



PYTHON

DEVOIR MAISON

Distribué le 20 Oct. 2020

À rendre le 26 Oct. 2020

Préambule

1. Créez un dossier `dm_python_1_prenom` (en remplaçant `prenom` par votre prénom). Créez un fichier `.py` par exercice, les noms de fichier sont donnés dans les intitulés de chaque exercice. Enregistrez les fichiers dans le dossier que vous avez créé
2. Quand vous avez terminé, compressez votre dossier et envoyez moi le fichier compressé sur Slack en DM
3. Commentez votre code, décrivez ce que vous faites: **pas de commentaire = -1 point**
4. Les questions sont indépendantes les unes des autres, ne bloquez pas sur un bug ou une question. Si vous n'y arrivez pas ou que votre bug vous résiste, commentez ce qui crée le bug et passez à la question suivante

Debugging (1 points - `debug.py`)

Copiez les blocs de code suivants dans votre fichier et corrigez les erreurs qu'ils contiennent.

Rappel: VScode vous retourne des informations dans le terminal concernant le bug

Consigne	Bloc code
Affichage terminal	<pre>print "Hello World!"</pre>
Déclaration string	<pre>var_str = "hello"</pre>
Déclaration bool	<pre>var_bool = Right</pre>
Déclaration list	<pre>var_list_a = [1, 2 3, 4]</pre>
Déclaration list	<pre>var_list_b = [1, 2, 3, 4]</pre>
Déclaration dict	<pre>var_dict_a = "user": "name", "age": 34</pre>
Déclaration dict	<pre>var_dict_b = ["user": "name", "age": 34]</pre>
Déclaration dict	<pre>var_dict_c = {"user": "name", "age": 34, "last_log": "2020-01-01"}</pre>

Collections

List (6 points)

Easy (2 points - `list_easy.py`)

1. Créez une liste vide nommée `lst_easy`
2. Ajoutez l'élément `1` à `lst_easy`
3. Ajoutez les éléments de la liste `add_to_lst` à `lst_easy` : `add_to_lst = [2, 3]`
4. Affichez la taille de `lst_easy`
5. Ajoutez la liste `list_to_lst` à `lst_easy` : `list_to_lst = [4, 5]`

Bonus round (+ 0.25 points)

1. Affichez la taille de `lst_easy`, et expliquez en quelques mots dans un commentaire pourquoi sa taille est de 4 et non 5

Résultat après exécution:

```
lst_easy = [1, 2, 3, [4, 5]]
```

Medium (4 points - `list_medium.py`)

À partir de code suivant:

```
days = ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday"]
```

1. Ajoutez `"sunday"` à la liste `days`
2. Affichez les 3 premiers jours de la semaine
3. Affichez les 3 derniers jours de la semaine
4. Affichez `"thursday"` via à son index

Dict (13 points)

Easy (3 points - `dict_easy.py`)

Le dict `products_easy` représente des produits et leur prix. Les clefs étant le nom des produits et les valeurs leur prix.

```
products_easy = {  
    "smartphone": 1000,  
    "laptop": 300  
}
```

Affichage de données

1. Affichez le prix de `"smartphone"`

Ajout de données

1. Ajoutez la clef `"watch"` ayant pour valeur `2000`

Modification de données

1. Changez la valeur de "laptop" pour 4000

Résultat après exécution:

```
products_easy = {  
    "smartphone": 1000,  
    "laptop": 4000,  
    "watch": 2000  
}
```

Medium (4 points - dict_medium.py)

Le dict `products_medium` représente des produits, leur prix et la quantité en stock. Les clefs étant le nom des produits et les valeurs étant une liste de deux éléments, le premier représentant le prix et le second la quantité en stock.

```
{product_name: [price, quantity]}
```

```
products_medium = {  
    "smartphone": [1000, 16],  
    "laptop": [4000, 0],  
    "watch": [2000, 54]  
}
```

Affichage de données

1. Affichez le prix de "smartphone"
2. Affichez la quantité de "laptop" en stock

Ajout de données

1. Ajoutez la clef "keyboard" ayant pour prix 500 et pour quantité 72
("keyboard": [500, 72])

Modification de données

1. Changez le prix de "smartphone" pour 1500

Bonus round (+ 0.25 points)

1. À l'aide d'une structure conditionnelle, affichez "Out of Stock" si "laptop" a

une quantité de `0`, sinon affichez la quantité

Résultat après exécution:

```
products_medium = {  
    "smartphone": [1500, 16],  
    "laptop": [4000, 0],  
    "watch": [2000, 54],  
    "keyboard": [500, 72]  
}
```

Hard (6 points - `dict_hard.py`)

Le dict `products_hard` représente des produits, chaque produit a pour valeur un dictionnaire avec des caractéristique propre au produit. Les caractéristiques principales sont le prix (`"price"`), le quantité (`"quantity"`), les composants (`"components"`), les tags (`"tags"`) et un flag précisant si le produit est en stock ou non (`"in_stock"`).

```

products_hard = {
    "smartphone": {
        "price": 1500,
        "quantity": 16,
        "components": {
            "screen": "4K Gorilla Glass",
            "CPU": "High Efficiency CPU",
            "GPU": "High Efficiency GPU"
        },
        "tags": ["tech", "smartphone"],
        "in_stock": True
    },
    "laptop": {
        "price": 4000,
        "quantity": 0,
        "components": {
            "screen": "27in 8K",
            "CPU": "Best Tech CPU",
            "GPU": "Best Tech GPU"
        },
        "tags": ["tech", "computer"],
        "in_stock": True
    },
    "watch": {
        "price": 2000,
        "quantity": 54,
        "components": {
            "screen": "Small Screen Inc.",
            "CPU": "Downsize CPU",
            "GPU": "Downsize GPU"
        },
        "in_stock": True
    }
}

```

Affichage de données (2 points)

1. Affichez le prix de "smartphone"
2. Affichez la quantité de "laptop" en stock
3. Affichez les composants ("components") de "watch"
4. Affichez les tags de "laptop"

Ajout de données (3 points)

1. Ajoutez la clef "keyboard" ayant pour valeur un dict vide

-
2. Dans `"keyboard"` , ajoutez la clef `"price"` ayant pour valeur `500`
 3. Dans `"keyboard"` , ajoutez la clef `"quantity"` ayant pour valeur `72`
 4. Dans `"keyboard"` , ajoutez la clef `"components"` ayant pour valeur un dict vide (`{}`)
 5. Dans `"watch"` , créez la clef `"tags"` ayant pour valeur la liste de tags suivants:
`["tech", "smartwatch", "iot"]`
 6. Ajouter le tag `"iot"` au `"tags"` de `"smartphone"`

Modification de données (1 points)

1. Dans `"watch"` dans `"components"` , changez la valeur de `"screen"` pour `Little Screen Corp.`
2. Mettez à jour la valeur de la clef `"in_stock"` de `"laptop"` à `False`

Bonus round (+ 0.25 points)

1. À l'aide d'une boucle `for` , affichez pour chaque produit
`product_name: €price, quantity units` . Par exemple:
`smartphone: €1500, 16 units`

Résultat après exécution:

```
products_hard = {
  "keyboard": {
    "price": 500,
    "quantity": 54,
    "components": {}
  },
  "smartphone": {
    "price": 1500,
    "quantity": 16,
    "components": {
      "screen": "4K Gorilla Glass",
      "CPU": "High Efficiency CPU",
      "GPU": "High Efficiency GPU"
    },
    "tags": ["tech", "smartphone"],
    "in_stock": True
  },
  "laptop": {
    "price": 4000,
    "quantity": 0,
    "components": {
      "screen": "27in 8K",
      "CPU": "Best Tech CPU",
      "GPU": "Best Tech GPU"
    },
    "tags": ["tech", "computer"],
    "in_stock": False
  },
  "watch": {
    "price": 2000,
    "quantity": 54,
    "components": {
      "screen": "Little Screen Corp.",
      "CPU": "Downsize CPU",
      "GPU": "Downsize GPU"
    },
    "tags": ["tech", "smartwatch", "iot"],
    "in_stock": True
  }
}
```