

# Programmation en langage Python

Séance n°1 - Vendredi 3 novembre 2023

Découverte du langage Python - Version 3.1X

La notion de chaîne de caractères : str Objets de type str (chaîne de caractères)

```
[1]: mot = "Hello"
```

```
[2]: mot
```

```
[2]: 'Hello'
```

Pour déterminer le type d'un objet, on dispose de l'instruction `type()`

```
[6]: type(mot)
```

```
[6]: str
```

On constate que l'on a créé un objet de type `str`. Nous découvrirons d'autres types d'objets comme `int`, `float`, `list`, `tuple`, etc. Un objet peut être manipulé par des fonctions et des méthodes.

La fonction `len()`. Cette fonction renvoie la longueur d'un objet. La longueur d'une chaîne de caractères est le nombre de caractères contenus dans la chaîne.

```
[7]: len(mot)
```

```
[7]: 5
```

La fonction `id()`. Celle-ci renvoie l'identificateur associé à l'objet créé dans la mémoire de l'ordinateur.

```
[8]: id(mot)
```

```
[8]: 4445546568
```

Les méthodes `upper()`, `lower()` et `capitalize()`. Celles-ci mettent un objet de type `str` en majuscules, en minuscules ou la première lettre en majuscule.

```
[9]: mot.upper()
```

```
[9]: 'HELLO'
```

```
[10]: mot.lower()
```

```
[10]: 'hello'
```

```
[12]: "mathématiques".capitalize()
```

```
[12]: 'Mathématiques'
```

Pour mémoriser ou sauvegarder une chaîne modifiée à l'aide d'une méthode, il faut utiliser une nouvelle variable.

```
[13]: Mot_majuscule = mot.upper()
```

```
[14]: Mot_majuscule
```

```
[14]: 'HELLO'
```

Un objet de type str peut être 'slicé'. Il s'agit d'une technique de découpage de chaîne et d'extraction d'informations. En tapant chaîne[0] et chaîne[-1], on obtient respectivement les première et dernière lettres de la chaîne.

```
[16]: mot[0]
```

```
[16]: 'H'
```

```
[17]: mot[-1]
```

```
[17]: 'o'
```

Le slicing permet d'extraire aussi n'importe quelle portion d'une chaîne de caractères.

```
[18]: mot
```

```
[18]: 'Hello'
```

```
[19]: mot[1:3]
```

```
[19]: 'el'
```

```
[20]: mot[1:4]
```

```
[20]: 'ell'
```

```
[21]: mot[1:5]
```

```
[21]: 'ello'
```

```
[26]: mot_complexe = "eschatologie"
```

```
[27]: mot_complexe
```

```
[27]: 'eschatologie'
```

```
[28]: long = len(mot_complexe)
      mot_complexe[4:long]
```

```
[28]: 'atologie'
```

On peut rechercher si un caractère se trouve dans un mot. L'interpréteur Python renvoie True ou False selon le résultat.

```
[32]: 'h' in mot
```

```
[32]: False
```

```
[33]: 'H' in mot
```

```
[33]: True
```

On peut remplacer une sous-chaîne dans une chaîne par une autre sous-chaîne.

```
[35]: mot.replace('l', 'w')
```

```
[35]: 'Hewwo'
```

Lorsqu'on veut élaguer une chaîne de caractères, on utilise la méthode `strip()` qui élimine les espaces inutiles situés en début et fin de chaîne.

```
[36]: mot = "   utile   "
```

```
[37]: mot
```

```
[37]: '   utile   '
```

```
[38]: mot.strip()
```

```
[38]: 'utile'
```

Une chaîne de caractère est un itérable.

```
[39]: for c in "Hello":  
      print(c)
```

```
H  
e  
l  
l  
o
```

Une chaîne de caractères peut être composée de plusieurs mots.

```
[29]: phrase = "Apprendre à programmer en langage Python est facile"
```

```
[30]: phrase
```

```
[30]: 'Apprendre à programmer en langage Python est facile'
```

Il est possible de fractionner une phrase en ses mots constitutifs à l'aide de la méthode `split()`.

```
[41]: phrase.split()
```

```
[41]: ['Apprendre', 'à', 'programmer', 'en', 'langage', 'Python', 'est', 'facile']
```

On peut alors aisément dénombrer les mots d'une phrase avec la fonction `len()`.

```
[42]: len(phrase.split())
```

```
[42]: 8
```

```
[43]: phrase.replace("Python", "Java")
```

```
[43]: 'Apprendre à programmer en langage Java est facile'
```