

# 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.
- [ 4 : -1 ] → de l'index 4 à l'avant-dernier index.

## 1. Les opérations simples

a. L'opération d'entrée :

```
Entree("Donner ch"), Lire (ch)
ch=input('donner ch')
```

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

```
n=int(input("Donner entier="))
```

Ecrire("Donner réel="), Lire (x)

```
x=float(input("Donner 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

Tableau (array) d'entier, réel, booléen

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

>>>T[0]=13.5
>>>a[0];a[2]
```

## 3. Les structures de contrôle conditionnelles

<p>• <b>Conditionnelle simple</b></p> <p>Initialisation</p> <p><b>Si</b> Condition <b>Alors</b> Traitement1</p> <p>    <b>else :</b> Traitement2</p> <p>    <b>Sinon</b> Traitement2</p> <p><b>FinSi</b></p>	<p><b>if</b> condition : Traitement1</p> <p><b>else :</b> Traitement2</p>	<p>• <b>Conditionnelle généralisée</b></p> <p>Initialisation</p> <p><b>Si</b> Condition 1 <b>Alors</b> Traitement 1</p> <p>    <b>Sinon Si</b> Condition 2 <b>Alors</b> Traitement 2</p> <p>    <b>Sinon</b> ..... Traitement N-1</p> <p>    <b>Sinon Si</b> condition N-1 <b>Alors</b> Traitement N-1</p> <p>    <b>Sinon</b> Traitement N</p> <p><b>FinSi</b></p>	<p><b>if</b> condition1 : Traitement1</p> <p><b>elif</b> condition2 : Traitement2</p> <p><b>else :</b> Traitement3</p>
--	---	---	--

## 4. Les structures de contrôle itératives

<p>a. La structure de contrôle itérative complète :</p> <p>Initialisation</p> <p><b>Pour</b> compteur de Début à Fin (pas) <b>Faire</b> Traitement</p> <p><b>Fin Pour</b></p> <table border="1"> <tr> <th>Algo</th> <th>Python</th> </tr> <tr> <td>x ∈ [0..5]</td> <td>x in range(6)</td> </tr> <tr> <td>x ∈ [2..8]</td> <td>x in range(2,9) 2&lt;=x&lt;=8</td> </tr> </table> <p>En Python :</p> <pre>for compteur in range(BI,BS,pas):     Traitement</pre>	Algo	Python	x ∈ [0..5]	x in range(6)	x ∈ [2..8]	x in range(2,9) 2<=x<=8	<p>b. La structure de contrôle itérative à condition d'arrêt</p> <p>• <b>1ère formulation</b></p> <p><b>Répéter</b> Traitement</p> <p><b>Jusqu'à</b> condition(s)</p> <p>En Python :</p> <pre># while not(condition): while True:     Traitement     if condition: break</pre>	<p>• <b>Deuxième formulation</b></p> <p>Initialisation</p> <p><b>Tant que</b> Condition(s) <b>Faire</b> Traitement</p> <p><b>Fin Tantque</b></p> <p>En Python :</p> <pre>while condition:     Traitement</pre>
Algo	Python							
x ∈ [0..5]	x in range(6)							
x ∈ [2..8]	x in range(2,9) 2<=x<=8							

## Les fonctions sur le type 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	sqrt(2)=1.4142
Aléa(vi,vf)	randint(vi,vf)	Retourne un entier aléatoire de l'intervalle [vi, vf].	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)	Retourne la conversion d'une chaîne ch en une valeur numérique, si c'est possible.	int('4') ou float('4')
sous_chaine(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	Algorithme	Python
'A'+ 'B' donne 'AB'		Pour c de 0 à long(ch)-1 (pas=1) faire	for c in range(len(ch)):
3* 'A' donne 'AAA'		Ecrire(ch[c])	print(ch[c])