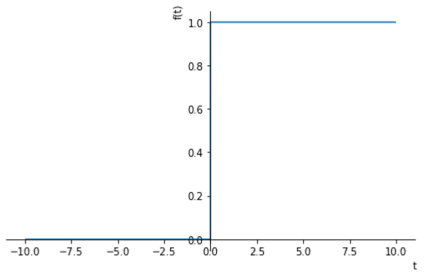
```python
import numpy as np
import matplotlib.pyplot as plt
import sympy
sympy.init_printing()
#%matplotlib inline
```

$$\mathcal{L}\{f(t)\} = \int_0^\infty f(t) e^{-st} \mathrm{d}s$$

```python
"""تعريف كل من ثابت الحساسية و ثابت الزمن و الزمن """
K, tau = sympy.symbols('K, tau',real=True, positive=True)
t, s = sympy.symbols('t, s')
#s = sympy.symbols('s')
a = sympy.symbols('a', real=True, positive=True)
"""تعريف دالة الخطوة """
u = sympy.Heaviside(t)
'''def u(t):
    """Unit step function"""
    return 1 * (t >= 0)'''
"""تعريف تابع لحساب تحويل لابلاس"""
def L(f):
    return sympy.laplace_transform(f, t, s, noconds=True)
"""تعريف تابع لحساب مقلوب لابلاس"""
def invL(F):
    return sympy.inverse_laplace_transform(F, s, t)
f = sympy.exp(-a*t)
U = L(u)
G = K/(tau*s + 1)
sympy.integrate(u*sympy.exp(-s*t) ,(t, 0, sympy.oo))
```

$$\begin{cases} \frac{1}{s} & \text{for}\ |\arg(s)| < \frac{\pi}{2} \\ \int\limits_0^\infty e^{-st}\theta(t)\, dt & \text{otherwise} \end{cases}$$

```python
sympy.plot(u)
```



```
<sympy.plotting.plot.Plot at 0x7f9f0277f910>
```

Y(s) = $\frac{K}{(s\tau+1)s}$

سوف نفرق الكسور و بتعويض قيمة S

* Y(s) = $\frac{A1}{s\tau+1} + \frac{A2}{s}$

A1 = | $(s\tau + 1)$ $\frac{K}{s(s\tau+1)}$ $|s = -\frac{1}{\tau}$

A2 = | s $\frac{K}{(s\tau+1)s}$ $|s = 0$

A1 = $-\tau K$

A2 = K

و بتعويض في المعادلة *

Y(s) = $-\frac{\tau K}{s\tau+1} + \frac{K}{s}$

سوف نقوم بتجزئة الكسور من خلال التابع apart

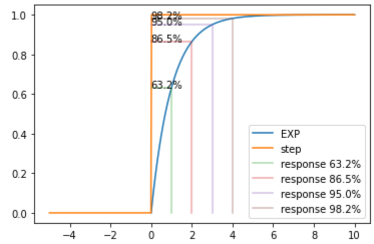
```python
(G*U).apart(s)
```

$$-\frac{K\tau}{s\tau+1} + \frac{K}{s}$$

```python
invL((G*U).apart(s))
```

$$K\theta(t) - Ke^{-\frac{t}{\tau}}\theta(t)$$

```
t_step ,T= np.linspace(-5, 10, 500) ,1
t = np.linspace(0, 10, 500)
y ,u = 1 - np.exp(-t/T),1
tau_height_01 ,tau_height_02 = 1 - np.exp(-1) ,1 - np.exp(-2)
tau_height_03 ,tau_height_04 = 1 - np.exp(-3) ,1 - np.exp(-4)
tau_height_1 ,tau_height_2 = np.round(((1 - np.exp(-1))*100),1),np.round(((1 - np.exp(-2))*100),1)
tau_height_3 ,tau_height_4 = np.round(((1 - np.exp(-3))*100),1),np.round(((1 - np.exp(-4))*100),1)
def u(t):
    """Unit step function"""
    return 1 * (t >= 0)
fig, ax = plt.subplots()
line1 = ax.plot(t, y, label='EXP')
line2 = ax.plot(t_step, u(t_step) , label='step')
T=1
Line3 = ax.plot([0, T, T], [1*tau_height_01]*2 + [0], alpha=0.4,label='response {:.3}%'.format(tau_height_01*100))
ax.text(0, 1*tau_height_01, '{:.3}%'.format(tau_height_01*100))
T=2
Line4 = ax.plot([0, T, T], [1*tau_height_02]*2 + [0], alpha=0.4,label='response {:.3}%'.format(tau_height_02*100))
ax.text(0, 1*tau_height_02, '{:.3}%'.format(tau_height_02*100))
T=3
Line5 = ax.plot([0, T, T], [1*tau_height_03]*2 + [0], alpha=0.4,label='response {:.3}%'.format(tau_height_03*100))
ax.text(0, 1*tau_height_03, '{:.3}%'.format(tau_height_03*100))
T=4
Line6 = ax.plot([0, T, T], [1*tau_height_04]*2 + [0], alpha=0.4,label='response {:.3}%'.format(tau_height_04*100))
ax.text(0, 1*tau_height_04, '{:.3}%'.format(tau_height_04*100))
#fig.legend((line1, line2), ('Line 1', 'Line 2','line3'), 'upper left')
ax.legend()
plt.show()
print(str(tau_height_1)+"%" , str(tau_height_2)+"%", str(tau_height_3)+"%",str(tau_height_4)+"%")
```




```
63.2% 86.5% 95.0% 98.2%
```

0s completed at 7:42 AM