|  |
| --- |
|  |
| Python |
|  |

|  |
| --- |
| Soubika BISOO  13/12/2021 |

Table des matières

[Veille 1 : Les variables 2](#_Toc90457776)

[Qu’est-ce qu’une variable ? 2](#_Toc90457777)

[Lister les types de variables : 2](#_Toc90457778)

[Lister les opérateurs Python : 2](#_Toc90457779)

[Lister les fonctions utiles pour traiter les chaînes de caractères 2](#_Toc90457780)

[Veille 2 : Les listes et les Tuples 3](#_Toc90457781)

[Qu’est-ce qu’une liste en Python ? 3](#_Toc90457782)

[Lister les fonctions utiles pour manipuler les listes 3](#_Toc90457783)

[Comment parcourir une liste ? 3](#_Toc90457784)

[Explorer les listes de compréhension 3](#_Toc90457785)

[Veille 3 : Les Dictionnaires 5](#_Toc90457786)

[Qu’est ce qu’un dictionnaire en Python ? 5](#_Toc90457787)

[Comment créer un dictionnaire en Python ? 5](#_Toc90457788)

[Lister les opérations communes pour manipuler les dictionnaires 5](#_Toc90457789)

[Comment parcourir un dictionnaire ? 5](#_Toc90457790)

# Veille 1 : Les variables

## Qu’est-ce qu’une variable ?

Zone de la mémoire de l’ordinateur dans laquelle une valeur est stockée. Aux yeux du programmeur, cette variable est définie par un nom, alors que pour l’ordinateur, il s’agit d’une adresse, cad d’une zone particulière de la mémoire

## Lister les types de variables :

* Char
* Integer int
* Floats
* String str
* Boleen

## Lister les opérateurs Python :

* Arithmétiques 🡺 +, -, \*, /, %, \*\*, //
* Relationnels 🡺 == != < > <= >=
* Logiques 🡺 and or not
* Affectation 🡺 = += -= \*=
* Spéciaux 🡺 is, is not, in, not in

## Lister les fonctions utiles pour traiter les chaînes de caractères

* Concaténation 🡺 +
* .format()
* .find()
* Type()
* .lower()
* .capitalize()
* .title()
* .replace()
* .translate
* .split()

# Veille 2 : Les listes et les Tuples

## Qu’est-ce qu’une liste en Python ?

Une liste est une structure de données qui contient une série de valeurs. Python autorise la construction de liste contenant des valeurs de types différents (par exemple entier et chaîne de caractère), ce qui leur confère une grande flexibilité. Une liste est déclarée par une série de valeurs séparées par des virgules, encadré par des crochets.

## Lister les fonctions utiles pour manipuler les listes

* append()
* len()
* list(range())
* min()
* max()
* sum()

## Comment parcourir une liste ?

Elément par élément : liste[indice]

|  |
| --- |
| Boucle For :  for element in liste :  print(element) |

## Explorer les listes de compréhension

La notion de liste de compréhension représente une manière originale et très puissante de générer des listes. La syntaxe de base consiste au moins en une boucle for au sein de crochets précédés d’une variable.

new\_list = [function(item) **for** item **in** list **if** condition(item)]

Ex :

>>> [i for i in range (10)]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

>>> [2 for i in range (10)]

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

List VS Tuples

Les tuples correspondent aux listes à la différence qu’ils sont non modifiables. Ils utilisent les parenthèses au lieu des crochets

# Veille 3 : Les Dictionnaires

## Qu’est ce qu’un dictionnaire en Python ?

Les dictionnaires sont des collections non ordonnées d’objet, c'est-à-dire qu’il n’y a pas de notion d’ordre (donc pas d’indice). On accède aux valeurs d’un dictionnaire par des clés.

## Comment créer un dictionnaire en Python ?

La définition d’un dictionnaire vide se fait avec les accolades {}.

Ensuite, on remplit le dictionnaire avec différentes clés auxquelles on affecte des valeurs.

On peut mettre autant de clés qu’on veut dans un dictionnaire.

Exemple :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

## Lister les opérations communes pour manipuler les dictionnaires

* Ajouter des clés et des valeurs
* Modifier les valeurs des clés
* Supprimer des clés valeurs
* Modifier les clés 🡺 plus compliquer

## Comment parcourir un dictionnaire ?