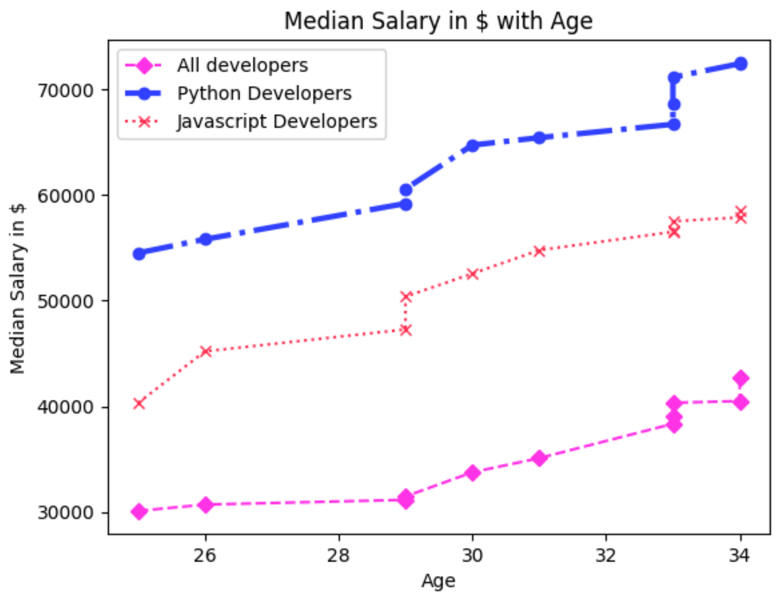
```
#creating 10 random numbers between 40k to 60k

js_devs = [random.randrange(40000,60000) for js_devs in range(11)]
js_devs = sorted(js_devs, reverse=False)
```

```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="#FF33E9", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="#3344FF", linestyle = "-.", marker = "o", label = "Python Developers")
plt.plot(ages, js_devs, color="#FF3355", linestyle = ":", marker = "x", label = "Javascript Developers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend()
plt.show()
```



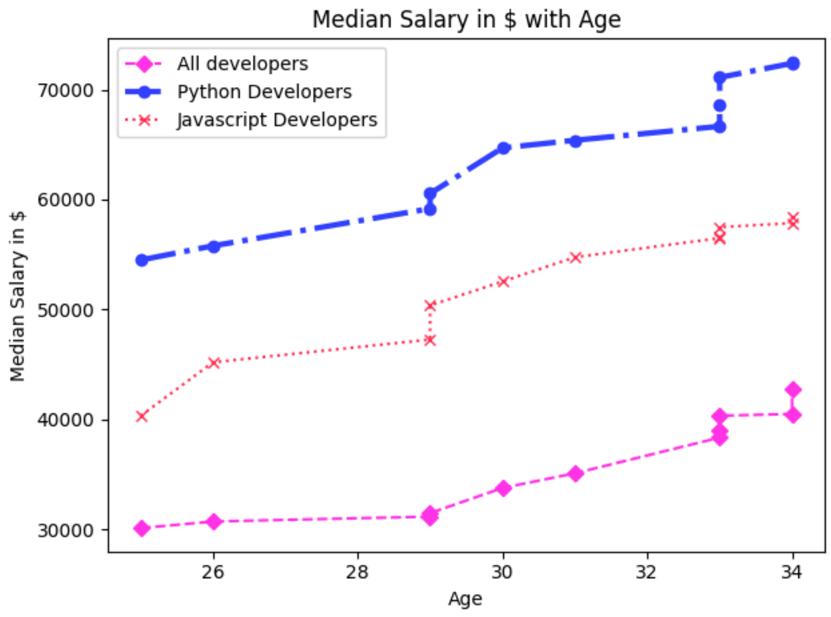
1.7. Changing the line width

```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="#FF33E9", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="#3344FF", linestyle = "-.", marker = "o", linewidth=3, label = "Python Developers")
plt.plot(ages, js_devs, color="#FF3355", linestyle = ":", marker = "x", label = "Javascript Developers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend()
plt.show()
```



1.8. Add padding to the plot

```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="#FF33E9", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="#3344FF", linestyle = "-.", marker = "o", linewidth=3, label = "Python Developers")
plt.plot(ages, js_devs, color="#FF3355", linestyle = ":", marker = "x", label = "Javascript Developers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.legend()
plt.tight_layout() #adds padding
plt.show()
```



1.9. Adding grid to the plot

```
#https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

plt.plot(ages, devs, color="#FF33E9", linestyle = "--", marker = "D", label = "All developers")
plt.plot(ages, py_devs, color="#3344FF", linestyle = "-.", marker = "o", linewidth=3, label = "Python Developers")
plt.plot(ages, js_devs, color="#FF3355", linestyle = ":", marker = "x", label = "Javascript Developers")
plt.title("Median Salary in $ with Age")
plt.xlabel("Age")
plt.ylabel("Median Salary in $")
plt.grid(True)
plt.legend()
plt.show()
```

Median Salary in $ with Age

1.10. Changing style of the plot