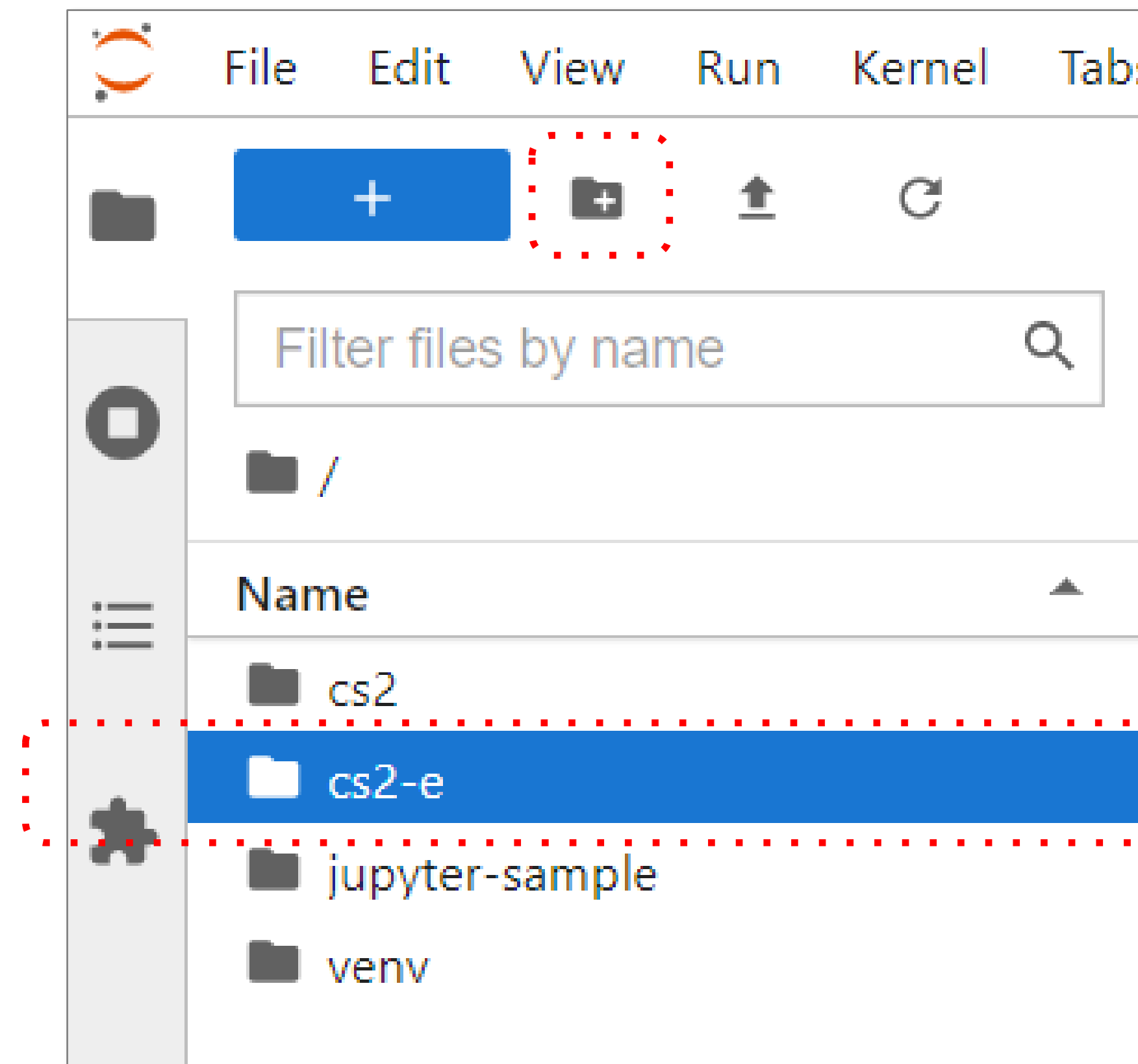


2. Matplotlibによる 様々な関数のグラフ描画

演習用フォルダの作成

- コンピュータサイエンス基礎演習Ⅱの演習用に「cs2-e」フォルダを作成しましょう



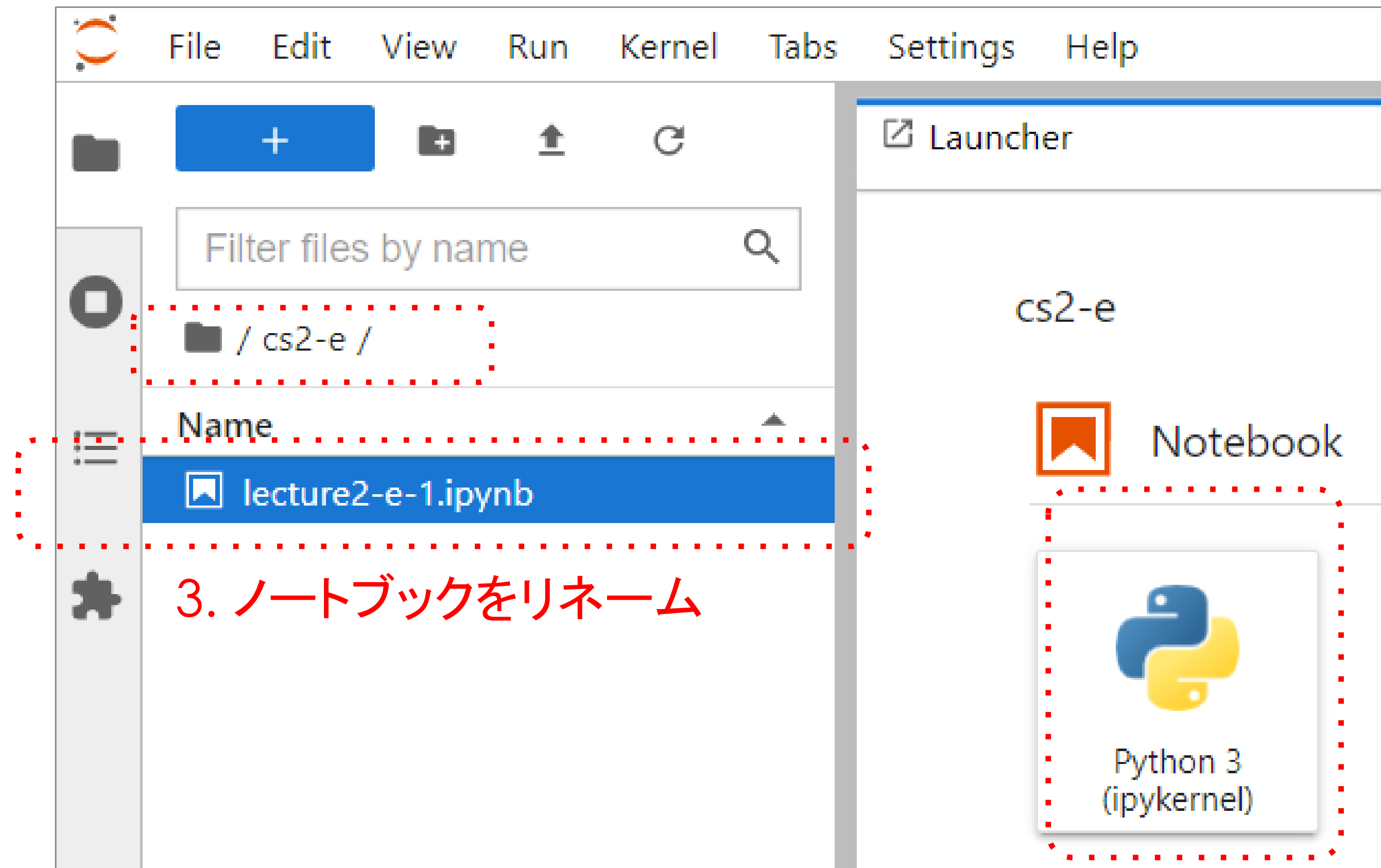
1. フォルダを作成

2. フォルダをリネーム

演習用のノートブックの作成

- 「cs2-e」の下に、lecture2-e-1.ipynb を作成しましょう

1. cs2-e
フォルダに移動



3. ノートブックをリネーム

2. ノートブックを作成

Matplotlib をインポートする

- lecture2-e-1.ipynbに以下のセルを作成します
 - Markdownセル
 - Codeセルを追加し、matplotlib.pyplot を plt という別名でインポートする

Markdownセル →

```
## 2-e-1 matplotlibで関数のグラフを描く
```

Codeセル →

```
import matplotlib.pyplot as plt  
import math
```

これから描画するグラフ

● x が 1 以上 10 以下の範囲で、次の関数のグラフを描画してみましょう

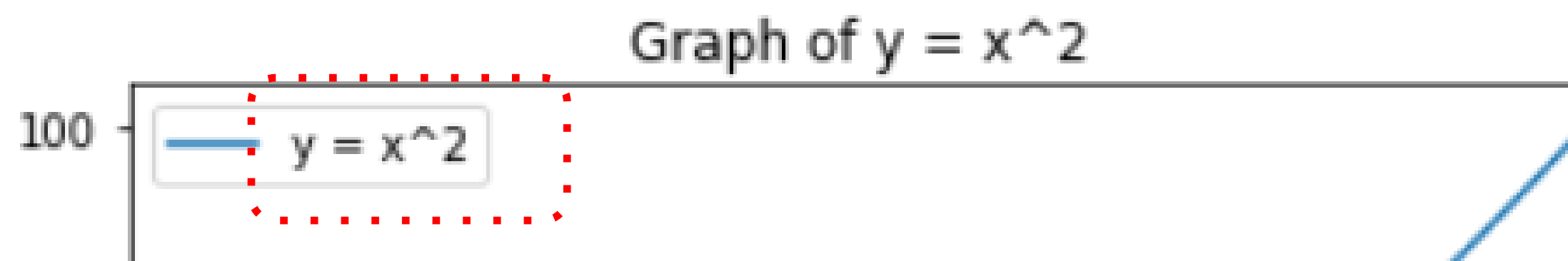
- 一次関数 $y = x$
- 二次関数 $y = x^2$
- 指数関数 $y = 2^x$
- 対数関数 $y = \log x$

● x の値は0.1ずつ変化させて直線で滑らかに繋ぎましょう

グラフのラベルに数式を表示する方法

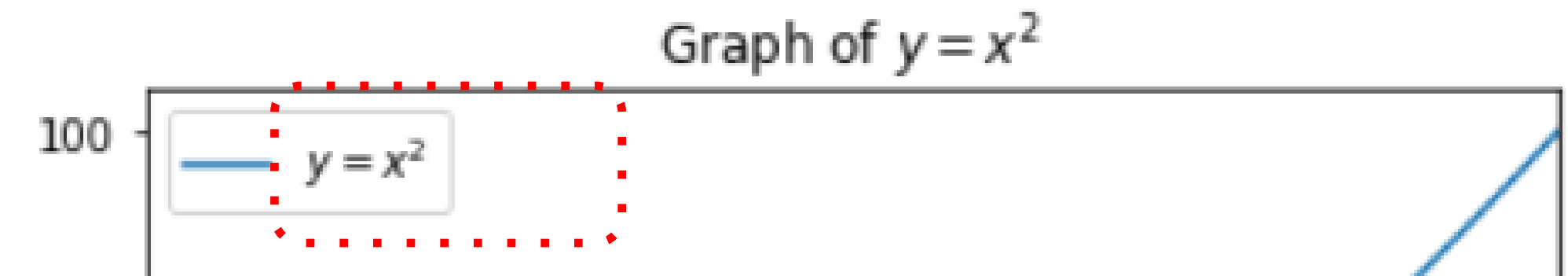
- グラフの凡例に、関数を表す数式を含めたい場合には、plotメソッドの引数 label に \$で囲んだ文字列（LaTeXと呼ばれる記法）を指定します

- '\$y = x\$' $\rightarrow y = x$
- '\$y = x ^ 2\$' $\rightarrow y = x^2$
- '\$y = 2 ^ x\$' $\rightarrow y = 2^x$
- '\$y = \log x\$' $\rightarrow y = \log x$



```
plt.plot(xs, ys, label='y = x^2')
```

labelの引数を文字列で指定した場合、
文字列がそのまま表示される



```
plt.plot(xs, ys, label='$y = x^2$')
```

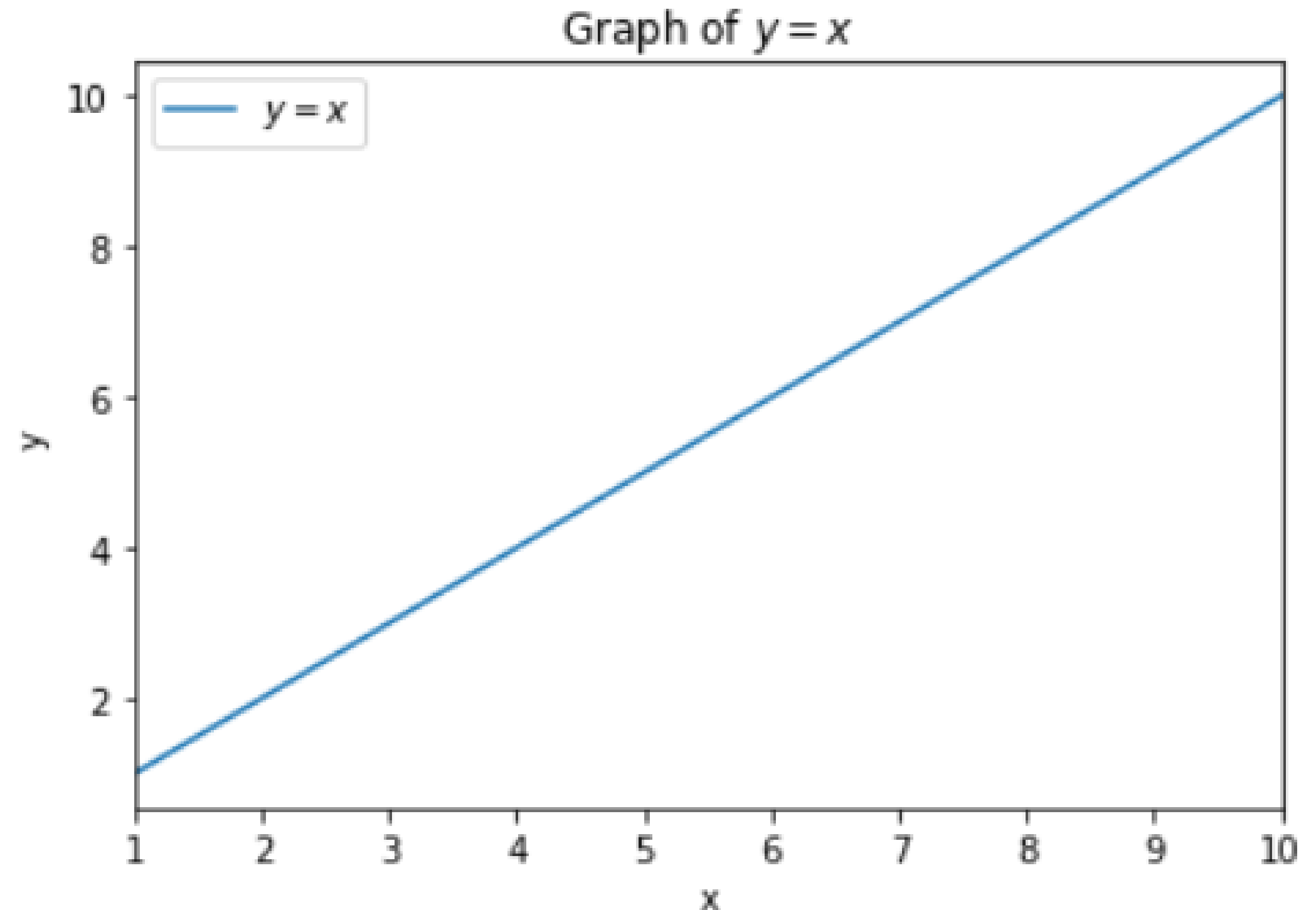
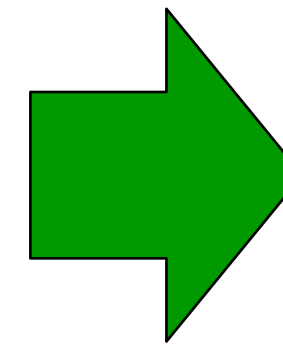
labelの引数をLaTeX記法で指定した場合、
数式として整形される

一次関数の描画 ($y = x$)

- Codeセルを作成し、以下のプログラムを書いて実行してみましょう
 - xlimとylim関数で、x軸とy軸の範囲を設定することができます

```
xs = []
ys = []
for i in range(10, 101):
    x = i * 0.1
    xs.append(x)
    ys.append(x)

plt.plot(xs, ys, label='$y = x$')
plt.xlim([1, 10])
plt.title('Graph of $y = x$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

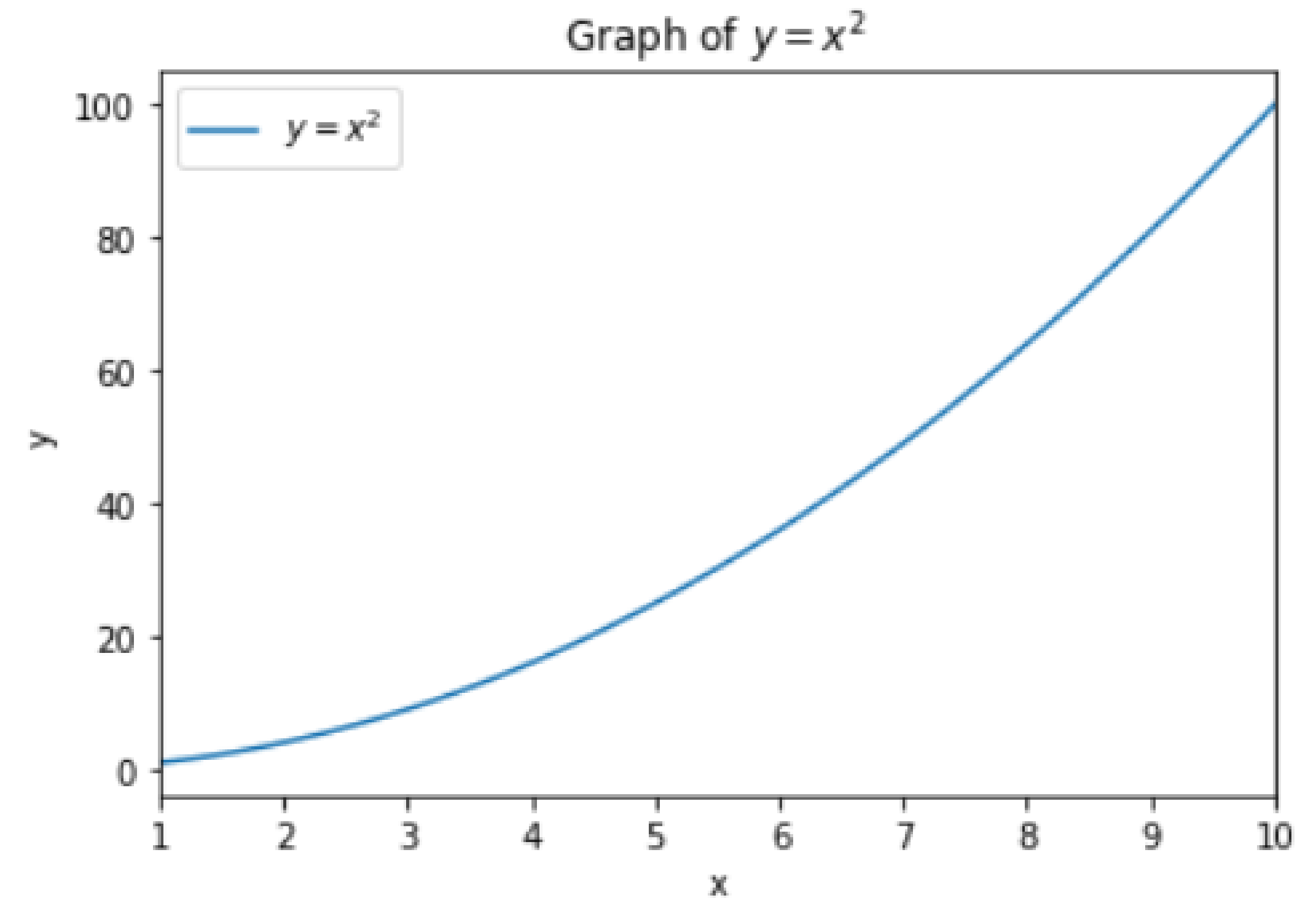
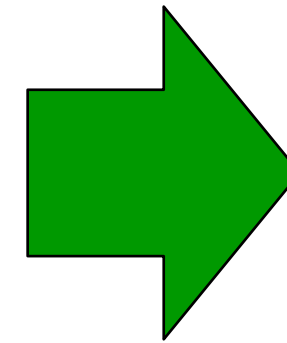


二次関数の描画 ($y = x^2$)

- Codeセルを作成し、以下のプログラムを書いて実行してみましょう

```
xs = []
ys = []
for i in range(10, 101):
    x = i * 0.1
    xs.append(x)
    ys.append(x*x)

plt.plot(xs, ys, label='$y = x^2$')
plt.xlim([1,10])
plt.title('Graph of $y = x^2$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

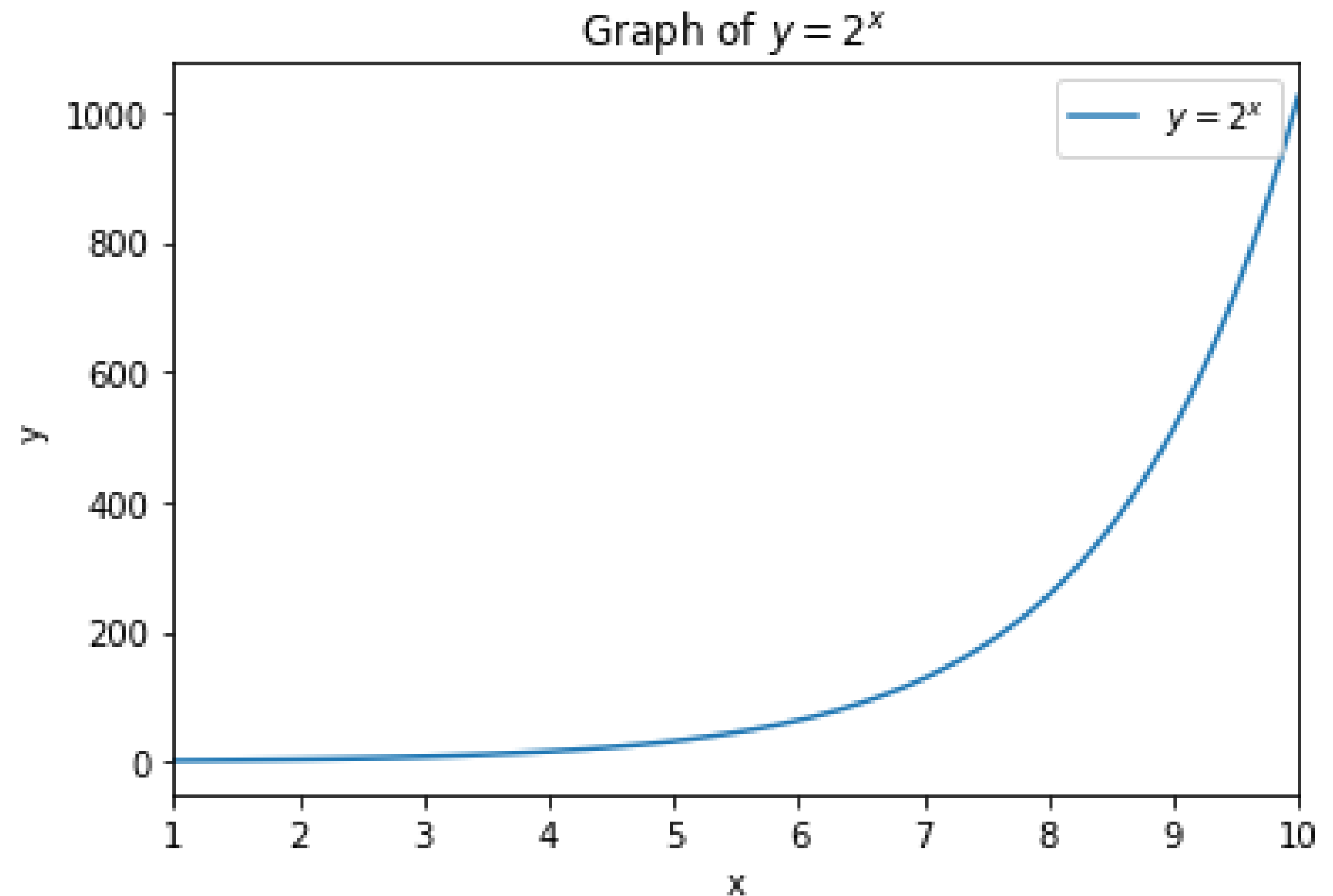
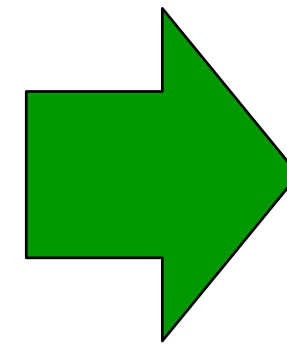


指数関数の描画 ($y = 2^x$)

- Codeセルを作成し、以下のプログラムを書いて実行してみましょう

```
xs = []
ys = []
for i in range(10, 101):
    x = i * 0.1
    xs.append(x)
    ys.append(2**x)

plt.plot(xs, ys, label='$y = 2^x$')
plt.xlim([1,10])
plt.title('Graph of $y = 2^x$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



対数関数の描画 ($y = \log x$)

- Codeセルを作成し、以下のプログラムを書いて実行してみましょう

```
xs = []
ys = []
for i in range(10, 101):
    x = i * 0.1
    xs.append(x)
    ys.append(math.log(x))

plt.plot(xs, ys, label='$y = \log x$')
plt.xlim([1,10])
plt.title('Graph of $y = \log x$')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

