# Manipulation de chaînes de caractères

Totalement hors programme, la manipulation des chaînes de caractères (type string, str) peut être pourtant bien pratique pour gérer l'affichage de certaines sorties de programmes, ou bien pour faire du *text parsing* dans des fichiers textes.

### i À savoir

- On écrit les chaînes de caractères avec des guillemets simples 'NSI' ou doubles "NSI". Peu importe, sauf si la chaîne de caractères comporte une apostrophe ', auquel cas il est obligatoire d'utiliser les doubles.
- Une chaîne de caractères est un objet **iterable**: on peut donc le parcourir, récupérer sa longueur et ses caractères grâce à leur indice.

## & Script Python

```
>>> mot = "Guido"
>>> len(mot)
5
>>> mot[2]
'i'
>>> for l in mot:
    print(l)

G
u
i
d
o
>>>>
```

• Une chaîne de caractères est un objet **non mutable** : on ne peut pas le modifier.

#### **&** Script Python

```
>>> mot = "Guido"
>>> mot[4] = 'e'
Traceback (most recent call last):
   File "<pyshell>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>>
```

#### **6** Concaténation

La concaténation de plusieurs chaînes de caractères consiste tout simplement à les mettre bout à bout. Elle se fait en Python avec l'opérateur +. On peut également multiplier une chaîne de caractères par un entier<sup>1</sup>.

#### **%** Script Python

#### Conversions

• On peut convertir un entier, un flottant en chaîne de caractères avec la fonction str:

#### **%** Script Python

```
>>> str(2)
'2'
>>> str(2.0)
'2.0'
```

• On peut convertir une chaîne de caractères en liste avec la fonction list:

#### **%** Script Python

```
>>> list("Python")
['P', 'y', 't', 'h', 'o', 'n']
```

### **b** split **and** join

• On peut découper une chaîne de caractères avec la méthode split. Sans paramètre, elle découpe sur le caractère espace. Avec un paramètre de type str, elle découpe selon ce caractère. Dans les deux cas, le caractère de découpe est supprimé. Elle retourne une liste.

```
% Script Python
```

```
>>> p = "Un ordinateur, c'est complètement con."
>>> p.split()
['Un', 'ordinateur,', "c'est", 'complètement', 'con.']
>>> m = 'Abracadabra'
>>> m.split('a')
['Abr', 'c', 'd', 'br', '']
```

• Dans l'autre sens, on peut fusionner avec la méthode join les élements d'une liste dont tous les éléments sont de type str, avec un caractère d'insertion entre les éléments.

### **%** Script Python

```
>>> l = ['P', 'y', 't', 'h', 'o', 'n']
>>> "".join(l)
'Python'
>>> " ".join(["NSI", "c'est", "de", "l'eau"])
"NSI, c'est de l'eau"
```

1. puisque la multiplication par un entier n'est qu'un raccourci pour l'addition de mêmes termes. ←