

Opérations sur les chaînes de caractères ou les tableaux

Rappelons qu'une chaîne de caractères est une suite de caractères ASCII codés sur un octet (256 caractères codables différents donc) et est du type **string**. Chaque caractère repéré par sa position dans la chaîne appelée son rang et commençant à 0.

Ainsi dans la chaîne "Dick" la lettre D a pour rang 0, la lettre i pour rang 1, la lettre c pour rang 2 et la lettre k pour rang 3. La longueur de la chaîne est 4 et peut s'obtenir avec la commande `len("Dick")`.

Il est possible de créer une chaîne de caractères sans aucun caractère. C'est la chaîne vide "".

Un tableau est une liste d'éléments pouvant être du même type, exemple des entiers, ou bien de type différents. C'est une variable de type **array**.

1) Concaténation :

Elle consiste à créer une seule chaîne de caractères (ou tableau) à partir de plusieurs chaînes (ou tableaux) et elle se fait en utilisant le signe +

Exemple avec des chaînes de caractères :

```
Nom="Tatür"
```

```
Prenom = "Dick"
```

```
Identite = Prenom + " " + Nom
```

```
print(Identite)
```

2) Extraction d'une sous-chaîne

Voici une liste de commande permettant d'extraire des caractères de rang donnés dans une chaîne de caractères désignée par la variable chaîne :

`chaîne[:n]` extrait tous les caractères situés avant le rang n donc du rang 0 au rang n-1

`chaîne[n:]` extrait tous les caractères situés après le rang n, rang n compris

`chaîne[n:m]` extrait tous les caractères situés entre le rang n compris et le rang m non compris

`chaîne[-1]` extrait le dernier caractère, `chaîne[-2]` l'avant dernier etc...

Des commandes analogues existent pour extraire des éléments d'un tableau

Exemples :

Voilà un programme et ce qu'il affiche en l'exécutant

```
Prenom = "Isidore"
```

```
print(Prenom[:3])
```

```
print(Prenom[3:])
```

```
print(Prenom[3:5])
print(Prenom[-1])
print(Prenom[-2])
print(Prenom[-3:])
```

ce programme renvoie :

```
Isi
dore
do
e
r
ore
```

Voici un programme analogue mais traitant un tableau déclaré par la variable tab :

```
tab=[10,20,30,40,50,60,70,80]
print(tab[:3])
print(tab[3:])
print(tab[3:5])
print(tab[-1])
print(tab[-2])
print(tab[-3:])
```

et voici ce qu'il renvoie

```
[10, 20, 30]
[40, 50, 60, 70, 80]
[40, 50]
80
70
[60, 70, 80]
```

3) Ne pas confondre l'usage du signe + pour les chaînes de caractères, les nombres entiers ou flottants et les tableaux.

Le même signe + est utilisé pour différents usages mais ne produit pas forcément les mêmes effets.

Exemple :

Voici trois commandes qui se ressemblent :

```
print(4+5)
```

```
print("4"+"5")
```

```
print([4]+[5])
```

Pourtant l'exécution de ces commandes donne trois versions différentes :

```
9  
45  
[4, 5]
```

Dans la première commande, il s'agit d'une addition, dans la seconde une concaténation de deux chaînes de caractères réduites à un seul caractère et dans la troisième, une concaténation de deux tableaux réduits à un seul élément.