1. FONCTIONS

Solution L1

1) [console]

```
In []:   K=np.arange(0,2*np.pi,1.5)

In []:   K
Out[]:   array([0. , 1.5, 3. , 4.5, 6. ])
```

variante :

```
In []:   print(K)
[0.  1.5 3.  4.5 6. ]
```

2) [console]

```
In []:   L=np.linspace(0,2*np.pi,5)

In []:   print(L)
[0.         1.57079633 3.14159265 4.71238898 6.28318531]
```

3) [console]

```
In []:   len(L)
Out[]:   5
```

Solution L2 [console]

```
In []:   X=10

In []:   T0=1

In []:   w0=2*np.pi/T0

In []:   X*np.cos(w0*.5)
Out[]:   -10.0

In []:   X*np.cos(w0*(np.linspace(0,2,20)))
Out[]:
array([10.        ,  7.89140509,  2.45485487, -4.01695425, -8.79473751,
       -9.86361303, -6.77281572, -0.82579345,  5.46948158,  9.45817242,
        9.45817242,  5.46948158, -0.82579345, -6.77281572, -9.86361303,
       -8.79473751, -4.01695425,  2.45485487,  7.89140509, 10.        ])
```

Solution F1

1) [script]

```
# mouvement oscillatoire en régime critique

#chargement module
import numpy as np

# paramètres connus
xeq=-0.5                # position d'équilibre
T=0.8                   # période propre
x0=1                    # position initiale
v0=0                    # vitesse initiale

#paramètres calculés
w=2*np.pi/T             #pulsation propre
A=v0+w*(x0-xeq)
B=x0-xeq

#fonction
def x(t):
    return xeq+(A*t+B)*np.exp(-w*t)
```

2) [console]

```
In [ ]:   x(0)
Out[ ]:   1.0

In [ ]:   x(0.1)
Out[ ]:   0.7210466439044212

In [ ]:   x(0.3)
Out[ ]:   -0.022848697408352436

In [ ]:   x(1)
Out[ ]:   -0.49484428394327357

In [ ]:   x(2)
Out[ ]:   -0.4999962231216078
```

3) [console]

```
In [ ]:   x(np.linspace(0,1,10))
Out[ ]:
array([ 1.        ,  0.67370197,  0.21895024, -0.10410612, -0.29468175,
       -0.39753912, -0.45022217, -0.47629038, -0.48887709, -0.49484428])
```

Solution F2 [script]

```
#définition du seuil de température pour la baignade (à l'appréciation de chacun !)
Tmin=22

#question
T = float(input('Quelle température fait-il dehors (en °C) ? : '))
if T>Tmin:
    température='chaud'
else:
    température='froid'
print ('il fait ', température)
```

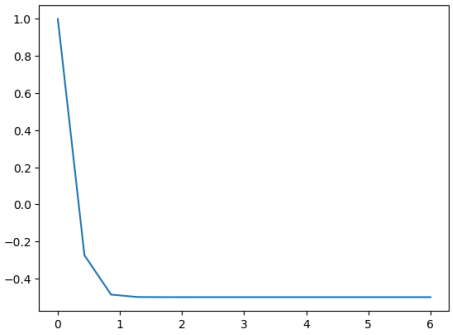
2. GRAPHE D'UNE FONCTION À 1 VARIABLE

Solution G1 [script]

Nous devons ajouter les lignes suivantes :

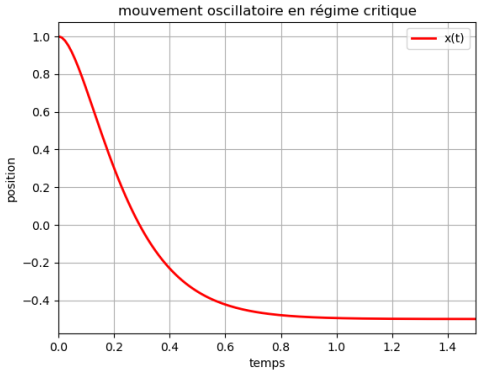
```
import matplotlib.pyplot as plt      # à placer de préférence
                                     au début du script
#graphe
t=np.linspace(0,6,15)
plt.plot(t,x(t))
plt.show()
```

Nous obtenons :



Solution G2 [script]

```
#graphe
t=np.linspace(0,6,500)
plt.plot(t,x(t),color='r',linewidth=2,label='x(t)')
plt.legend()
plt.xlim(0, 1.5)
plt.title('mouvement oscillatoire en régime critique')
plt.xlabel('temps')
plt.ylabel('position')
plt.grid()
plt.show()
```



Solution G3 [script]

```
#chargement modules
import numpy as np
import matplotlib.pyplot as plt

#graphe 1
exec(open('mouvement oscillatoire - tout régime.py').read())
t=np.linspace(0,3,500)
plt.plot(t,x(t),color='b',label=f'Q={Q}')
plt.legend()

#graphe 2
exec(open('mouvement oscillatoire - tout régime.py').read())
t=np.linspace(0,3,500)
plt.plot(t,x(t),color='r',label=f'Q={Q}')
plt.legend()

#graphe 3
exec(open('mouvement oscillatoire - tout régime.py').read())
t=np.linspace(0,3,500)
plt.plot(t,x(t),color='g',label=f'Q={Q}')
plt.legend()

#graphe 4
exec(open('mouvement oscillatoire - tout régime.py').read())
t=np.linspace(0,3,500)
plt.plot(t,x(t),color='c',label=f'Q={Q}')
plt.legend()

#améliorations graphiques communes
plt.xlim(0, 3)
plt.title('oscillations libres amorties, influence de Q')
plt.xlabel('temps')
plt.ylabel('position')

plt.show()
```