

@import "../../../my_style.less"

PYTHON {.text_center}

DEVOIR MAISON {.text_center}

Notes et Commentaires {.text_center}

Notes

Nom	Note
Alexandre	15.00
Aréna	19.50
Lilou	20.00
Lucas	19.75
Lucie	20.00
Lyvahne	14.50
Marie	20.00
Mirana	20.00
Raphaël	18.25
Simon	18.75
Yann	19.75

Alexandre

15/20

Exercice	Note	Commentaire
debug.py	0.5 / 1	(0.375 arrondi à 0.5) Je n'ai pas trouvé le reste de l'exercice
list_easy.py	2.25 / 2	
list_medium.py	3 / 4	3 derniers jours de la semaine: days[3:6] affiche ["thursday", "friday", "saturday"], les 3 derniers jours sont: ["friday", "saturday", "sunday"]. Privilégie la méthode days[-3:] qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier

Exercice	Note	Commentaire
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	2 / 6	Ajout de données: tu ajoutes une <code>list</code> et non un <code>dict</code> . La fin de l'exercice était réalisable même en ne réussissant pas cette partie

Aréna

19.5/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	1.75 / 2	Ajoutez éléments de la liste <code>add_to_lst</code> à <code>lst_easy</code> : il fallait utiliser la méthode <code>.extend()</code> pour ajouter tous les éléments d'une liste dans une autre liste
<code>list_medium.py</code>	4 / 4	
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	5.5 / 6	"in_stock" de "laptop" à <code>False</code> : <code>False</code> est un <code>boolean</code> et non une <code>string</code> => <code>False</code> au lieu de " <code>False</code> "

Lilou

20/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	
<code>list_medium.py</code>	3.5 / 4	Afficher " <code>thursday</code> ": le concept est compris mais tu affiches " <code>tuesday</code> " à la place de " <code>thursday</code> "
		3 derniers jours de la semaine: c'est correct, mais privilégie l'utilisation de <code>days[-3:]</code> qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier

Exercice	Note	Commentaire
<code>dict_easy.py</code>	3 / 3	Tu utilises parfois <code>.update()</code> et parfois <code>dct[key]</code> pour mettre à jour la valeur d'une clef. Même si les deux méthodes font la même chose, essaye de ne t'en tenir qu'à une seule.
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	6.25 / 6	

Lucas

19.75/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	
<code>list_medium.py</code>	3 / 4	3 derniers jours de la semaine: <code>days[3:6]</code> affiche <code>["thursday", "friday", "saturday"]</code> , les 3 derniers jours sont: <code>["friday", "saturday", "sunday"]</code> . Privilégie la méthode <code>days[-3:]</code> qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	6.25 / 6	

Lucie

20/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	<code>lst_easy = lst_easy + add_to_lst</code> peut s'écrire <code>lst_easy += add_to_lst</code> mais ta façon est juste
<code>list_medium.py</code>	3 / 4	3 derniers jours de la semaine: c'est correct, mais privilégie l'utilisation de <code>days[-3:]</code> qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier

Exercice	Note	Commentaire
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	5.75 / 6	"in_stock" de "laptop" à False: False est un boolean et non une string => False au lieu de "False"

Lyvahne

14.50/20

Consigne non-respectée: Créez un fichier .py par exercice, ...

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>lst_easy.py</code>	1.5 / 2	<code>lst_easy = [1]</code> : tu crées une liste avec un élément, cela est différent de créer une liste puis d'ajouter un élément à cette liste. La solution était <code>lst_easy.append(1)</code>
		Bonus: La taille est de 4 car la liste <code>[4, 5]</code> compte pour un seul élément dans <code>lst_easy</code> .
<code>lst_medium.py</code>	3 / 4	3 derniers jours de la semaine: <code>days[3:6]</code> affiche <code>["thursday", "friday", "saturday"]</code> , les 3 derniers jours sont: <code>["friday", "saturday", "sunday"]</code> . Privilégie la méthode <code>days[-3:]</code> qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4 / 4	
<code>dict_hard.py</code>	2 / 6	Il manque une parenthèse sur ton <code>print()</code> ligne 113
		Ajoutez la clef "keyboard": tu ajoutes une list et non un dict => Voir corrections pour les éléments suivants
		Dans "watch" , créez la clef "tags": Comme tu peux le voir avec ton <code>print()</code> ligne 105, tu écrases tous les éléments de "watch" par une liste contenant <code>["tags", ["tech", "smartwatch", "iot"]]</code> . La solution était <code>products_hard["watch"].update({"tags": ["tech", "smartwatch", "iot"]})</code> ou <code>products_hard["watch"]["tags"] = ["tech", "smartwatch", "iot"]</code>

Exercice	Note	Commentaire
		Ajouter le tag "iot" au "tags" de "smartphone": Même commentaire que ci-dessus, tu écrases au lieu d'ajouter un élément => <code>products_hard["laptop"]["tags"].append("iot")</code>
		"screen" pour Little Screen Corp: <code>products_hard["watch"]</code> est un <code>dict</code> , les <code>dict</code> n'ont pas de méthodes <code>.replace()</code> , donc le terminal t'indique une erreur d'attribut => <code>products_hard["watch"]["components"].update({"screen": "Little Screen Corp."})</code>
		"in_stock" de "laptop" à False: "in_stock" est une clef de "laptop", qui est lui même une clef de <code>products_hard</code> => <code>products_hard["laptop"].update({"in_stock": False})</code> ou <code>products_hard["laptop"]["in_stock"] = False</code>

Marie

20/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	
<code>list_medium.py</code>	4 / 4	
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	Pas besoin de boucle <code>for</code> pour le bonus
<code>dict_hard.py</code>	6.25 / 6	Tu utilises parfois <code>.update()</code> et parfois <code>dct[key]</code> pour mettre à jour la valeur d'une clef. Même si les deux méthodes font la même chose, essaye de ne t'en tenir qu'à une seule.

Mirana

20/20

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2 / 2	Ajoutez les éléments de la liste <code>add_to_lst</code> : Il fallait utiliser la méthode <code>.extend()</code> qui ajoute tous les éléments d'une liste dans une autre liste

Exercice	Note	Commentaire
<code>list_medium.py</code>	4 / 4	
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	6.25 / 6	Bonus: tu utilises <code>,</code> et <code>+</code> pour afficher les string. Même si les deux opérateurs font la même chose, essaye de ne t'en tenir qu'à un seul.

Note: il n'est pas nécessaire `print()` à chaque fin de question

Raphaël

18.25

Consigne non-respectée: Créez un fichier `.py` par exercice, ...

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	<code>var_dict_b</code> : il manque les <code>:</code> entre une paire clef/valeur
<code>list_easy.py</code>	2.25 / 2	
<code>list_medium.py</code>	4 / 4	3 derniers jours de la semaine: c'est correct, mais privilégie l'utilisation de <code>days[-3:]</code> qui récupère tous les éléments à partir du 3ème en partant de la fin jusqu'au dernier
<code>dict_easy.py</code>	3 / 3	Tu utilises parfois <code>.update()</code> et parfois <code>dct[key]</code> pour mettre à jour la valeur d'une clef. Même si les deux méthodes font la même chose, essaye de ne t'en tenir qu'à une seule.
<code>dict_medium.py</code>	3 / 4	Ici <code>0</code> représente le premier caractère de la string => Ici <code>0</code> représente le premier caractère de la list prix de "smartphone" pour 1500: <code>products_medium["smartphone"] = [1500, 16]</code> change l'intégralité de la valeur, afin de ne changer que le prix => <code>products_medium["smartphone"][0] = 1500</code> Bonus: <code>if products_medium["smartphone"] == [1500, 0]</code> vérifie si <code>products_medium["smartphone"]</code> est une liste de 2 éléments avec <code>1500</code> à l'index 0 et <code>0</code> à l'index 1. Ce qui est différent de vérifier si le produit est en stock => <code>if products_medium["laptop"][1] == 0</code>

Exercice	Note	Commentaire
<code>dict_hard.py</code>	5 / 6	Dans "watch" , créez la clef "tags": <code>products_hard["watch"]["tags"] = ("tech", "smartwatch", "iot")</code> insère un <code>tuple</code> (structure de données similaire au <code>list</code>), il fallait insérer une liste avec <code>["tech", "smartwatch", "iot"]</code>
		"screen" pour Little Screen Corp.: <code>products_hard["watch"]["components"].clear()</code> supprime tous les éléments du dict, or il ne fallait changer que "screen", les autres éléments ne devaient pas être modifiés => <code>products_hard["watch"].update({"tags": ["tech", "smartwatch", "iot"]})</code>

Simon

18.75

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	<code>lst_easy = lst_easy + add_to_lst</code> peut s'écrire <code>lst_easy += add_to_lst</code> mais ta façon est juste
<code>list_medium.py</code>	4 / 4	
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	3.25 / 4	prix de "smartphone" pour 1500: <code>products_medium["smartphone"] = [1500, 16]</code> change l'intégralité de la valeur, afin de ne changer que le prix => <code>products_medium["smartphone"][0] = 1500</code>
		Bonus: tu n'as fait que la moitié du bonus, il faut afficher la quantité si celle-ci n'est pas égale à 0 => <code>else:</code> <code>print(products_medium["laptop"][1])</code>
<code>dict_hard.py</code>	4.75 / 6	Dans "keyboard" , ajoutez la clef "price": tu as oublié d'ajouter "price"
		Dans "keyboard" , ajoutez la clef "quantity": <code>products_hard["keyboard"]={"quantity":72}</code> écrases la valeur pour en créer une nouvelle, c'est pour ça que tu es obligé de réécrire "quantity":72 quand tu ajoutes "components"
		Bonus: pas besoin boucle <code>for</code> imbriquées, voir correction

Yann

19.75

Exercice	Note	Commentaire
<code>debug.py</code>	1 / 1	
<code>list_easy.py</code>	2.25 / 2	<code>lst_easy = lst_easy + add_to_lst</code> peut s'écrire <code>lst_easy += add_to_lst</code> mais ta façon est juste
<code>list_medium.py</code>	4 / 4	
<code>dict_easy.py</code>	3 / 3	
<code>dict_medium.py</code>	4.25 / 4	
<code>dict_hard.py</code>	5.25 / 6	<p>Affichez la quantité de "laptop" en stock: tu affiches la quantité de smartphone en stock: <code>products_hard["smartphone"]["quantity"]</code>. Le raisonnement est correct donc pas de points en moins, mais ça ne respecte pas la consigne</p> <p>"components" de "watch": Il fallait afficher les clefs/valeurs sans boucle <code>for</code> (nous ne les avons pas encore vu). Lucie a d'ailleurs posé la question sur Slack sur ce qu'il fallait afficher =></p> <pre>print(products_hard["watch"]["components"])</pre> <p>Dans "keyboard" , ajoutez la clef "price": <code>products_hard["keyboard"] = {"price": 500}</code> écrase le dict par <code>{"price": 500}</code>, tu n'ajoutes donc pas de clef mais crée un nouveau dict =></p> <pre>products_hard["keyboard"].update({"price": 500})</pre>