

Traitements conditionnels

La syntaxe d'un traitement conditionnel est la suivante :

```
SI <condition> ALORS  
    <instructions>  
FIN SI
```

*Le bloc <instructions> est **indenté** c.à.d. **décalé** par rapport à la ligne de départ.*

Les <instructions> ne sont exécutées que si <condition> est vérifiée

Exemple d'algorithme :

```
SI A = 0 ALORS  
    écrire "La valeur de la variable A est nulle."  
FIN SI
```

Traduction de l'algorithme en python:

```
#entrez une valeur  
a=int(input('entrez la valeur de A : '))  
#Verifiez le contenu de a  
if a==0 :  
    print('la valeur de A est nulle')  
|
```

*à tester avec $a=0$ et $a=5$.
Commentez les deux cas.*

Traitements conditionnels

Commentaire:

Si la variable A , au moment du test, a une valeur nulle, alors l'instruction Afficher "La valeur de la variable A est nulle." est exécutée,

```
>>>  
entrez la valeur de A : 0  
la valeur de A est nulle  
>>> |
```

sinon, elle est ignorée et le programme s'arrête.

```
>>>  
entrez la valeur de A : 5  
>>> |
```

Traitements conditionnels étendus

La syntaxe d'un traitement conditionnel étendu est la suivante :

```
SI <condition> ALORS
  <instructions>
SINON
  <autres instructions>
FIN SI
```

Ne pas oublier l'indentation!

Si <condition> est vérifiée, les <instructions> sont exécutées.

Dans le cas contraire, ce sont les <autres instructions> qui sont exécutées.

Traduire l'algorithme ci-dessous en python:

```
SI A = 0 ALORS
  écrire "La valeur de la variable A est nulle."
SINON
  écrire "La valeur de la variable A est non nulle."
```

*En python, la syntaxe du SINON est **else**:*

Programmes à réaliser:

Traduisez l'algorithme suivant en python:

Algorithme Valeurs distinctes

Variables :

entiers : a, b

DEBUT

écrire "Saisissez deux valeurs entières"

lire a, b

SI a = b ALORS

écrire "Vous avez saisi deux fois la même valeur, à savoir ", a, "."

SINON

Ecrire "Vous avez saisi deux valeurs différentes, ", a, " et ", b, "."

FIN SI

FIN

Programmes à réaliser:

Traduisez l'algorithme suivant en python: *imbrication des si*

```
Algorithme positif,négatif
Variables :
  entiers : a, b
DEBUT
  écrire "Saisissez deux entiers relatifs"
  lire a, b
SI a < 0 ALORS
  SI b < 0 ALORS
    écrire "a et b sont négatifs"
  SINON
    écrire "a est négatif, b est positif"
  FIN SI
SINON
  SI b < 0 ALORS
    écrire "b est négatif, a est positif"
  SINON
    écrire "a et b sont positifs"
  FIN SI
FIN SI
```

*Montrer la partie du programme
qui sera exécutée si $a > 0$.*

Connecteurs logiques

Les connecteurs logiques permettent de d'évaluer des conditions plus complexes:

Le connecteur **et** :

```
SI <condition1> et <condition2> ALORS  
  <instructions>  
SINON  
  <autres instructions>  
FIN SI
```

*En python:
la syntaxe du **et** est **and**:*

Les <instructions> sont exécutées si <condition1> **et** <condition2> sont **vérifiées simultanément**.

Le connecteur **ou** :

```
SI <condition1> ou <condition2> ALORS  
  <instructions>  
SINON  
  <autres instructions>  
FIN SI
```

*En python:
la syntaxe du **ou** est **or**:*

Les <instructions> sont exécutées **si une au moins** des deux conditions est **vérifiée** .

Programmes à réaliser:

Traduisez l'algorithme qui demande à l'utilisateur de saisir deux valeurs, et qui lui dit si le produit de ces deux valeurs est positif ou négatif sans en calculer le produit.

Algorithme Signe du produit

Variables :

entiers : a, b

DEBUT

écrire "Saisissez deux entiers relatifs"

lire a, b

SI (a \leq 0 et b \leq 0) ou (a \geq 0 et b \geq 0)

ALORS

écrire "positif ou nul"

SINON

écrire "négatif"

FIN SI

écrire "."

FIN

Programmes à réaliser:

Ecrivez un algorithme puis un programme qui donne le maximum de 2 nombres entrés au clavier

Ecrivez un algorithme puis un programme qui donne le maximum de 3 nombres entrés au clavier

Ecrivez un algorithme puis un programme qui donne la mention d'un étudiant en fonction de sa note entrée au clavier.

Les étudiants ayant passé l'examen du module U3 en session de Juin ont été classés selon leurs notes en trois catégories :

- pour une note inférieure strictement à 5, l'étudiant est éliminé,
- pour une note supérieure ou égale à 5 et inférieure strictement à 10, l'étudiant passe la session de rattrapage,
- pour une note supérieure ou égale à 10, l'étudiant valide le module

Ecrivez un algorithme puis un programme qui demande à l'utilisateur d'entrer la note du module, puis affiche la situation de l'étudiant selon sa note (l'utilisateur entre une note valide entre 0 et 20).

Le module random

Le **module** random permet de générer des nombres aléatoires.
Pour l'utiliser, il faut l'importer.

randrange(n): permet d'obtenir un nombre entier compris entre 0(inclus) et n(exclus)

à tester en mode console

randint(a,b): permet d'obtenir un nombre entier compris dans [a,b]

à tester en mode console

uniform(a,b): permet d'obtenir un flottant compris dans [a,b]

à tester en mode console

Programmes à réaliser:

Jeu: feuille , pierre, ciseaux (jouer contre l'ordi)

Jeu: lancer de dé.(miser sur une valeur)

Jeu: blackjack : (3 lancer de dé sans dépasser 21. on joue contre l'ordi)