

# Algorithmique Appliquée

BTS SIO SISR

Structures de données  
fondamentales en Python

# Plan

- Notion de conteneur
- Notion d'opérations CRUD
- Tuples
- Ranges
- Lists
- Clonage et copie profonde
- Sets
- Dictionaries
- Technique "Pythonic": comprehensions
- Structure personnalisée

# Correction du travail à la maison

# **DM : Retours sur les fonctions et le débogage**

**Lien vers le sujet de DM.**

# Notion de conteneur

Container

str = liste de caractères

On veut pouvoir manipuler des ensembles d'autres choses

=> Ce sont les conteneurs (également appelées collections)

Il existe différents types de conteneurs, pour répondre à différents besoins



# Notion d'opérations CRUD

Create, Read, Update, Delete

Opérations de base  
Création, Recherche/Lecture, Mise à Jour,  
Suppression  
Insertion  
Mutabilité  
Recherche uniformisée quelque soit le conteneur



# Tuples

## Notation Immutable

Contient généralement des types différents

Utilisé pour retourner plusieurs valeurs dans une  
fonction

Utilisé pour représenter un type customisé simple

Peut contenir plusieurs fois la même valeur

Ordre d'insertion conservé

Tuple à 1 élément (1,)

Tuple de tuples

# Bornes et itérateurs

Ranges & iterables

Retour sur la fonction range qui renvoie un objet range

Immutable

Peut être comparé

Peut être itéré

Léger en mémoire => lazy evaluation

Peut être utilisé partout où l'on attend un iterable  
d'entiers

# Listes

Notation

Mutable

Le conteneur le plus utilisé en Python

Contient la plupart du temps des objets d'un unique type

Peut contenir des objets de types variés

Peut contenir plusieurs fois la même valeur

Ordre d'insertion conservé

Liste de listes

Liste à N dimensions

Liste de tuples

Tuple de listes



# **TD : Implémenter les opérations matricielles les plus classiques**



# **TD : Opérations matricielles classiques**

[Lien vers le sujet de TD.](#)

# Clonage et copie profonde

Shallow and deep copy

Rabachage : tout est OBJET

Assignation d'une même liste à 2 variables et  
modification du contenu de la liste : les 2 variables  
sont affectées

Technique pour créer un clone : utiliser `[ : ]` ou  
`.copy()` .

Problème : les objets à l'intérieur de la liste restent les  
mêmes.

Technique pour créer une copie profonde :  
`copy.deepcopy()` .

Exemples

# Ensembles

Sets

Notation

Mutable

Éléments uniques

Pas dans l'ordre d'insertion

Requis lorsque l'on a besoin de garantir l'unicité de  
chaque élément du conteneur

# Dictionnaires

Dictionaries

Notation

Mutable

Dans l'ordre d'insertion depuis Python 3.8

Utilisé très souvent

Clés uniques

Parcours des éléments

Parcours des clés uniquement ou des valeurs  
uniquement



# Technique "Pythonic": comprehensions

@TODO : vérifier comment on dit en français!!

Exemple simple avec liste

Exemple avec liste + condition

Exemple imbriqué

Warning : ne pas faire de choses trop compliquées =>  
les boucles classiques sont plus simples à  
comprendre!

Exemple avec tuple

Exemple avec set

Exemples avec dico

**TP : Utiliser un dictionnaire  
pour gérer un hôpital avec  
des patients, des médecins  
et des soins à apporter**

# **TP : Gestion d'un hôpital**

[Lien vers le sujet de TP.](#)

# Structure personnalisée

Notion de classe comme Tuple avancé

Comme un tuple, mais à la place d'accéder aux éléments par index `t[0]`, `t[1]`, `t[2]`, on peut faire `t.x`, `t.y`, `t.z` par exemple.

Fondamental pour définir des abstractions de niveau supérieur et simplifier la programmation

Mot clé `class`

La fonction spéciale `init(self, ...)`

La spécification de données membres

Usage

Shortcut en Python 3.7 :

```
from dataclasses import dataclass  
  
@dataclass
```