

# Les chaînes de caractères en Python

---

## 1. Les chaînes de caractères : un type défini en Python.

---

```
1 >>> texte="abcdeFGHI"
2 >>> type(texte) # renvoie le type de la variable texte
3 <class 'str'>
4 >>> type(texte)=='str'
5 False
6 >>> type(texte)==str
7 True
```

## 2. Accéder aux caractères par leurs indices.

---

```
1 >>> texte[0] # renvoie le caractère d'indice 0
2 'a'
3 >>> len(texte) # renvoie la longueur de la chaîne de caractère
4 9
5 >>> texte[len(texte)]
6 Traceback (most recent call last):
7   File "<pyshe11>", line 1, in <module>
8 IndexError: string index out of range
9 >>> texte[len(texte)-1] # renvoie le dernier caractère
10 'I'
11 >>> texte[-1] # renvoie le dernier caractère
12 'I'
```

## 3. Découper des chaînes de caractères.

---

```
1 >>> texte="abcdeFGHI"
2 >>> texte[:5] # renvoie les caractères d'indices 0 à 4
3 'abcde'
4 >>> texte[5:] # renvoie les caractères d'indices 5 au dernier
5 'FGHI'
6 >>> texte[2:6] # renvoie les caractères d'indices 2 à 5
7 'cdeF'
```

## 4. Déterminer si un caractère est présent dans une chaîne de caractères

---

```
1 >>> texte="AADBBNNAC"
2 >>> "a" in texte
3 False
4 >>> "A" in texte
5 True
6 >>> texte.count("A") # renvoie le nombre d'occurrence du caractère dans la chaîne
7 3
8 >>> texte.count("n")
9 0
10 >>> texte.find("B") # renvoie le premier indice correspondant au caractère
11 3
12 >>> texte.find("U") # renvoie -1 si le caractère n'est pas présent
13 -1
```

## 5. Des méthodes qui peuvent être utiles

---

```
1 >>> texte="AZerty"
2 >>> texte.lower()
3 'azerty'
4 >>> texte.upper()
5 'AZERTY'
6 >>> texte
7 'AZerty'
```

## 6. Rappel : parcourir une chaîne de caractères

---

```
1 >>> texte="AZerty"
2 >>> for caract in texte:
3     print(caract)
4
5 A
6 Z
7 e
8 r
9 t
10 y
```

## Exercices

---

Compléter les fonctions suivantes pour quelles correspondent à leur docstring :

### Exercice 1

---

```
1 def creation_mail(nom, prenom):
2     '''
3     renvoie la chaîne de caractère prenom_nom@nsi_mail.fr
4     nom, prenom : str en majuscules ou minuscules
5     return : str
6     >>>creation_mail("Python","Bob")
7     "bob_python@nsi_mail.fr"
8     '''
```

## Exercice 2

---

```
1 def bonne_adresse_mail(adresse):
2     '''
3     renvoie True si l'adresse contient le caractère @ et un . après celui-ci,
4     renvoie False sinon
5     adresse : str
6     return : bool
7     >>>bonne_adresse_mail("adresse")
8     False
9     >>>bonne_adresse_mail("adresse@nsi.fr")
10    True
11    >>>bonne_adresse_mail("nom.prenom@nsifr")
12    False
13    >>>bonne_adresse_mail("nom.prenom@nsi.com")
14    True
15    '''
```

## Exercice 3

---

```
1 def conversion_bin(nombre):
2     '''
3     convertit un entier naturel écrit en décimal en binaire
4     nombre : int
5     return : str
6
7     >>>conversion_bin(100)
8     '1100100'
9     '''
```