

# TP2 Algorithmique/Python : utilisation de tests...



Exemple de programme en Python utilisant un test :

<p>① temp=float(input("Quelle température fait-il ?"))</p> <p>② if temp&lt;=0 :</p> <p style="padding-left: 40px;">print("L'eau gèle")</p> <p style="padding-left: 40px;">print("Attention au 'erglas'")</p> <p>else :</p> <p>④ print("Il y a un risque de 'erglas'")</p> <p>⑤ print("2in du contrôle de température")</p>	<p>① La machine affiche le texte "Quelle température fait-il ?" et demande à l'utilisateur d'entrer une réponse. Cette réponse est un <i>texte</i> et est convertie en <i>nombre</i> par float( )#. Enfin la réponse est stockée dans la variable temp#</p> <p>② Python regarde si (if) la valeur dans temp est inférieure ou égale à 0#</p> <p>③ , si oui Python affiche "L'eau gèle" et "Attention au 'erglas'". Les deux instructions sont <b>décalées à l'échelle</b> pour indiquer qu'elles doivent être exécutées si temp &lt;= 0#</p> <p>④ , si non (else:) Python affiche "Il y a un risque de 'erglas'". L'indentation de l'instruction est décalée#</p> <p>⑤ Tous les ans (pas de décalage) Python affiche "2in du contrôle de température"#</p>
--	--



Remarque :

En Python pour écrire une condition (qui suit le if) on utilise un des opérateurs suivants :

==			pour tester l'égalité
<			pour tester une non égalité
<	ou	>	pour tester une infériorité ou supériorité stricte
<=	ou	>=	pour tester une infériorité ou supériorité large
in			pour tester une appartenance à un ensemble



## Exercice 1 : compréhension d'algorithmes

89) Écrivez deux algorithmes écrits en langage naturel (ou pseudo-code) et un écrit en Python. Indiquez la valeur de la variable à la fin de chaque un d'entre eux.

**Si**

**Alors**

  

**Sinon**

  

**FinSi**

**Si**

**Alors**

  

**Sinon**

  

**FinSi**

**E = =**

  

**if** :

  

**else:**



) ! "\$ 4"" ' ! ! 15) '\$ " &(' \$ #1 \$ #1 \$ ' + # ! '\$ " &.

! 4

!&\* \$ \$' " &

) ! "\$ 4' \$ #1 \$ #1 \$ \_\_\_\_\_ + # ! " ' ! ! 15) '\$ " &.

6 \* 7 ! !! \$ ' + # ! '\$ " &89 '! ! 4

\$' " &.\$' " &89

8 !! 7 ' :5#1 \$ 1' " #' ! \$ \* \$\$ ' # \$ # # &&' " 9

! ' 7!&\* \$ % ' ! ! 15) # &\* 5 ( ' \$ 7' !'##5 2 ' + # ! '\$ " ! 89() ! #1 ! !

& \$ ! \$ ' 1' '\$' (\* \$ - &\* \* \$ !& \$ ' # \$ " "%( '\* 7 4

\$' " &.\$' " ! 8 (; 9



Remarque : nous allons décrire l'automatisme d'importation dans le IP8 avec la fonction `sqrt()` de la bibliothèque `math` et on importe avec `from math import sqrt` puis on utilise en tapant `math.sqrt( )`



### Exercice 4 : un petit jeu

Il s'agit d'un jeu de hasard :

- la machine choisit un nombre mystère entre 0 et 100
- un premier joueur entre un nombre dans la machine
- un second joueur entre un autre nombre dans la machine
- la machine dit lequel des deux a été le plus proche du nombre mystère



### Une opération utile

< '\* 7 "" ' # ! \$ # ! ! ' 4

; =

> =

? =

@ =

A \*%\$' \$ = " \$ " ' " ! ! ! " & \$ \* '\$ ' \$ .

/'\$ - &\* ( ; × B C > × C ? × C #.

= \$5 ! \* \$ ' ! \$ ! & \$ " ! ! ! \* '\$ ' \$ .

/'\$ - &\* 4

' =

" \$ B ! ' \* ! \$ " \$ ' ! ' ! & \* ! \$.



## Exercice 5 : les années issextiles

Tous les ans sans doute une année fait en l'an 2000.

La plupart des années calendaires font 365 jours tandis que certaines (dites \*issextiles) en font 366.

L'année sera \*issextile :

1) si l'année est divisible par 4 et non divisible par 100

ou

2) si l'année est divisible par 400

, sinon l'année n'est pas \*issextile (elle a 365 jours).

Écrivez un programme qui demande à un utilisateur de saisir une année puis qui dit si elle est \*issextile ou pas.



### And et Or

Dans ce programme, on utilise les opérateurs `and` et `or`.

Exemple : `(a > 0 and b > 0) or (a < 0 and b < 0)`

!8' "!!! \*\$ FBB9 8!' "!!! \*\$ F ' "!!! \*\$ BB9

\$

G#1 \$ % ! - !

!

G#1 \$ % ! - !

! !



## Exercice 5 bis : les années issextiles

Écrivez un programme de l'exercice précédent en utilisant `and` et `or`.



## Exercice 6 : voyage scolaire

Mme Lydie organise un voyage scolaire en Espagne.

Seuls les élèves qui sont déjà partis peuvent y aller :

- tous les élèves de seconde qui pratiquent l'espagnol
- tous les élèves de première désireux de voyager

Écrivez un programme qui demande à un élève quelle est sa classe, puis qui lui demande s'il pratique l'espagnol puis lui dit s'il peut aller au voyage.



## Exercice bonus : le jour de votre naissance

Voici une méthode pour déterminer le jour de la semaine d'une date donnée entre 1900 et 2099 :

– Déterminer le code du mois avec le tableau de correspondance suivant :

Janvier	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre	Novembre	Décembre
0	3	3	6	1	4	6	2	5	0	3	5

– Ajouter le nombre formé par les « deux chiffres de droite » de l'année, le quart de ce nombre (trouqué à l'entier inférieur), le numéro du jour et le code du mois.

– Si la date est après 2000, enlever 1 au résultat.

– Si l'année est bissextile et si la date est avant le 1<sup>er</sup> mars, ajouter 1 au résultat.

– Calculer le reste de la division du résultat obtenu par 7.

– Convertir ce reste en jour avec la correspondance : 0 → dimanche ; 1 → lundi ; 2 → mardi ;

*Cette méthode permet par exemple de savoir quel jour était le 6 juin 1944 (année bissextile).*

*Le code du mois est 4, donc le calcul est :  $44 + 11 + 6 + 4 = 65$ .*

*Le reste de la division de 65 par 7 étant 2, ce jour était un mardi.*

Il y a un programme qui demande à un utilisateur sa date de naissance et lui dit quel jour de la semaine il est né.



Il y a trois sortes de **tests** :

- le test A, si ( alors B s'exécute ainsi :

```
if condition :
    instruction
    instruction>
    et #
suite du programme
```

- le test A, si ( alors ( sinon ( B s'exécute ainsi :

```
if condition :
    instruction
    instruction>
    et #
else :
    instructionNalternative
    instructionNalternative>
    et #
suite du programme
```

- le test A, si ( alors ( sinon si ( sinon ( B s'exécute ainsi :

```
if condition :
    instruction
    instruction>
    et #
elif autreCondition :
    instructionNalternative
    instructionNalternative>
    et #
else :
    instructionNalternative*is
    instructionNalternative*>is
    et #
suite du programme
```

On utilise les tabulations pour décaler les instructions qui doivent être exécutées en cas de réalisation de la condition. On dit que ces instructions sont **indentées**.

Les **conditions** s'écrivent : *variable* *symbole* *variable ou nombre ou texte ou calcul*

OP *symbole* est un des symboles suivants :

4 7 4= 7= == 6= in