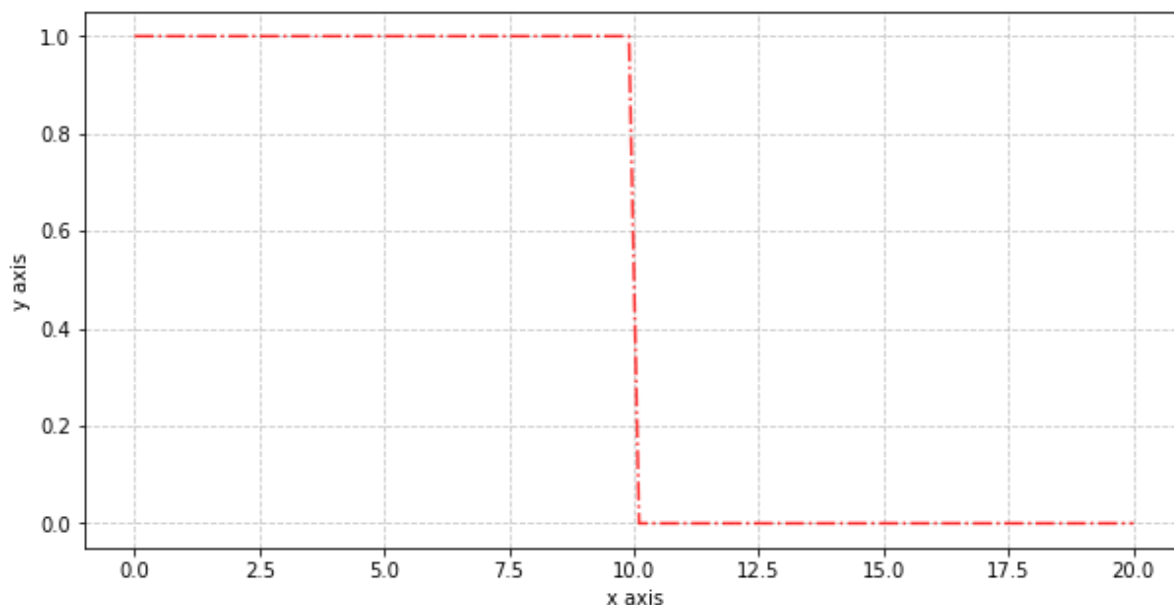


```
In [1]: # plot for given equations
```

```
In [2]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

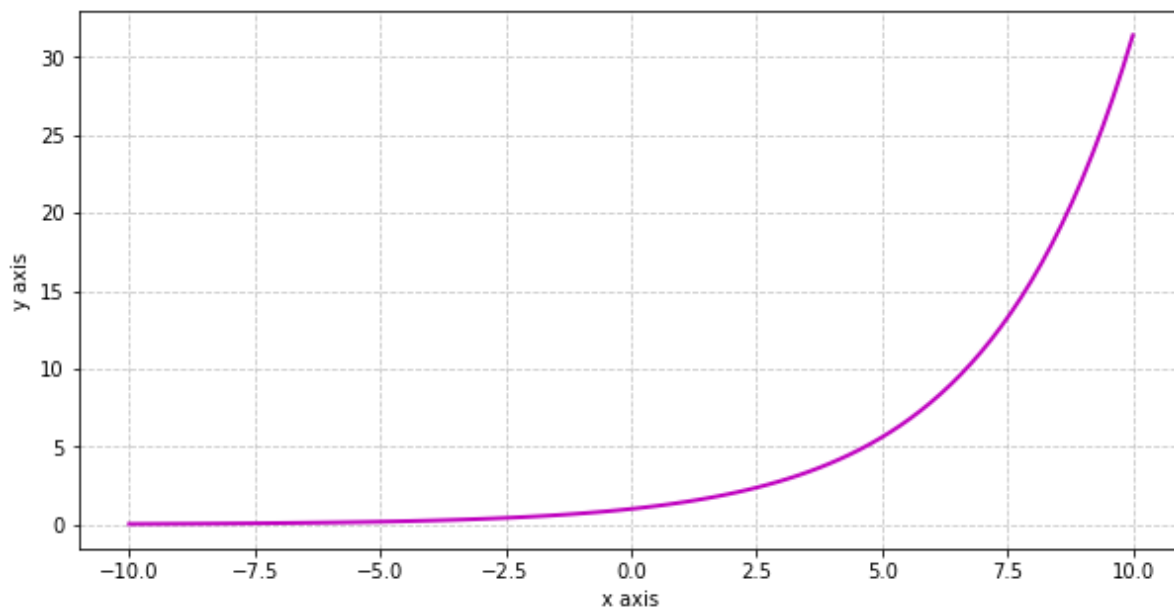
### 1. $y=1, x<10; y=0$ elsewhere

```
In [3]: x = np.linspace(0, 20, 100) #using 100 points to draw the plot
y = np.sign((10-x)+abs(10-x))
fig = plt.figure(figsize = (10, 5))
plt.plot(x, y, 'r', alpha=0.8, linestyle = '-.') # Create the plot
plt.grid(alpha = .7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.show()
```



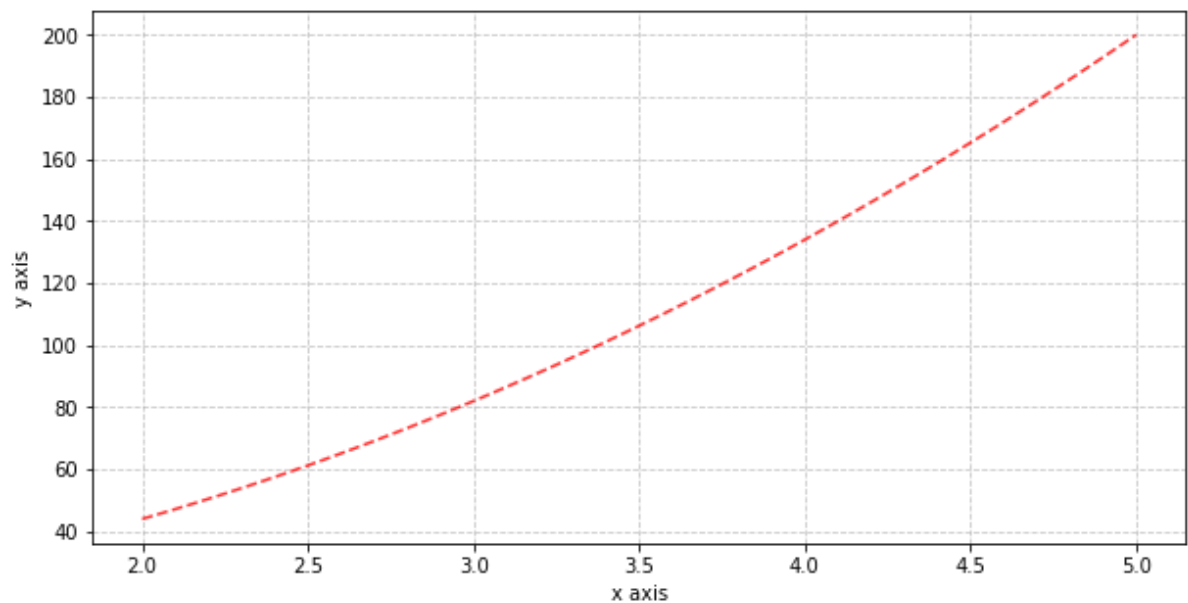
### $y=e^{ax}$ for different values of $a$

```
In [4]: import random
x = np.linspace(-10, 10, 100)
a=np.random.random()
y = np.exp(a*x)
fig = plt.figure(figsize = (10,5))
plt.grid(alpha =.7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.plot(x,y, 'm',linewidth=2)
plt.show()
```



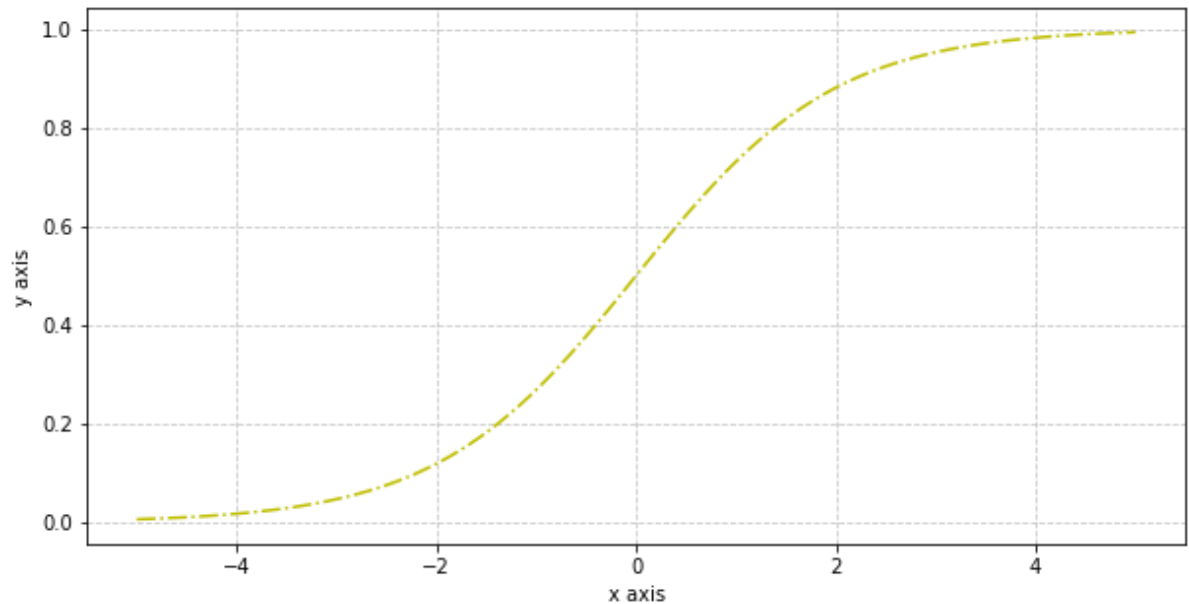
**$y=7x^2 + 3x + 10$  for  $2 \leq x \leq 5$**

```
In [5]: # Creating vectors X and Y
x = np.linspace(2, 5, 100) #using 100 points to draw the plot
y = 7*x**2 + 3*x + 10
fig = plt.figure(figsize = (10, 5))
plt.plot(x, y, 'r', alpha=0.8, linestyle = '--') # Create the plot
plt.grid(alpha = .7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.show() # Show the plot
```



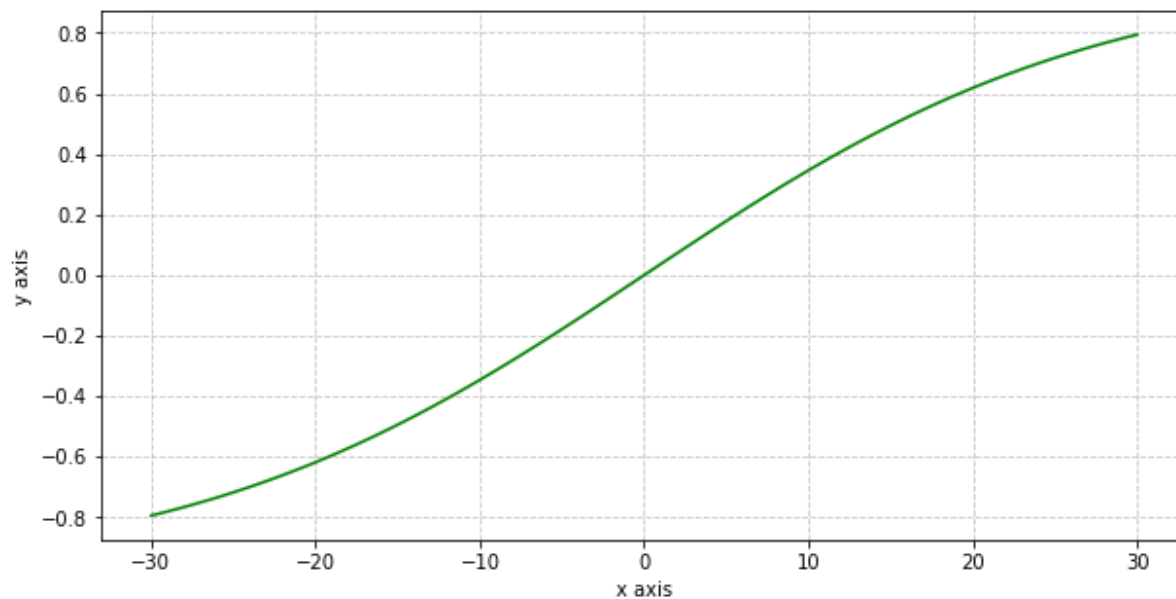
$$y=1/1+e^{-x}$$

```
In [6]: x = np.linspace(-5, 5, 100)
y = 1/(1+np.exp(-x))
fig = plt.figure(figsize = (10,5))
plt.plot(x,y,'y',linestyle='-.')
plt.grid(alpha =.7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.show()
```



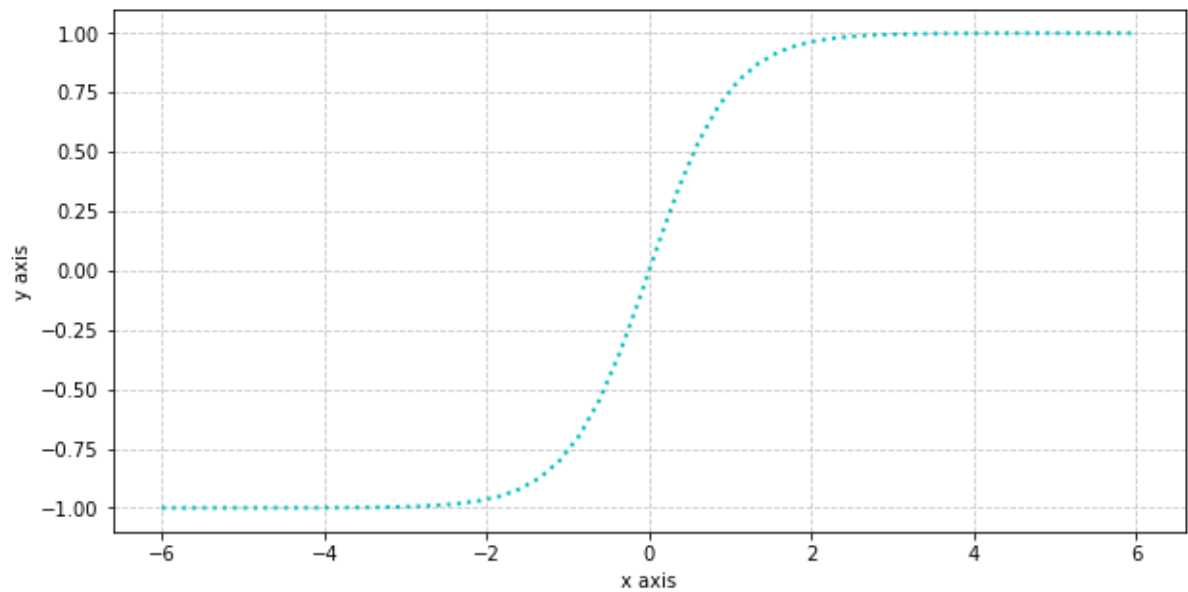
**$y = \frac{1-e^{-ax}}{1+e^{ax}}$  for different values of  $a$**

```
In [7]: import random
x = np.linspace(-30, 30, 100)
a=np.random.random()
y = (1-np.exp(-a*x))/(1+np.exp(-a*x))
fig = plt.figure(figsize = (10,5))
plt.grid(alpha =.7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.plot(x,y, 'g')
plt.show()
```



**y=tanhx**

```
In [8]: x = np.linspace(-6, 6, 50)
y = np.tanh(x)
fig = plt.figure(figsize = (10,5))
plt.plot(x,y,'c',linestyle=':',linewidth=2)
plt.grid(alpha =.7, linestyle = '--')
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.show()
```



```
In [ ]:
```