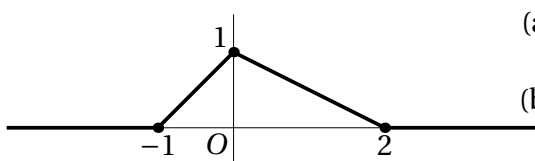


TP 5 : Instructions conditionnelles

- ⚠ • Démarrer chaque exercice en ouvrant un nouveau fichier dans l'éditeur (menu *Fichier / Nouveau Fichier*) ;
- Sauvegarder immédiatement ce nouveau fichier (menu *Fichier / Enregistrer sous*) en lui donnant un nom approprié. Pour vous aider, des noms sont suggérés pour chaque exercice.

1 Applications directes

Exercice 1 Une fonction / `fonction_triangle.py`. On considère la fonction f dont la représentation graphique est la suivante :



- Définir cette fonction $\mathbf{f(x)}$. Vérifier en calculant quelques valeurs particulières de f .
- Représenter graphiquement la fonction f sur l'intervalle $[-2, 3]$; on donne pour cela les commandes :

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-2, 3, 100)
plt.plot(x, [f(xi) for xi in x])
plt.show()
```

Exercice 2 Une suite chaotique / `suite_chaotique.py`. On considère la suite (u_n) définie par :

$$u_0 = 0.1 \quad \text{et} \quad \forall n \in \mathbb{N}, u_{n+1} = \begin{cases} 3u_n & \text{si } 3u_n < 1 \\ 3u_n - 1 & \text{si } 1 \leq 3u_n < 2 \\ 3u_n - 2 & \text{sinon} \end{cases}$$

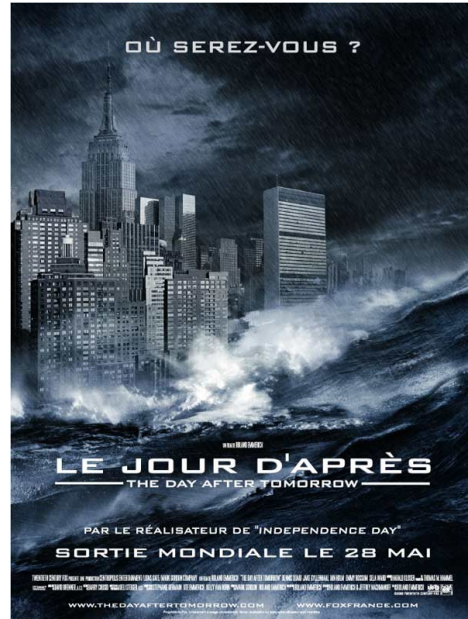
- Écrire une fonction $\mathbf{u(n)}$ qui calcule les termes de cette suite. Vérifier que $\mathbf{u(10)}$ est proche de 0.9. Noter la valeur de $\mathbf{u(1000)}$.
- On prend maintenant $u_0 = 0.1 + 10^{-10}$. Calculer à nouveau $\mathbf{u(1000)}$.

On dit que le système décrit par la suite (u_n) est chaotique : de petites variations sur la condition initiale peuvent conduire à des variations importantes à long terme.

2 Mise en œuvre

Exercice 3 Le jour d'après / `jour_suivant.py`. Le but de l'exercice est de créer un programme qui, à partir d'un numéro de jour, de mois et d'année, donne la date du lendemain. On donne ci-dessous quelques exemples de ce que l'on souhaite que fasse le programme :

En entrée			En sortie		
jour	mois	année	jour	mois	année
14	3	2014	15	3	2014
31	3	2014	1	4	2014
31	12	2014	1	1	2015
28	2	2014	1	3	2014
28	2	2016	29	2	2016
28	2	2100	1	3	2100
28	2	2400	29	2	2400



Rappel : Les années bissextiles sont les années qui sont multiples de 4, sauf celles qui sont multiples de 100 qui ne sont pas bissextiles. Toutefois, les années qui sont multiples de 400 sont bissextiles.

- (a) On considère l'extrait de code suivant (à recopier dans votre programme). Prenez le temps de comprendre ce que fait cette fonction puis complétez le dernier commentaire.

```
def moisLong(m) :
    # La valeur du paramètre correspond au numéro
    # du mois considéré
    if m==1 or m==3 or m==5 or m==7 or m==8 or m==10 or m==12:
        rep = True
    else:
        rep = False
    return rep
    # Cette fonction renvoie...
```

- (b) Écrire une fonction **finAnnee(j,m)** prenant en entrée un numéro de jour **j** et de mois **m** et retournant une variable de type booléen indiquant s'il s'agit du dernier jour de l'année.
- (c) Écrire une fonction **estBissextile(a)** prenant en entrée un numéro d'année **a** et retournant une variable de type booléen indiquant si l'année est bissextile.
- (d) Écrire une fonction **finMois(j,m,a)** prenant en entrée un numéro de jour **j**, de mois **m** et d'année **a** et retournant une variable de type booléen, indiquant s'il s'agit du dernier jour du mois.
- (e) Écrire, à la suite des fonctions précédemment écrites, un programme répondant au problème posé initialement. Ce programme devra faire appel aux fonctions écrites précédemment et pourra commencer par la définition de trois variables, représentant le numéro de jour, de mois et d'année, auxquelles vous pourrez donner les valeurs qui vous sembleront les plus pertinentes.

Corrections

Ex 1. Sur l'intervalle $[-1, 0]$, la fonction f est égale à la fonction affine $x \mapsto x + 1$ et sur l'intervalle $[0, 2]$, f est égale à la fonction affine $x \mapsto 1 - x/2$.

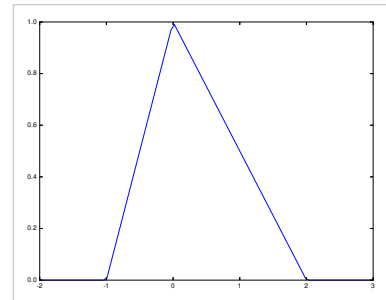
```
def f(x):
    if -1<=x<=0:
        return x+1
    elif 0<x<=2:
        return 1-float(x)/2
    else:
        return 0
print(f(-2), f(-0.5), f(0), f(0.5), f(1), f(2))
```

0 0.5 1 0.75 0.5 0.0

Représentation graphique :

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-2, 3, 100)
plt.plot(x, [f(xi) for xi in x])
```

```
plt.show()
```



Ex 2. Pour simplifier on définit une fonction $u(u0, n)$:

```
def u(u0, n):
    u = u0
    for k in range(n):
        if 3*u<1:
            u = 3*u
        elif 1<=3*u<2:
            u = 3*u-1
        else:
            u = 3*u-2
    return u
print(u(0.1, 10))
```

0.90000000000003885

```
print(u(0.1, 1000), u(0.1+10**(-10), 1000))
```

0.2899557559269601 0.8206415852159625

Ex 3.

```

def moisLong(m) :
    # La valeur du paramètre correspond au numéro
    # du mois considéré
    if m==1 or m==3 or m==5 or m==7 or m==8 or m==10 or m==12:
        rep = True
    else:
        rep = False
    return rep
    # Cette fonction renvoie True si le mois m comporte
    # 31 jours et False sinon

def finAnnee(j,m) :
    if j==31 and m==12:
        return True
    else:
        return False

def estBissextile(a) :
    if a%4==0 and (a%100!=0 or a%400==0):
        return True
    else:
        return False

def finMois(j,m,a) :
    if moisLong(m) :
        # C'est un mois de 31 jours
        return j==31
    elif m==2:
        # C'est le mois de février
        if estBissextile(a):
            return j==29
        else:
            return j==28
    else:
        # C'est un mois de 30 jours
        return j==30

```

```

def jourSuivant(j,m,a):
    if finMois(j,m,a):
        # C'est la fin du mois,
        # il faut passer au mois suivant
        # sauf si c'est la fin de l'année
        if finAnnee(j,m):
            return (1,1,a+1)
        else:
            return (1,m+1,a)
    else:
        # Ce n'est pas la fin du mois,
        # il suffit d'augmenter le jour de 1
        return (j+1,m,a)

```

```

print(jourSuivant(14,3,2014))
print(jourSuivant(31,3,2014))
print(jourSuivant(31,12,2014))
print(jourSuivant(28,2,2014))
print(jourSuivant(28,2,2016))
print(jourSuivant(28,2,2100))
print(jourSuivant(28,2,2400))

```

```

(15, 3, 2014)
(1, 4, 2014)
(1, 1, 2015)
(1, 3, 2014)
(29, 2, 2016)
(1, 3, 2100)
(29, 2, 2400)

```