

1. Le type chaîne de caractère (str):

a. Définition :

C'est une succession de n caractère (lettre, symbole, chiffre) avec $n \geq 0$.

□ Remarque :

- Si $n = 0$ alors la chaîne est dite vide ("" : chaîne vide).
- Les valeurs chaîne de caractères sont définies entre guillemets

b. Manipulation de chaîne de caractère :

On peut accéder en lecture au $i^{\text{ème}}$ caractère d'une chaîne Ch en utilisant la notation **CH[i]** avec $0 \leq i \leq \text{Long}(Ch)-1$.

✂ **Exemple :** Soit Ch $\leftarrow$ "Bonjour papa"

- Ch [0] = "B"
- Ch [1] = "o"
- Ch [:3] = "Bon" renvoie les trois premiers caractères
- Ch [3:] = "jour papa" renvoie Ch on élimine les trois premiers caractères
- Ch [3:7] = "jour" renvoie une sous chaîne de Ch de la position 3 jusqu'à la position 7-1=6

c. Fonctions prédéfinis sur les chaînes de caractère:

Algorithme	Python	Role	exemple
ch[n]	ch[n]	ch[n] prendre un caractère d'indice n(On commence par 0)	ch = 'tester' y=ch[1] y='e' y=ch[0] y='t'
sous-chaîne(ch,debut,fin)	ch[debut:fin]	ch[debut:fin] prendre une chaîne de l'indice debut à l'indice fin-1	ch = 'tester' y=ch[1:3] y='es' y=ch[1:] y='ester'
long(ch).	Lo=len(ch)	len() :Déterminer la longueur (c'est-à-dire le nombre de caractères) d'une chaîne.	ch='test' n=len(ch) n=4 y=ch[0:len(ch)] y='test'
pos(ch1,ch2)	P=ch2.find(ch1)	find(chaîne) : cherche la position d'une chaîne ch1 dans la chaîne ch2, en partant du début(retourne -1 si ch1 n'existe pas dans ch2).	ch1 = 'info' ch2 = 'fo' p=ch1.find(ch2) p=2 ch1 = 'info' ch2 = 'mo' p=ch1.find(ch2) p=-1

majus (Ch)	ChM = Ch.upper ()	Retourne la chaîne ChM représentant la conversion en Majuscule de la chaîne Ch .	Ch = "Bonjour" ChM = Ch.upper () ChM = 'BONJOUR'
convch(n)	str(n)	Convertir un entier en une chaîne	n=25 Ch=str(n) Ch= '25'
valeur (ch)	int(ch)	Convertir un entier en une chaîne	ch= '25' n=int(ch) n =25

- **N.B. :** On utilise l'opérateur + pour concaténer deux chaînes.

Application 1 :

Donner le resultat d'affichage de chacune des instructions suivantes :

a)

```
chain = '123456789'
```

```
print(chain[0])
```

```
.....
```

```
print(chain[1:3])
```

```
....
```

```
print(chain[0:2])
```

```
.....
```

```
print(chain[1:len(chain)])
```

```
.....
```

Application 2 :

b)

```
ch1='informatique'
```

```
ch2=ch1[0 :4]
```

```
ch3='2025'
```

```
ch4=ch2+ch3
```

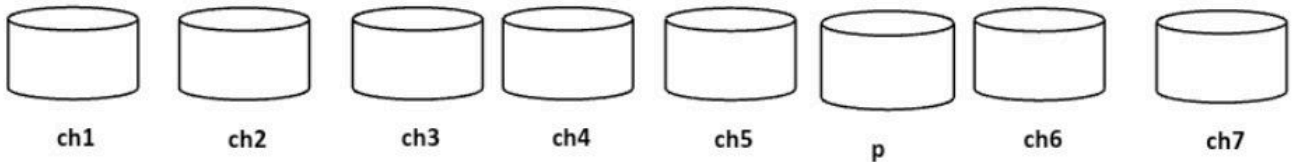
```
ch5='l'+ch1[1:5]
```

```
p= ch1.find('f')
```

```
ch6=ch1[1:p]
```

```
ch7=ch1[p+1:5]
```

Donner le contenu de ch1, ch2, ch3, ch4, ch5, p, ch6 et ch7



Page 20

Application 3 :

En utilisant les chaînes Ch1, Ch2 et Ch3 donner pour chaque instruction la valeur de X Ainsi qu'une implémentation possible en Python :

Ch1="Algorithmique"		Ch2="Python"	Ch3="502"
	Instruction algorithmique	Valeur de X	Python
1	X ← long(Ch2)		
2	X ← long(Ch2+Ch3)		
3	X ← pos("Go",Ch1)		
4	X ← pos("i", Ch1)		
5	X ← sous_chaine(Ch2, 2,long(Ch2))		

Application 4 :

Compléter le tableau suivant et corriger les instructions non valide:

opération	valide/non valide	Résultat de S
a=5 b=2 s=a+b print(s)		
a='5' b='2' s=a+b print(s)		
a=5 b='2' s=a+b print(s)		

Correction Application 3

Ch1="Algorithmique" Ch3="502"		Ch2="Python"	
	Instruction algorithmique	Valeur de X	Python
1	$X \leftarrow \text{long}(\text{Ch2})$	6	<code>X = len(Ch2)</code>
2	$X \leftarrow \text{long}(\text{Ch2}+\text{Ch3})$	9	<code>X = len(Ch2+Ch3)</code>
3	$X \leftarrow \text{pos}(\text{"Go"}, \text{Ch1})$	-1	<code>X = Ch1.find("Go")</code>
4	$X \leftarrow \text{pos}(\text{"i"}, \text{Ch1})$	5	<code>X = Ch1.find("i")</code>
5	$X \leftarrow \text{sous_chaîne}(\text{Ch2}, 2, \text{long}(\text{Ch2}))$	thon	<code>X=Ch2[2 :len(Ch2)]</code>

Correction Application 4

opération	valide/non valide	Résultat de S
a=5 b=2 s=a+b	Valide	7
a='5' b='2' s=a+b	Valide	"52"
a=5 b='2' s=a+b	Non valide	Solution on affiche 7 : a=5 b='2' s=a+int(b)