

## Exercices sur les types natifs

### 1.1 Réviser les types natifs en Python

Flash card :

- \*Python avancé > C22. Variables\*
- \*Python avancé > C32. Méthodes de Listes, Chaines et Dict\*
- \*Python avancé > C33. Boucles\*

### 1.2 Choisir un type

Quel type de variable sera le plus adapté pour effectuer les actions suivantes ?

- Stocker un nombre de pieces d'or dans un jeu vidéo
- Gérer une liste d'objets dans un jeu de quête
- stocker des numéros de téléphone dans un agenda téléphonique
- stocker les coordonnées d'un point fixe du plan
- stocker les coordonnées onstantanées d'un point mobile du plan.

### 1.3 Quel est le type ?

En examinant le code de ces 4 fonctions, préciser le type renvoyé en fin de script, sachant que n est un entier, et mot1, mot2 des chaines de caractères :

```
1 def parite(n):
2     return n % 2 == 0
3
4 def epele(mot1):
5     L = []
6     for lettre in mot1:
7         L.append(lettre)
8     return L
9
10 def concatenation(mot1, mot2):
11     return mot1 + mot2
12
13 def taille(mot1, mot2):
14     return len(mot1, mot2)
```

### 1.4 Panier de fruits

On cherche à représenter le panier de fruits comme sur l'image ci-dessous :



FIGURE 1 – panier de fruits : bananes, pommes rouges, pommes vertes, et oranges

On utilisera 2 implémentations différentes au cours de l'exercice.

#### 1.4.1 Liste Python

- a. Représenter le panier à l'aide d'une variable `panier` de type List en Python. Chaque élément de liste est un entier (le nombre de fruits). La place de chaque élément se réfère à l'un des fruits : banane (rang 0), pomme rouge (rang 1), pomme verte (rang 2), orange (rang 3).
- b. Créer une fonction `set` qui prend en arguments le nom du fruit (présent dans le panier), et le nombre de fruits de ce genre. Cette fonction va alors modifier la liste `panier` en conséquence, en positionnant la valeur au bon endroit.
- c. Créer une fonction `ajoute` qui prend en argument le nom du fruit à ajouter. Cette fonction ajoute un rang dans la liste `panier` avec pour valeur None.
- d. Créer une fonction `get` qui prend en argument un nom de fruit, et qui retourne le nombre de fruits de ce genre dans le panier.
- e. Créer une fonction `total` qui retourne le nombre total de fruits du panier.
- f. Créer une fonction `variete_fruits` qui retourne le nombre de fruits *différents* présents dans le panier.
- g. Tests : Comment évolue la liste `panier` au fur et à mesure des instructions suivantes ?
  - Ajouter le fruit *pêches*
  - Mettre le nombre de pommes rouges à 2
  - Mettre le nombre de bananes à 4
  - Mettre le nombre de pêches à 6
  - Demander le nombre de fruits différents du panier
  - Demander le nombre total de fruits dans le panier

Donner les instructions nécessaires pour effectuer chacun de ces changements.

- h. Prevision des erreurs.
  - Que se passe t-il si on cherche à savoir le nombre de *kakis* dans le panier ?
  - Modifier alors la fonction `get` pour éviter le déclenchement d'une erreur de ce genre.
  - Proposer également un test d'assertion dans la fonction `set` pour prévenir cette erreur.

#### 1.4.2 Dictionnaire

Répondre aux mêmes questions, mais avec cette fois, une implémentation à l'aide d'un *dictionnaire*.

### 1.4.3 Conclusion

Entre les 2 parties : implémentation en Liste, et implémentation en Dictionnaire :

- Qu'est ce qui change ?
- Qu'est ce qui reste identique ?

Conclure.

Partie 2

## Bac 2021 Metropole Sept : Exercice 5 (extrait)

Cet exercice est en entier sur la page : [allophysique.com/docs/NSI/bac/page1/](http://allophysique.com/docs/NSI/bac/page1/)

*Principaux thèmes abordés : Traitement de données en tables (CSV) et langages et programmation (spécification).*

Afin d'améliorer l'ergonomie d'un logiciel de traitement des inscriptions dans une université, un programmeur souhaite exploiter l'intelligence artificielle pour renseigner certains champs par auto-complétion. Il s'intéresse au descripteur « genre » (masculin, féminin ou indéterminé).

Pour cela il propose d'exploiter les dernières lettres du prénom pour proposer automatiquement le genre.

Pour vérifier son hypothèse, il récupère un fichier CSV associant plus de 60 000 prénoms du monde entier au genre de la personne portant ce prénom. En utilisant seulement la dernière lettre, le taux de réussite de sa démarche est de 72,9% avec la fonction définie ci-dessous :

```

1 def genre(prenom):
2     liste_M = ['f', 'd', 'c', 'b', 'o', 'n', 'm', 'l', 'k',
3               'j', 'é', 'h', 'w', 'v', 'u', 't', 's', 'r',
4               'q', 'p', 'i', 'p', 'z', 'x', 'ç', 'ö', 'ä',
5               'â', 'ï', 'g']
6     liste_F = ['e', 'a', 'ä', 'ü', 'y', 'e']
7     # ligne 7
8     if prenom[len(prenom)-1].lower() in liste_M :
9         return "M"
10    elif prenom[len(prenom)-1].lower() in liste_F :
11        return "F"
12    else :
13        return "I"
14    # Pour rappel, C.lower() convertit le caractère C en minuscule.

```

### 2.1 1. Appropriation

- Expliquer ce qu'est un fichier CSV.
- Donner le type de l'argument prenom de la fonction genre, et le type de la réponse renvoyée.

### 2.2 3. Améliorations

En prenant en compte les deux dernières lettres du prénom, il parvient à augmenter son taux de réussite à 74,4%. Pour cela, son étude du fichier CSV lui permet de créer deux listes : `liste_M2` pour les terminaisons de deux lettres associées aux prénoms masculins et `liste_F2` pour les prénoms féminins. Sur votre copie, recopier et modifier la structure conditionnelle (lignes 8 à 13) de la fonction genre afin de prendre en compte les terminaisons de deux lettres des listes `liste_M2` et `liste_F2`.

## Bac 2021 Etranger1 : Exercice 2 (extrait)

Cet exercice est en entier sur la page : [allopysique.com/docs/NSI/bac/page1/](http://allopysique.com/docs/NSI/bac/page1/)

*Notion abordée : structures de données (dictionnaires)*

Une ville souhaite gérer son parc de vélos en location partagée. L'ensemble de la flotte de vélos est stocké dans une table de données représentée en langage Python par un dictionnaire contenant des associations de type `id_velo`: `dict_velo` où `id_velo` est un nombre entier compris entre 1 et 199 qui correspond à l'identifiant unique du vélo et `dict_velo` est un dictionnaire dont les clés sont : "type", "etat", "station".

Les valeurs associées aux clés "type", "etat", "station" de `dict_velo` sont de type chaînes de caractères ou nombre entier :

- "type" : chaîne de caractères qui peut prendre la valeur "électrique" ou "classique"
- "état" : nombre entier qui peut prendre la valeur 1 si le vélo est disponible, 0 si le vélo est en déplacement, -1 si le vélo est en panne
- "station" : chaînes de caractères qui identifie la station où est garé le vélo.

Dans le cas où le vélo est en déplacement ou en panne, "station" correspond à celle où il a été dernièrement stationné.

Voici un extrait de la table de données :

```

1  flotte = {
2      12 : {"type" : "électrique", "etat" : 1, "station" : "Prefecture"},
3      80 : {"type" : "classique", "etat" : 0, "station" : "Saint-Leu"},
4      45 : {"type" : "classique", "etat" : 1, "station" : "Baraban"},
5      41 : {"type" : "classique", "etat" : -1, "station" : "Citadelle"},
6      26 : {"type" : "classique", "etat" : 1, "station" : "Coliseum"},
7      28 : {"type" : "électrique", "etat" : 0, "station" : "Coliseum"},
8      74 : {"type" : "électrique", "etat" : 1, "station" : "Jacobins"},
9      13 : {"type" : "classique", "etat" : 0, "station" : "Citadelle"},
10     83 : {"type" : "classique", "etat" : -1, "station" : "Saint-Leu"},
11     22 : {"type" : "électrique", "etat" : -1, "station" : "Joffre"}
12 }
```

`flotte` étant une variable globale du programme.

Toutes les questions de cet exercice se réfèrent à l'extrait de la table `flotte` fourni cidessus.

### 3.1 Question 1

- 1.a. Que renvoie l'instruction `flotte[26]` ?
- 1.b. Que renvoie l'instruction `flotte[80]["etat"]` ?

### 3.2 Question 2

Voici le script d'une fonction :

```

1  def proposition(choix):
2      for v in flotte:
3          if flotte[v]["type"] == choix and flotte[v]["etat"] == 1:
4              return flotte[v]["station"]
```

- 2.a. Quelles sont les valeurs possibles de la variable `choix` ?
- 2.b. Expliquer ce que renvoie la fonction lorsque l'on choisit comme paramètre l'une des valeurs possibles de la variable `choix`.

### 3.3 Question 3

- 3.a. Écrire un script en langage Python qui affiche les identifiants (`id_velo`) de tous les vélos disponibles à la station "Citadelle".
- 3.b. Écrire un script en langage Python qui permet d'afficher l'identifiant (`id_velo`) et la station de tous les vélos électriques qui ne sont pas en panne.

Partie 4

## Travail pratique

### 4.1 Le jeu de 52 cartes

Vous devez implémenter un jeu de cartes de cartes et programmer les fonctions qui permettront de l'utiliser.

- a. Ecrire les listes Python `couleurs` et `valeur` qui contiendront respectivement :
  - les couleurs de cartes (pique, trefle, carreau, coeur)
  - les valeurs ('1', '2', '3', ... '10', 'V', 'D', 'R')
- b. Ecrire la liste `jeu` qui contient le jeu de cartes. Les cartes se présentent sous la forme de tuple, par exemple ('pique', '1')
- c. Faire une copie par valeur de la liste `jeu` dans une nouvelle liste `jeu_melange`
- d. La librairie `random` apporte la fonction `shuffle` qui mélange les éléments d'une liste. Utiliser cette librairie pour mélanger la liste `jeu_melange`. Ecrire alors une fonction appelée `melange` qui retourne une nouvelle liste mélangée à partir de celle passée en paramètre.
- e. Comment distribuer alors 8 cartes de ce jeu mélangé à chacun des `n` joueurs de la partie ? (`n` pas trop grand;-)

## Problèmes

### 5.1 Des mathématiciens célèbres

Chaque lettre des noms des six mathématiciens a été remplacée par un nombre. Cette substitution suit une règle très spéciale. Pouvez-vous la percer ?

- 10 8 3 4 4 3 - 13 3 - 14 3 4 20 1 5
- 8 12 1 1 23 - 26 3 21 5 9 26
- 3 7 23 19 8 13 3
- 19 3 9 26 16 1 4 13 - 3 7 19 3 4
- 12 9 10 16 8 3 - 13 3 4 20 1 8 26
- 10 6 5 16 1 15 9 4 3

Comment réaliser ceci à l'aide d'un algorithme ?

### 5.2 Lettres manquantes

Quelle séquence suivent les lettres qui ont été supprimées dans les mots suivants ?

- \_LM\_N\_CH
- \_AO\_A\_
- \_ON\_O\_TER
- \_IVI\_EN\_E
- \_L\_M\_NT

Peut-on reconstruire ces chaînes grâce à un algorithme ?

### 5.3 METHODOLOGIE

Pour résoudre le problème :

#### 1. Lecture du sujet

- Avoir une lecture active (*anoter l'énoncé*)
- Tester, chercher des solutions (*utiliser un cahier d'essais*)

#### 2. Ré-Énoncer le problème :

- quelles sont les données présentes ?
- y-a-t-il des données manquantes ?
- Énoncer la VRAIE question à résoudre.
- peut-on résoudre un sous-problème, plus facile, moins général ?

### 3. Créer un visuel à partir du problème

cela peut-être :

- un dessin
- un diagramme
- une table
- des objets connectés par des flèches
- ...
- de quelles structures de données va-t-on avoir besoin :
  - les objets peuvent se disposer en séquence, dans un ordre précis => liste
  - les objets sont plutôt disposés en paires (clé :valeur) => dictionnaire

### 4. Algorithme

- de quelles fonctions particulières dispose t-on ?
- Y-a-t-il un algorithme déjà connu pour résoudre ? \* Exprimer celui-ci en langage naturel

### 5. Ecriture

Une fois l'algorithme créé et essayé :

- mettre autant que possible son code en fonctions
- quels tests de débogage, assertions, ... vont rendre celui-ci plus robuste