

# Python-ECG1-03-if-cor

September 7, 2022

## 1 P3 - Instruction conditionnelle if

### 1.0.1 Exercice 1

On considère la fonction

$$f: x \mapsto \begin{cases} 2x - 1 & \text{si } x \leq 2 \\ \ln(x) & \text{si } x > 2 \end{cases}$$

Écrire un programme qui, étant donné un réel  $x$ , affiche la valeur de  $f(x)$ . On testera tous les cas.

On pensera à charger la bibliothèque numpy.

```
[1]: import numpy as np

# Cas 1
x = 2
if x <= 2 :
    print(2*x-1)
else :
    print(np.log(x))

# Cas 2
x = np.e
if x <= 2 :
    print(2*x-1)
else :
    print(np.log(x))
```

3  
1.0

### 1.0.2 Exercice 2

Écrire une suite d'instructions en langage Python qui, étant donnés des nombres  $a$  et  $b$ , affiche lequel est le plus petit des deux (par exemple "a est le plus petit") ou s'il y a égalité.

```
[2]: # Cas 1
a = 2
b = 5
```

```

if a < b :
    print('a est le plus petit')
elif b < a:
    print('b est le plus petit')
else :
    print('a = b')

# Cas 2
a = 3
b = 2
if a < b :
    print('a est le plus petit')
elif b < a:
    print('b est le plus petit')
else :
    print('a = b')

# Cas 3
a = 2
b = 2
if a < b :
    print('a est le plus petit')
elif b < a:
    print('b est le plus petit')
else :
    print('a = b')

```

a est le plus petit  
b est le plus petit  
a = b

### 1.0.3 Exercice 3

Écrire un programme prenant en entrée trois nombres a,b,c (avec  $a \neq 0$ ) et affichant les éventuelles solutions *réelles* de l'équation  $ax^2 + bx + c = 0$ .

Tester votre fonction avec les équations résolues sur papier.

```

[3]: # Équation 1
a = 2
b = -6
c = 4

Delta = b**2 - 4*a*c
if Delta > 0 :
    x1 = (-b-np.sqrt(Delta))/(2*a)
    x2 = (-b+np.sqrt(Delta))/(2*a)
    print('Deux solutions :',x1,'et',x2)

```

```

elif Delta == 0:
    x0 = -b/(2*a)
    print('Une solution :',x0)
else:
    print('Pas de solution')

# Équation 2
a = 2
b = 12
c = 18

Delta = b**2 - 4*a*c
if Delta > 0 :
    x1 = (-b-np.sqrt(Delta))/(2*a)
    x2 = (-b+np.sqrt(Delta))/(2*a)
    print('Deux solutions :',x1,'et',x2)
elif Delta == 0:
    x0 = -b/(2*a)
    print('Une solution :',x0)
else:
    print('Pas de solution')

# Équation 3
a = 1
b = 1
c = 1

Delta = b**2 - 4*a*c
if Delta > 0 :
    x1 = (-b-np.sqrt(Delta))/(2*a)
    x2 = (-b+np.sqrt(Delta))/(2*a)
    print('Deux solutions :',x1,'et',x2)
elif Delta == 0:
    x0 = -b/(2*a)
    print('Une solution :',x0)
else:
    print('Pas de solution')

```

Deux solutions : 1.0 et 2.0  
 Une solution : -3.0  
 Pas de solution

#### 1.0.4 Exercice 4 - Simulation d'un tirage dans une urne

On considère une urne contenant 5 boules rouges et 7 boules jaunes. Écrire un programme réalisant une simulation d'un tirage dans cette urne. Celui-ci doit afficher "rouge" ou "jaune" avec les bonnes probabilités.

On chargera la bibliothèque `numpy.random`

```
[4]: import numpy.random as rd

# numéros 1 à 5 -> rouge : numéros 6 à 12 -> jaune
alea = rd.randint(1,13)

if alea <= 5 :
    print('rouge')
else :
    print('jaune')
```

rouge

### 1.0.5 Exercice 5

1) Comment tester, à l'aide d'un booléen, si un entier  $n$  est pair ?

```
[5]: # Cas 1
n = 15
np.floor(n/2) == n/2
```

[5]: False

```
[6]: # Cas 2
n = 10
np.floor(n/2) == n/2
```

[6]: True

Il est possible de réaliser des **divisions euclidiennes** (division entière, avec reste) en python à l'aide des commandes suivantes :

`a // b` : quotient  
`a % b` : reste

Par exemple, dans la division euclidienne de 17 par 5, le quotient vaut 3 et le reste 2 car

$$17 = 5 \times 3 + 2.$$

```
[7]: 17 // 5
```

[7]: 3

```
[8]: 17 % 5
```

[8]: 2

2) En déduire une deuxième façon de tester si un entier  $n$  est pair.

```
[9]: # cas 1
n = 15
n%2 == 0
```

[9]: False

```
[10]: # cas 2
n = 10
n%2 == 0
```

[10]: True

3) Écrire un programme qui, étant donnée une année, affiche **True** si celle-ci est bissextile et affiche **False** sinon.

Une année est bissextile (366 jours) si l'un des deux cas suivants se produit :

- l'année est divisible par 4 et non divisible par 100 ;
- l'année est divisible par 400.

Sinon, elle est non-bissextile (365 jours).

bissextiles : 2020, 2000

non bissextiles : 2022, 2100

```
[11]: annee = 2020

if (annee%4 == 0 and annee%100 != 0) or (annee%400 == 0):
    print('bissextile')
else :
    print('non bissextile')
```

bissextile

```
[12]: annee = 2000

if (annee%4 == 0 and annee%100 != 0) or (annee%400 == 0):
    print('bissextile')
else :
    print('non bissextile')
```

bissextile

```
[13]: annee = 2022

if (annee%4 == 0 and annee%100 != 0) or (annee%400 == 0):
    print('bissextile')
else :
    print('non bissextile')
```

non bissextile

```
[14]: annee = 2100
```

```
if (annee%4 == 0 and annee%100 != 0) or (annee%400 == 0):  
    print('bissextile')  
else :  
    print('non bissextile')
```

non bissextile