basics

resources used

- matplotlib tutorial 2020

In [2]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

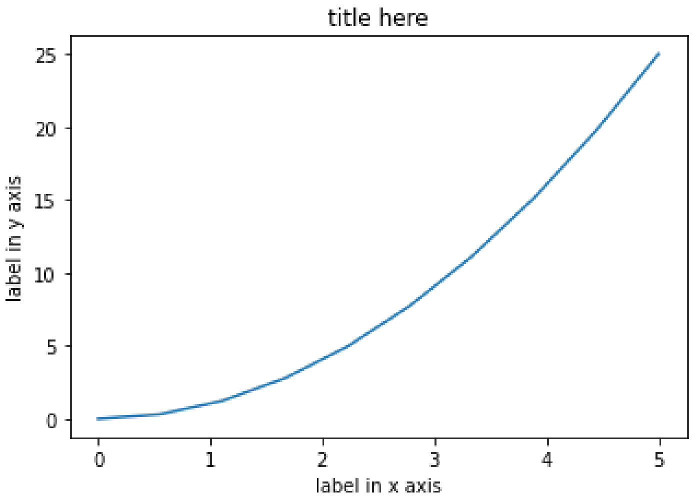
simple plots

In [3]:

```
# simple plot

x_1 = np.linspace(0,5,10) # generate 10 float from 0 to 5
y_1 = x_1**2

plt.plot(x_1, y_1)
plt.title("title here")
plt.xlabel("label in x axis")
plt.ylabel("label in y axis")
```

Out[3]:

```
Text(0, 0.5, 'label in y axis')
```

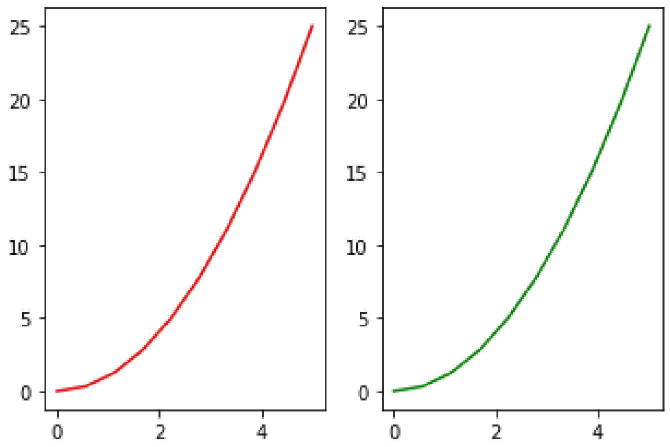


In [4]:

```
# multiple plot

# means I want to plot (row, col, pos)
# row (col) num of plots in row (col)
# pos of the next plot defined like
# the number in a smarthphone
'''
row: 1
col: 2
pos:
|1|2|
'''
plt.subplot(1,2,1)
plt.plot(x_1, y_1, 'r') # r is color red
plt.subplot(1,2,2)
plt.plot(x_1, y_1, 'g') # g is color green
```

Out[4]:

```
[<matplotlib.lines.Line2D at 0x7fbfc7ce77d0>]
```



In [5]:

```
'''
row: 2
col: 2
pos:
|1|2|
|3|4|
'''
plt.subplot(2,2,1)
plt.plot(x_1, y_1, 'r') # r is color red
plt.subplot(2,2,2)
plt.plot(x_1, y_1, 'g') # g is color green
plt.subplot(2,2,3)
plt.plot(x_1, y_1, 'b') # r is color red
plt.subplot(2,2,4)
plt.plot(x_1, y_1, 'y') # g is color green

# subplot can handle max 4 images
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x7fbfc7baa450>]
```

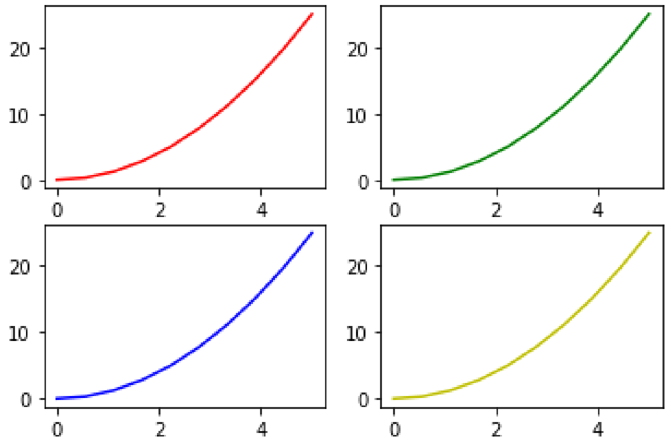


Figure Object

is an object that contains plot elements, can contains multiple axis

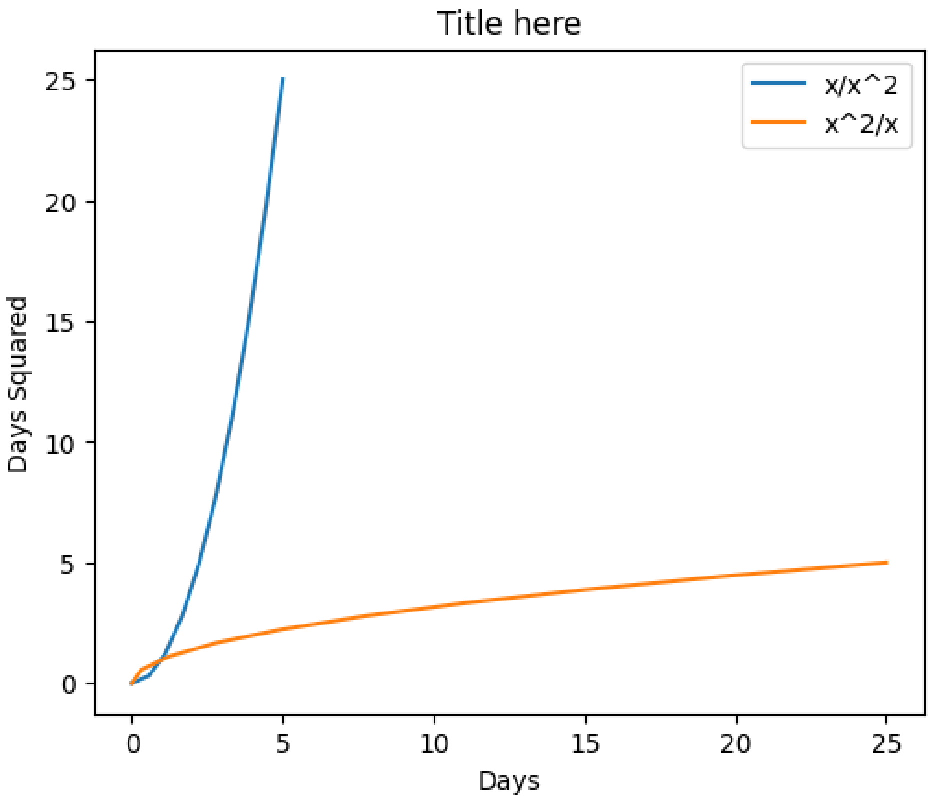
In [6]:

```
fig_1 = plt.figure(figsize=(5, 4), dpi=100)
axes_1 = fig_1.add_axes([0.1, 0.1, 0.9, 0.9])
```

```
axes_1.set_xlabel('Days')
axes_1.set_ylabel('Days Squared')
axes_1.set_title('Title here')

# apply 2 different plots
axes_1.plot(x_1, y_1, label='x/x^2')
axes_1.plot(y_1, x_1, label='x^2/x')

# legend position:
# - 0: plt figure out where to put it
# - 1: upper right
# - 2: upper left
# - 3: lower left
# - 4: lower right
# - manual (x, y) from lower left

axes_1.legend(loc=1)
```

Out[6]:

```
<matplotlib.legend.Legend at 0x7fbfc7b4a350>
```

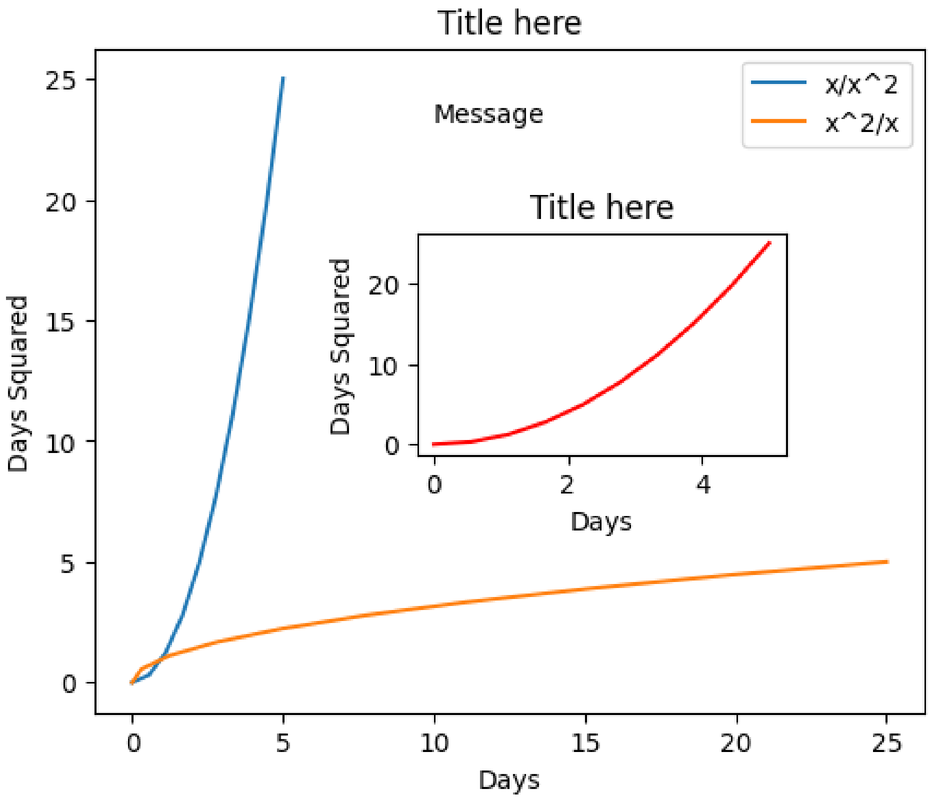


In [11]:

```
# plot inside a plot

fig_1 = plt.figure(figsize=(5, 4), dpi=100)
axes_1 = fig_1.add_axes([0.1, 0.1, 0.9, 0.9])


axes_1.set_xlabel('Days')
axes_1.set_ylabel('Days Squared')
axes_1.set_title('Title here')

# apply 2 different plots
axes_1.plot(x_1, y_1, label='x/x^2')
axes_1.plot(y_1, x_1, label='x^2/x')

# legend position:
# - 0: plt figure out where to put it
# - 1: upper right
# - 2: upper left
# - 3: lower left
# - 4: lower right
# - manual (x, y) from lower left

axes_1.legend(loc=1)

# second plot

axes_2 = fig_1.add_axes([0.45, 0.45, 0.4, 0.3])
axes_2.set_xlabel('Days')
axes_2.set_ylabel('Days Squared')
axes_2.set_title('Title here')
axes_2.plot(x_1, y_1, 'r')

# add message inside the plot
axes_2.text(0, 40, 'Message')
```

Out[11]:

```
Text(0, 40, 'Message')
```



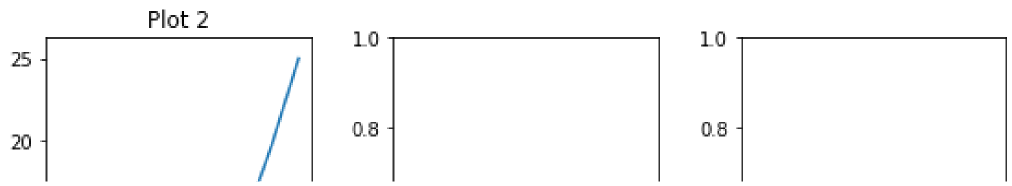
Sub Plots

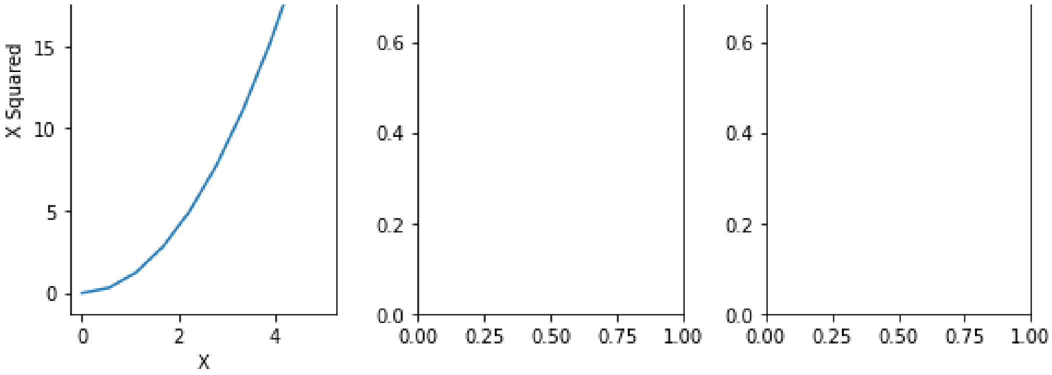
In [17]:

```
fig_2, axes_2 = plt.subplots(figsize=(8,4), nrows=1, ncols=3)
plt.tight_layout() # fixes overlapping between subplots
# we can access plots by their indexes
# in the example: |0|1|2|

axes_2[0].set_title('Plot 2')
axes_2[0].set_xlabel('X')
axes_2[0].set_ylabel('X Squared')
axes_2[0].plot(x_1, y_1)
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x7fbfbacb4650>]
```



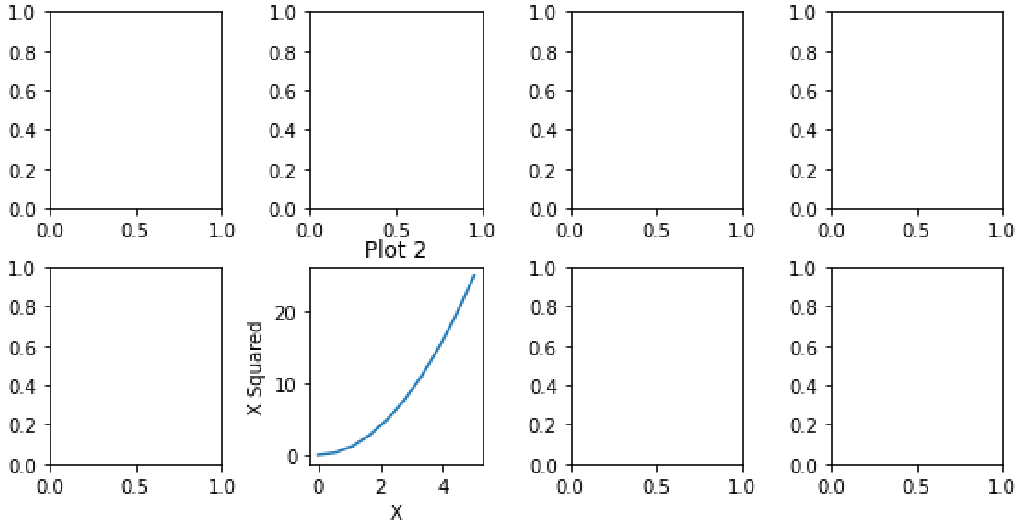


In [21]:

```
# if we have a matrix of plots we can
# tweak i, j to change position
rows = 2
cols = 4

fig_2, axes_2 = plt.subplots(figsize=(8,4), nrows=rows, ncols=cols)
plt.tight_layout() # fixes overlapping between subplots
# we can access plots by their indexes

i = 1
j = 1

axes_2[i, j].set_title('Plot 2')
axes_2[i, j].set_xlabel('X')
axes_2[i, j].set_ylabel('X Squared')
axes_2[i, j].plot(x_1, y_1)
```

Out[21]:

```
[<matplotlib.lines.Line2D at 0x7fbfba62f110>]
```



Apparences

Default colors:

- **b: blue**
- **g: green**
- **r: red**
- **c: cyan**
- **m: magenta**
- **y: yellow**
- **k: black**
- **w: white**

other options for colors:

- **color="0.75" creates a 75% gray**
- **color="#eeefff" to use hexcodes**
- **color="burlywoods" by names, list here**

style

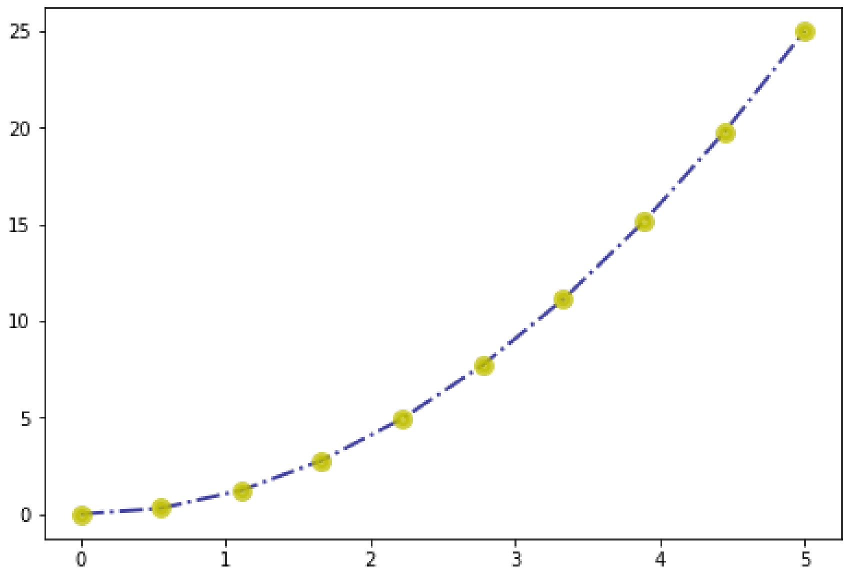
- **ls: linestile options here**
- **marker: options here**

In [23]:

```
# colors examples

fig_3 = plt.figure(figsize=(6,4))
axes_3 = fig_3.add_axes([0,0,1,1])
axes_3.plot(x_1, y_1,
            color="navy",
            alpha=.75,
            lw=2,
            ls="-.",
            marker='o',
            markersize=7,
            markerfacecolor='y',
            markeredgecolor='y',
            markeredgewidth=4)
```

Out[23]:

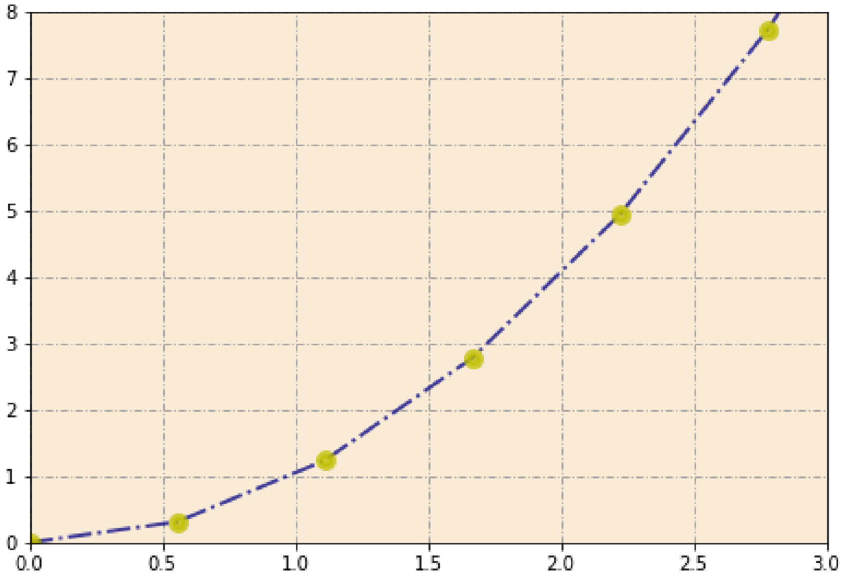
```
[<matplotlib.lines.Line2D at 0x7fbfba9913d0>]
```



In [26]:

```
# zoom into previous plot and background grid
# I want to zoom the axis x on 0 to 3 and y 0 to 8

# colors examples

fig_3 = plt.figure(figsize=(6,4))
axes_3 = fig_3.add_axes([0,0,1,1])
axes_3.plot(x_1, y_1,
            color="navy",
            alpha=.75,
            lw=2,
            ls="-.",
            marker='o',
            markersize=7,
            markerfacecolor='y',
            markeredgecolor='y',
            markeredgewidth=4)
```

```
# zoom
axes_3.set_xlim([0, 3])
axes_3.set_ylim([0, 8])

# grid
axes_3.grid(True, color='0.6', dashes=(5,2,1,2))

# background color
axes_3.set_facecolor('#FAEBD7')
```



Save fig to file

In [ ]:

```
fig_3.savefig('filename.png')
```