

Fiche syntaxe Python : manipulation des types de données élémentaires

Cette fiche résume les commandes essentielles à connaître pour manipuler les types de base de Python. Gardez-la pour les TP, mais par contre elle ne sera pas autorisée lors des DS car les commandes présentées ici sont élémentaires et doivent être connues par cœur.

1 Entiers

opérations arithmétiques élémentaires	:	+ - *
puissance	:	**
comparaison	:	< > <= >= == !=
division euclidienne	:	//
reste de la division euclidienne	:	%
reste et quotient de la division euclidienne	:	divmod
convertir quelque chose en entier	:	int

2 Flottants

- Les opérations arithmétiques élémentaires ont la même syntaxe que pour les entiers. L'opérateur / est une division dont le résultat est un flottant.
- On peut obtenir les fonctions mathématiques usuelles par exemple dans la bibliothèque `numpy`. Traditionnellement, on charge cette bibliothèque par `import numpy as np`, on peut alors utiliser `np.cos`, `np.sin`, `np.sqrt`, `np.exp`, `np.pi`, `np.sinh`, `np.cosh`,...
- Conversion en flottants : `float`.
- Le nombre π : `np.pi`.
- Attention : on utilise le point et non la virgule pour séparer partie entière de partie décimale. Par exemple `0.5 == 1/2`, alors que `0,5` serait compris comme le couple $(0, 5)$.

3 Chaînes de caractères

créer une chaîne	:	<code>maChaine = 'bonjour'</code> ou <code>maChaine= "bonjour"</code>
chaîne vide	:	<code>""</code>
concaténation	:	<code>+</code>
lire le caractère d'indice <code>i</code> d'une chaîne <code>c</code>	:	<code>c[i]</code>
afficher une chaîne	:	<code>print</code>
longueur d'une chaîne	:	<code>len</code>
convertir en chaîne	:	<code>str</code>

- Les commandes `<`, `>`, etc. sont encore valides pour les chaînes de caractères. Il s'agit alors de l'ordre alphabétique.
- Contrairement aux tableaux, les chaînes ne sont pas modifiables. La commande `maChaine[i] = 'b'` ne marche pas, et il n'y a pas non plus de méthode `append`.

4 Booléens

vrai, faux	:	<code>True</code> , <code>False</code>
et, ou, non	:	<code>and</code> , <code>or</code> , <code>not</code>

5 Tableaux dynamiques

Tableau vide	: []
Tableau prérempli	: monTableau = [1,2,3]
Longueur d'un tableau	: len
Accéder l'élément d'indice <i>i</i>	: monTableau[i]
Modifier l'élément d'indice <i>i</i>	: monTableau[i] = ...
Ajouter un élément à la fin	: monTableau.append(...)
Supprimer le dernier élément et le renvoyer	: monTableau.pop()
Concaténation	: +
Ajouter une liste d'éléments	: monTableau.extend([...])
Convertir en tableau	: list

Les tableaux sont *modifiables* (« mutables » en anglais). Les méthodes `append`, `extend` et `pop`, et la commande `monTableau[i]=...` modifient le tableau.

N.B. Les tableaux Python sont conçus pour pouvoir facilement ajouter ou supprimer un éléments *à la fin* du tableau. Il est possible d'insérer ou de supprimer des éléments ailleurs mais ce sera beaucoup plus lent.

6 Dictionnaires

Dictionnaire vide	: {}
Dictionnaire prérempli	: {clé1 : valeur1, clé2 : valeur2, ...}
Valeur associée à une clé	: dico[clé]
Insérer ou modifier une association	: dico[clé] = valeur

Comme les tableaux, les dictionnaires sont modifiables.