



**CONSEIL ET EXPERTISE TECHNIQUE**  
Infrastructure • Cybersecurity • Cloud • Data

# Python 02

Matthieu DESTOMBES  
Mail : [matthieu.destombes@ynov.com](mailto:matthieu.destombes@ynov.com)

01 Mars 2021  
Toulouse

# Travaux Pratiques

- Utilisation des notions avancées
  - Les listes. Les dictionnaires. Les tuples.
  - La portée des variables.
  - Les interactions console.
  - Les fonctionnalités arithmétiques.
  - La mise en place de fonctions avancées.
  - L'utilisation de modules.
- Utiliser les commentaires
- Utiliser les doc-chaînes

# Utilisation des notions avancées

## Listes / Portée des variables

- Dans un fichier python nommé « 01\_list\_management.py »
- Créer 3 listes
  - « my\_list\_one » contenant les nombres 10 et 20
  - « my\_list\_two »
  - « my\_list\_three »
- Ajouter les valeurs « Ten » et « Twenty » dans « my\_list\_two »
- Ajouter les listes « my\_list\_one » et « my\_list\_two » dans « my\_list\_three »
- Créer une fonction « display\_all\_list » qui réalise les actions suivantes
  - Utilisation de la « Structure de contrôle » FOR
  - Affiche le contenu des trois listes
- Dans la fonction « main », appeler la fonction « display\_all\_list »

# Utilisation des notions avancées

## Liste / Portée des variables / Fonction avancées (Part I)

- Dans un fichier python nommé « 02\_list\_management.py »
- Recopier le comportement du fichier précédent
- Créer une fonction « check\_add\_or\_remove » qui réalise les actions suivantes
  - Prendre 2 arguments en entrée
    - « input\_list », une liste...
    - « input\_value », une valeur...
  - Retourner une variable de sortie
    - « input\_list » la liste modifiée
  - Instruction de la fonction
    - Avec la « structure de contrôle » IF
      - Vérifier la présence de la valeur « input\_value » dans la liste « input\_list »
      - Si présente, la retirer de la liste
      - Si absente, l'ajouter dans la liste

## Liste / Portée des variables / Fonction avancées (Part II)

- Toujours dans le fichier python nommé « 02\_list\_management.py »
- Dans la fonction « main » réaliser les actions suivantes
  - Appeler la fonction « display\_all\_list »  
Pour afficher le contenu présent à l'origine du programme
  - Appeler la fonction « check\_add\_or\_remove » avec
    - « my\_list\_one » et « Thirty »
    - « my\_list\_two » et « 30 »
  - Appeler la fonction « display\_all\_list »  
Pour afficher le contenu modifié dans le du programme
  - Avec la « structure de contrôle » FOR et la « structure de donnée » 20, 10, "Twenty", "Ten"
    - Appeler la fonction « check\_add\_or\_remove » avec
      - « my\_list\_one » et la valeur courante de la « structure de contrôle »
      - « my\_list\_two » et la valeur courante de la « structure de contrôle »
  - Appeler la fonction « display\_all\_list »  
Pour afficher le contenu modifié dans le du programme
- A la fin du programme, vous devez obtenir
  - « my\_list\_one » avec des nombre en lettre de manière décroissante
  - « my\_list\_two » avec des nombre en chiffre de manière décroissante
  - « my\_list\_three » avec les deux listes modifiées

# Utilisation des notions avancées

## Utilisation de module / Les dictionnaires / Les interactions console (Part I)

- Dans un fichier python nommé « 03\_identities.py »
- Créer 1 dictionnaire « all\_identity »
- Créer une fonction « set\_identity » qui réalise les actions suivantes
  - Créer 1 dictionnaire « current\_identity »
  - Demander à l'utilisateur son prénom, puis son nom.
  - Sauvegarder les deux données dans le dictionnaire « current\_identity »
    - Clé « id\_f\_name » avec le prénom
    - Clé « id\_l\_name » avec le nom de famille
  - Générer un identifiant « unique\_id » entre 0 et 5
  - Convertir cet identifiant en chaîne de caractère sur 2 caractères (00)
  - Tant que l'identifiant existe en tant que clé de « all\_identity », relancer la génération
  - Affecter le dictionnaire « current\_identity », contenant les informations du nouvel utilisateur, dans le dictionnaire « all\_identity », en l'associant à la clé « unique\_id »
  - Retourner un Tuple contenant « unique\_id » et un booléen qui indique si la génération a dû être relancée au moins une fois.

## Utilisation de module / Les dictionnaires / Les interactions console (Part II)

- Toujours dans le fichier python nommé « 03\_identities.py »
- Dans la fonction « main »
  - Utiliser la « structure de contrôle » FOR
    - Faire 3 appels à la fonction « set\_identity »
    - Afficher le message « New user '[prénom] [nom de famille]' have been created » en utilisant les informations de l'utilisateur fraîchement créé
    - Dans le cas où la génération a été relancé
      - Afficher une alerte par le message « /\ Take care ID need to be re-created /\ »
  - Afficher le contenu du dictionnaire « all\_identity »
- Avez-vous reçu l'alerte ?
- Essayer avec 7 appels
  - Que se passe-t-il ?

# Utilisation des notions avancées

## Les fonctionnalités arithmétiques (Part I)

- Dans un fichier python nommé « 04\_math\_helper.py »
- Créer une fonction « multiplication\_op » qui réalise les actions suivantes
  - Prendre 1 argument en entrée
    - « input\_base », un chiffre
  - Utilisation de la « Structure de contrôle » FOR de 1 à 10 compris
    - Faire le calcul de la multiplication « input\_base » par l'élément en cours
    - Affiche la multiplication avec les chiffres de gauche sur 2 caractères et l'égalité avec le résultat
    - « XX \* YY = [résultat] »
    - Essayez de mettre en forme vos résultats pour qu'ils soient ordonnés
- Dans la fonction « main »
  - Utiliser la « structure de contrôle » FOR de 1 à 10 compris
  - Afficher le multiplicateur en cours « Table of '[current]' »
  - Appeler la fonction « multiplication\_op » avec l'élément courant comme paramètre

## Les fonctionnalités arithmétiques (Part II)

- Dans un fichier python nommé « 05\_math\_helper.py »
- Reprendre le principe de la fonction « multiplication\_op » et faire de même pour les opérations suivantes
  - La division dans une fonction « divide\_op »
  - L'addition dans une fonction « plus\_op »
  - La soustraction dans une fonction « minus\_op »
- Dans la fonction « main »
  - Utiliser la « structure de contrôle » FOR de 1 à 10 compris
  - Afficher le dénominateur commun « Tables of '[current]' »
  - Appeler la fonction « multiplication\_op » avec l'élément courant comme paramètre
  - Appeler la fonction « divide\_op » avec l'élément courant comme paramètre
  - Appeler la fonction « plus\_op » avec l'élément courant comme paramètre
  - Appeler la fonction « minus\_op » avec l'élément courant comme paramètre
- Faites en sorte que les résultats des 4 fonction soient affichée sur la même ligne, pour chaque occurrence
  - « XX \* YY = [résultat] | XX / YY = [résultat] | XX + YY = [résultat] | XX - YY = [résultat] »