

# Synthèse python

# Table des matières

I - Exercice : Synthèse premier programme en python	3
II - Exercice : Synthèse variables et données	4
III - Exercice : Synthèse listes	5
IV - Exercice : Synthèse dictionnaires	6
V - Exercice : Synthèse sur les conditions	7
VI - Exercice : Synthèse sur les boucles	8
VII - Exercice : Synthèse sur les fonctions	9
VIII - Exercice : Synthèse écriture	10

# Exercice : Synthèse premier programme en python



## Choix de l'environnement de programmation

Un langage de programmation est nécessaire pour l'écriture des programmes : un langage simple d'usage, interprété, concis, libre et gratuit, [REDACTED], largement répandu, riche de [REDACTED] adaptées aux thématiques étudiées et bénéficiant d'une [REDACTED] d'auteurs dans le monde éducatif est nécessaire. Le langage choisi est [REDACTED].

## Environnement de programmation

Pour programmer en python il faudra installer [REDACTED].

## Pyzo

Nous installerons également [REDACTED] fourni une aide à la programmation. En effet cet interface à l'aide de ses [REDACTED] :

- [REDACTED], l'endroit où l'on [REDACTED].
- [REDACTED], fenêtre dans laquelle [REDACTED]
- [REDACTED] qui permet de donner des informations sur les erreurs de code.

## Trinket

Sur le web<sup>1</sup> [REDACTED] permet de programmer en python à partir d'un [REDACTED].

## Affichage du code

- Pour [REDACTED] du code on utilise la commande ; [REDACTED] avec l'expression à imprimer [REDACTED]
- Si je veux afficher du [REDACTED] il faut le placer entre "[REDACTED]"

---

<sup>1</sup>. <https://trinket.io/>

# Exercice : Synthèse variables et données



## Les variables

C'est une [ ] que l'on stocke dans une case de la mémoire de l'ordinateur. On dit qu'elle est "[ ]" car c'est une valeur qui [ ] pendant le déroulement du [ ].

Si on affecte à la variable [ ] le contenu [ ]  
[ ] = "[ ]"

## Nommage des variables

- Utilisez des noms descriptifs : `quantite_en_stock`
- Utilisez des mots complets
- Utilisez uniquement des caractères [ ] et des [ ] et surtout [ ]

## Type de données

- Les entiers ou [ ]
- Les [ ] ou float
- Les [ ] ou strings
- Les booléens [ ]

## Les opérations

On retrouve les quatre opérations de base plus l'opération qui donne le reste de x divisé par y :  
`x%y`

# Exercice : Synthèse listes



## Les listes

Dans Python, on utilise [ ] pour indiquer une liste

## Opération sur les listes

La liste est numéroté de 0 à n de gauche à droite et de -1 à -n de droite à gauche

- Pour avoir accès à élément il faut indiquer son numéro [2] renvoi le troisième élément de la liste
- Pour ajouter un élément on utilise plateformes\_sociales.append("TikTok")
- Pour retirer un élément on utilise plateformes\_sociales.remove("TikTok")
- Pour connaître la longueur de la liste on utilise print(len(platformes\_sociales))
- Pour trier par ordre croissant on utilise plateformes\_sociales.sort()

Il faut au final utiliser la commande print(platformes\_sociales) pour voir le résultat.

## Les tuples

La principale différence est que les tuples sont immutables, alors que les listes sont mutables.

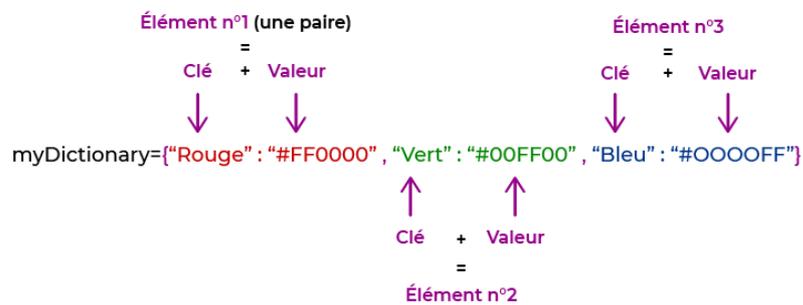
# Exercice : Synthèse dictionnaires

# IV

## A quoi sert un dictionnaire

- Les dictionnaires servent à enregistrer des [ ]
  - Un dictionnaire est une structure de données qui enregistre des données dans des paires [ ] .
- On peut aussi dire que chaque élément du dictionnaire est défini par un couplet composé d' [ ] et d' [ ] .

## Définition d'un dictionnaire



*Dictionnaire*

- Les dictionnaires sont indiqués par des [ ] au début et à la fin.
- Chaque paire clé-valeur comprend un [ ] placé entre la clé et la valeur
- Une [ ] à la fin.
- Chaque dictionnaire doit être [ ]

## Ajout clé-valeur

On utilise le nom du dictionnaire et la paire clé valeur

## Supprimer paire clé-valeur

Utiliser le mot-clé **del** et la clé que vous voulez supprimer

# Exercice : Synthèse sur les conditions

## V

### Instruction if else

Avec une `if else`, vous pouvez exécuter certaines lignes de code uniquement si une certaine condition est vraie. Si cette condition est fautive, le code `else` est exécuté.

### Clause elif

Les instructions `elif` vous permettent de définir plusieurs conditions. Le mot-clé `elif` vous permet d'ajouter autant de conditions que vous voulez. Vous devez ensuite terminer avec `else`.

### Conditions multiples avec opérateurs

Pour vérifier si une condition est vraie ou fautive dans une expression, vous pouvez utiliser les opérateurs logiques :

- `and` vérifie si deux conditions sont vraies.
- `or` vérifie si au moins une condition est vraie.
- `not` vérifie si une condition est fautive (c'est-à-dire fautive).

### Expressions comparatives

- Égal à : `==`
- Différent à : `!=`
- Inférieur à : `<`
- Inférieur ou égal à : `<=`
- Supérieur : `>`
- Supérieur ou égal à : `>=`

# Exercice : Synthèse sur les boucles

# VI

## Boucle for

- Si la boucle [ ] la condition de [ ] elle est donc [ ] et [ ] de la boucle.
- Une boucle for permet de répéter du code [ ]

## Boucle while

- Le code dans l'instruction while s'exécute jusqu'à ce que la [ ]
- Une boucle while permet de répéter du code jusqu'à ce qu' [ ]

# Exercice : Synthèse sur les fonctions

## VII

### Définition d'une fonction

On définit une fonction avec [REDACTED], le [REDACTED] de la fonction, [REDACTED] et [REDACTED]

### Pourquoi utiliser une fonction

- Les fonctions sont un moyen de répéter des fonctionnalités et de séparer du code dans des modules différents.
- Vous pouvez créer des fonctions avec ou sans paramètres d'entrée.
- Les fonctions sont [REDACTED] (code écrit) **appelées** [REDACTED] (code exécuté) et peuvent [REDACTED] des informations (une valeur est donnée comme résultat).

# Exercice : Synthèse écrite

# VIII

## Ne vous répétez pas

Utilisez des [ ]

## Une chose à la fois

Chaque fonction doit être responsable [ ], et rien de plus.

## Commenter!

- commentaire à [ ], il faut mettre en début de ligne : [ ]
- commentaires à [ ] sont entourés de trois guillemets : [ ]