**Créer des données avec Python**

**Description**

*Vous venez d'être nommé(e) responsable de la gestion des contacts pour l'organisation d'un festival de musique. Votre mission est de créer une application de gestion de contacts pour gérer les artistes, les organisateurs et les sponsors du festival.*

*Pour chaque contact, vous devez stocker les informations suivantes : nom, prénom, numéro de téléphone, adresse e-mail.*

*Vous devez utiliser des variables pour stocker les informations de chaque contact, des listes pour stocker l'ensemble des contacts, et des dictionnaires pour stocker les informations spécifiques à chaque contact.*

Haut du formulaire

**Question 1**

Correcte

Pour commencer, vous allez créer une variable nom\_contact et y assigner le nom de votre premier contact. Comment stocker le nom d'un contact en utilisant des variables ?

 nom\_contact = "Jackson"

 nom\_contact("Jackson")

 nom\_contact = [Jackson]

 nom\_contact = {"nom": "Jackson"}

*nom\_contact = "Jackson"  est la bonne réponse, car elle permet de stocker le nom du contact en utilisant une variable de type chaîne de caractères (string). Les autres options sont incorrectes, car elles utilisent une fonction, une liste et un dictionnaire pour stocker le nom du contact, ce qui n'est pas nécessaire pour une simple variable de nom de contact.*

**Question 2**

Correcte

Maintenant que vous avez pu stocker le nom du contact, vous devez également enregistrer les autres informations, telles que le prénom, le numéro de téléphone et l'adresse e-mail. Pour ce faire, vous allez utiliser un dictionnaire.

Comment stocker les informations de chaque contact dans un dictionnaire ?

 contact\_infos = [nom: "Jackson", prenom: "Michael", telephone: "123-456-7890", email: "j.michael@email.com"]

contact\_infos = {"nom": "Jackson", "prenom": "Michael", "telephone": "123-456-7890", "email": "j.michael@email.com"}

contact\_infos = [Jackson, Michael, 123-456-7890, j.michael@email.com]

contact\_infos = (nom="Jackson", prenom="Michael", telephone="123-456-7890", email="j.michael@email.com")

*Pour stocker les informations de chaque contact dans un dictionnaire, il faut utiliser la syntaxe spécifique de création de dictionnaire en Python. Cette syntaxe utilise des accolades  {}  pour entourer la liste des paires clé-valeur, séparées par des virgules.*

*Dans votre cas, chaque contact a un nom, un prénom, un numéro de téléphone et une adresse e-mail, donc il faut créer un dictionnaire pour chaque contact avec les quatre clés correspondantes. Les clés sont des chaînes de caractères qui représentent les noms des informations stockées, et les valeurs sont les données elles-mêmes.*

*contact\_infos = {"nom": "Jackson", "prenom": "Michael", "telephone": "123-456-7890", "email": "*[*j.michael@email.com*](mailto:j.michael@email.com)*"} est donc la bonne réponse.*

**Question 3**

Correcte

Vous hésitez à propos du format de l'adresse e-mail, et il vaut mieux dans ce cas vérifier si elle correspond à vos informations.

Comment accéder à l’e-mail d’un contact stocké dans le dictionnaire  contact\_infos   ?

 contact\_infos["email"]

 contact\_infos."email"

contact\_infos.get(email)

contact\_infos.email

*Pour accéder à une valeur dans un dictionnaire en Python, on utilise la syntaxe suivante :  nom\_du\_dictionnaire["clé"]   . Dans notre cas, la clé que nous voulons utiliser est email  . Ainsi, pour accéder à l'e-mail d'un contact stocké dans le dictionnaire contact\_infos, il faut utiliser la syntaxe contact\_infos["email"]  .*

**Question 4**

Correcte

Oh zut, il y a une erreur avec l'e-mail ! Le véritable e-mail estjackson.michael@email.com . Vous devez le modifier immédiatement.

Comment modifier l'adresse e-mail d'un contact stocké dans le dictionnaire  contact\_infos   ?

contact\_infos["email"] = "jackson.michael@email.com"

contact\_infos.set("email", "jackson.michael@email.com”)

contact\_infos.add("email", "jackson.michael@email.com »)

Réponse : 1) contact\_infos["email"] = "jackson.michael@email.com"`

*Pour modifier une valeur dans un dictionnaire, il suffit d'utiliser la syntaxe  nom\_dictionnaire[nom\_cle] = nouvelle\_valeur   . Dans le cas présent, la clé à modifier est "email"  et la nouvelle valeur est l'adresse e-mail mise à jour. Donc, la réponse correcte est :*

*contact\_infos[email]="jackson.michael@email.com"*

**Question 5**

Correcte

Très bien, vous avez stocké les informations de votre premier contact dans la variable contact\_infos  sous la forme d'un dictionnaire. Maintenant, vous devez l'ajouter à votre carnet de contacts. Donc vous allez commencer par créer une liste nommée contacts  pour stocker l'ensemble des contacts.

Comment créer une liste qu’on assignera à la variable contacts  ?

contacts = {}

contacts = create\_list()

contacts = list

contacts = []

*En Python, pour créer une liste, on utilise des crochets []  . Donc, la réponse correcte est  contacts = []   , qui crée une liste vide et la stocke dans la variable contacts. Cette liste pourra ensuite être remplie avec les informations de nos différents contacts.*

**Question 6**

Correcte

C’est parti ! Il est temps d’ajouter votre premier contact dans la liste.

Comment ajouter un contact stocké dans la variable contact\_infos  à une liste de contacts existante ?

contacts.append(contact\_infos)

contacts.add(contact\_infos)

contacts.insert(contact\_infos)

contacts.extend(contact\_infos)

*Pour ajouter un élément à une liste en Python, on utilise la méthode append()  . Cette méthode permet d'ajouter un élément à la fin de la liste, sans modifier les éléments déjà existants. La bonne réponse est donc contacts.append(contact\_infos)  .*

**Question 7**

Correcte

Maintenant, vous allez utiliser la fonction print()  pour afficher le contenu de la liste contacts  , afin de visualiser votre carnet de contacts.

Que va afficher la fonction print  en sortie qui stocke un contact dans la liste contacts  ?

{'nom': 'Jackson', 'prenom': 'Michael', 'telephone': '123-456-7890', 'email': 'jackson.michael@email.com’}

[{'nom': 'Jackson', 'prenom': 'Michael', 'telephone': '123-456-7890', 'email': 'jackson.michael@email.com’}]

['nom': 'Jackson', 'prenom': 'Michael', 'telephone': '123-456-7890', 'email': 'jackson.michael@email.com’]

En l'occurrence, nous avons une liste de contacts qui contient des dictionnaires représentant chaque contact. Ainsi, la sortie de la fonction print()  sera encadrée par des crochets []  pour représenter la liste. Chaque élément de la liste est un dictionnaire, qui sera lui-même délimité par des accolades {}  . Dans le cas d’une liste ne contenant qu'un seul élément, il n'y aura qu'une seule paire d'accolades pour représenter ce dictionnaire dans la sortie du print()  . La bonne réponse est donc : [{'nom': 'Jackson', 'prenom': 'Michael', 'telephone': '123-456-7890', 'email': 'jackson.michael@email.com’}]

**Question 8**

Correcte

Votre carnet de contacts est maintenant complet, avec plusieurs contacts ajoutés. Cependant, vous réalisez qu'un contact qui n'existe plus a été ajouté par erreur. Il est donc nécessaire de le supprimer.

Comment supprimer un contact d'une liste de contacts ? (Prenez en compte que vous souhaitez supprimer l'élément contenu dans la variable  contact\_a\_supprimer   .)

contacts.remove(contact\_a\_supprimer)

contacts.delete(contact\_a\_supprimer)

contacts.pop(contact\_a\_supprimer)

contacts.clear(contact\_a\_supprimer)

*Pour supprimer un élément spécifique d'une liste, on peut utiliser la méthode  remove()   , qui prend comme argument l'élément à supprimer. Cette méthode recherche l'élément dans la liste et le supprime. Si l'élément n'est pas présent dans la liste, elle renvoie une erreur ValueError  . La bonne réponse et donc* contacts.remove(contact\_a\_supprimer)   .

**Question 9**

Correcte

Vous souhaitez obtenir une estimation du nombre de contacts présents dans votre carnet avant de terminer.

Comment compter le nombre total de contacts dans la liste contacts  ?

contacts.count()

len(contacts)

contacts.length()

La fonction len()  est une fonction intégrée en Python qui permet de renvoyer la taille d'un objet, c'est-à-dire le nombre d'éléments qu'il contient. Dans le cas de la liste contacts  , la fonction len(contacts)  renverra le nombre total de contacts stockés dans cette liste. La bonne réponse est len(contacts)  .

Bas du formulaire