

# Résolution d'un problème :



ALGORITHME Nom  
DEBUT

Les entrées  
Traitements  
Les sorties

FIN

Déclaration des objets

Objet	Type/nature

### Sous-chaîne

- `ch[:]` → chaîne entière.
- `[4:]` → de l'index 4 jusqu'au dernier index.
- `[ : 8 ]` → à partir de l'index 0 jusqu'à l'index 7.
- `[ 2 : 7 ]` → à partir de l'index 2 jusqu'à l'index 6.

Tableau de chaînes :

```
T=array(['str']*25)
```

Tableau de caractères :

```
T=array(['str']*25)
```

## 1. Les opérations simples

a. L'opération d'entrée :

```
Ecrire("Donner ch") , Lire (ch)
```

```
ch=input('donner ch=')
```

```
Ecrire("Donner entier=") , Lire (n)
```

```
n=int(input('Entier='))
```

```
Ecrire("Donner réel=") , Lire (x)
```

```
x=float(input('Réel='))
```

b. L'opération de sortie

```
Ecrire ("Message", x, x+10)
```

```
print('Message', x, x+10)
```

c. L'opération d'affectation

```
x ← 15    x ← x+1    z ← "Ok"
```

```
x=15
x=x+1
z='Ok'
```

Div	//	7 // 2=3
Mod	%	7 % 2=1

## 2. Les types de données

Entier	int	15
Réel	float	12.3
Booléen	bool	True/False
Caractère	----	"A" / 'A'
Chaîne	str	"Ok" / 'Ok'

## 3. Opérations de base

Opérateurs: +, -, \*, /, %, //

Opérateurs de comparaisons :

=, !=, >, >=, <, <=

Opérateurs logiques: or, and, not

```
print ('---', x, '\n', '***')
\n : retour à la ligne
```

Tableau (array) d'entier, réel, chaîne

```
>>>from numpy import array
>>> a=array([4,3,6])
>>>T=array([float()*n])
```

```
>>>T[0]=13
```

```
>>>a[0];a[2]
```

## 3. Les structures de contrôle conditionnelles

• **Conditionnelle simple**

Initialisation  
Si Condition Alors  
    Traitement1  
    Sinon  
    Traitement2  
FinSi

```
if cond :
    Tr1
else :
    Tr2
```

• **Conditionnelle généralisée**

Initialisation  
Si Condition 1 Alors Tr1  
Sinon Si Condition 2 Alors Tr 2  
Sinon .....  
Sinon Si condition N-1 Alors Tr N-1  
Sinon Traitement N  
FinSi

```
if cond1 :
    Tr1
elif cond2:
    Tr2
else :
    Tr3
```

## 4. Les structures de contrôle itératives

a. La structure itérative complète :

Initialisation  
Pour compteur de Début à Fin (pas) Faire  
    Traitement  
Fin Pour

Algo	Python
x ∈ [0..5]	x in range(6)
x ∈ [2..8]	x in range(2,9)
	2 <= x <= 8

En Python :  
for compteur in range(BI,BS,pas):  
    Traitement

b. La structure de contrôle itérative à condition d'arrêt

• 1ère formulation

Répéter  
    Traitement  
Jusqu'à condition(s)

```
En Python :
valide=False
while valide==False:
    Traitement
    valide=(cond(s))
```

• Deuxième formulation

Initialisation  
Tant que Cond(s)  
Faire  
    Traitement  
Fin Tantque

```
En Python :
while condition :
    Traitement
```

## Les fonctions sur les types numériques :

Algorithme	Python	Rôle	Exemple
Arrondi(x)	round(x)	Retourne l'entier le plus proche de la valeur x	round(4.3)=4 round(4.5)=5
racineCarré(x)	sqrt(x)	Retourne la racine carrée d'un nombre x positif (from math import sqrt)	sqrt(2)=1.4142
Aléa(vi,vf)	randint(vi,vf)	Donne un entier aléatoire de l'intervalle [vi, vf]. (from random import *)	Aléa(1,10)
Ent(x)	trunc(x)	Retourne la partie entière de x.	trunc(3.5)
abs(x)	abs(x)	retourne la valeur absolue de x	abs(-5)=5

## Les fonctions sur le type chaînes de caractères :

Algorithme	Python	Rôle	Exemple
long(ch)	len(ch)	Retourne le nombre de caractères de la chaîne ch	len('Bac')=3
pos(ch1,ch2)	ch2.find(ch1)	Retourne la première position de la chaîne ch1 dans la chaîne ch2 (retourne -1 si pas trouvé)	'Bac'.find('ac') 1
convch(x)	str(x)	Retourne la conversion d'un nombre x en une chaîne de caractères.	Str(10) → '10'
Estnum(ch)	ch.isdigit() ch.isdecimal()	Retourne Vrai si la chaîne ch est convertible en une valeur numérique, elle retourne Faux sinon	'123'.isdigit() True
Valeur(ch)	int(ch) float(ch)	Conversion d'une chaîne ch en une valeur numérique, si c'est possible.	int('4') ou float('4')
sous_chaîne(ch,d,f)	ch[d:f]	Retourne une partie de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).	ch='bac' ch[1:3] 'ac'
Effacer(ch,d,f)	ch=ch[0:d]+ ch[f:len(ch)]	Efface des caractères de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).	ch='bonjour' ch=ch[0:1]+ ch[3:7] 'bjour'
majus(ch)	ch.upper()	Convertit la chaîne ch en majuscules.	'bac'.upper() BAC

## Les fonctions sur le type caractères:

Algorithme	Python	Rôle	Exemple
ord(c)	ord(c)	Retourne le code ASCII du caractère c	ord("A")=65
chr(d)	chr(d)	Retourne le caractère dont le code ASCII est d.	chr(65)= 'A'

Algorithme	Python	Algorithme	Python
x ∈ [ "A".."Z" ]	ord(x) in range(ord("A"), ord("Z")+1)	x ∈ {1,5,6}	x in {1,5,6}
	"A" <= x <= "Z"	x ∈ { "a", "e", "i", "o", "y", "u" }	x in { 'a', 'e', 'i', 'o', 'y', 'u' }

Algorithme	Python
'A'+ 'B' donne 'AB'	
3* 'A' donne 'AAA'	

Algorithme	Python
Pour c de 0 à long(ch)-1 (pas=1) faire Ecrire(ch[c])	for c in range(len(ch)): print(ch[c])