

P3

INSTRUCTION CONDITIONNELLE IF

I Les booléens

Un **booléen** est une variable Python qui vaut soit **True** (vrai) soit **False** (faux).

[In]: `x = 5
print(x > 2)`

[Out]: True

Pour écrire un booléen, on dispose des opérateurs suivants :

	Syntaxe Python	Exemples
Test d'égalité	<code>==</code>	<code>1 == 1.0</code> renvoie True <code>0.33 == 1/3</code> renvoie False
Test de non-égalité \neq	<code>!=</code>	<code>1 != 1.0</code> renvoie False <code>0.33 != 1/3</code> renvoie True
Comparaisons	<code><</code> , <code><=</code> <code>></code> , <code>>=</code>	<code>1 < 2</code> renvoie True <code>4 >= 5</code> renvoie False
Et	<code>and</code>	<code>1 == 1.0 and 2 == 3</code> renvoie False
Ou	<code>or</code>	<code>1 == 1.0 or 2 == 3</code> renvoie True
Contraire	<code>not</code>	<code>not(1 == 1.0)</code> renvoie False

!! Attention !! Ne pas confondre = et ==

- `A = 2` signifie « A prend la valeur 2 »
- `A == 2` signifie « A est-il égal à 2 ? » (test d'égalité)

Exemple

Soit x un nombre supposé défini.

Comment tester si x est dans $[2, 7]$?

Comment tester si x est dans $] -\infty, -5[\cup [2, +\infty[$?

II Instruction if

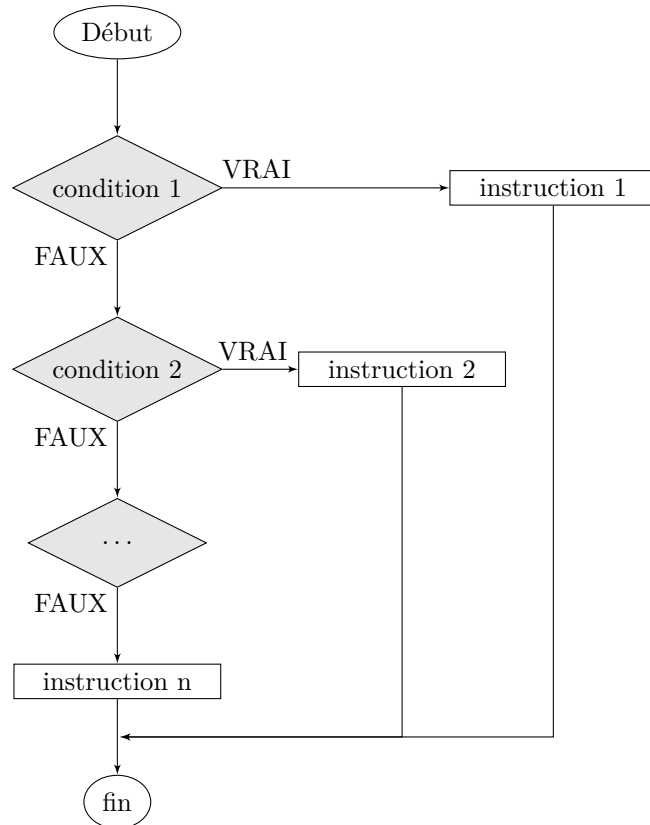
Nous allons voir comment écrire un programme python comportant une instruction conditionnelle du type **si ... alors ...**

Algorithme

Si condition 1,
Faire l'instruction 1
Sinon, si condition 2,
Faire l'instruction 2
...
Sinon
Faire l'instruction *n*
Fin du Si

Programme python

```
if condition 1 :
    instruction 1
elif condition 2 :
    instruction 2
elif condition 3 :
    instruction 3
...
else :
    instruction n
```



- Les *conditions* sont des booléens.
- **Ne pas oublier :**
 - les deux points qui indiquent la fin de chaque condition ;
 - l'indentation des blocs d'instructions par rapport à la ligne if, elif ou else dont ils dépendent ;
 - la fin de l'indentation indique la fin de l'instruction conditionnelle.
 - **elif et else sont facultatifs**

```
# Exemple
x = 17 # on peut changer x ici

if x > 15:
    print("x est strictement supérieur à 15")
elif x >= 11:
    print(" x est entre 11 et 15")
else :
    print(" x est inférieur ou égal à 10")
```

III Exercices

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.

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.

```
# Compléter avec des valeurs de a et b puis changer pour tester
# tous les cas de figure.
a =
b =
# votre programme
```

Exercice 3 1. [À préparer] Résoudre les équations suivantes.

$$2x^2 - 6x + 4 = 0; \quad 2x^2 + 12x + 18 = 0; \quad x^2 + x + 1 = 0$$

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

```
a =  
b =  
b =
```

Tester votre fonction sur les équations précédentes.

Exercice 4 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`

Exercice 5

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

Il est possible de réaliser des **divisions euclidiennes** (divisions entières, 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

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

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

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).

Exemples :

- bissextiles : 2020, 2000
- non bissextiles : 2022, 2100