



Initiation à la programmation - 42

Jour 07

Staff 42 bocal@42.fr

Résumé: Ce document est le sujet du jour 07 de la piscine d'initiation à la programmation.

Table des matières

I	Consignes	2
II	Exercice 00 : get_names	3
III	Exercice 01 : your_namebook	4
IV	Exercice 02 : family_affairs	5
V	Exercice 03 : persons_of_interest	7


Chapitre I

Consignes

- A 42, vous allez faire l'expérience d'une pédagogie un peu particulière : vous avez un "cours" d'introduction d'1h tous les matins, et le reste de la journée, vous avez des exercices à réaliser en autonomie.
- Vous avez une question ? Un problème ? Un blocage ? Demandez à votre voisine de droite. Sinon, essayez avec votre voisine de gauche. A 42, les étudiants ne sont pas en compétition, ils avancent ensemble, en s'entre-aidant.
- Votre manuel de référence s'appelle **Google / man / Internet /** Vous allez devoir apprendre à faire des recherches sur internet, toutes les infos dont vous avez besoin s'y trouvent !
- Lisez attentivement les exemples. Vous devez respecter le formatage des réponses : les majuscules, les retours à la ligne... tout est important. Programmer, c'est avant tout faire preuve de rigueur.
- Un tuteur vous accompagne tout au long de la piscine : il/elle est là pour vous donner des pistes, vous indiquer comment faire vos recherches sur internet et vous encourager si vous êtes démotivées. Le tuteur est un soutien pour vous, mais il n'est pas là pour vous donner les réponses !
- A la fin de la journée, le tuteur de votre rangée va corriger votre travail collectivement. C'est un bon moment pour échanger et s'expliquer, entre élèves, les erreurs que vous avez pu faire, ou au contraire expliquer aux autres ce que vous avez compris. Soyez attentives.
- Bon courage, et n'ayez pas peur de vous tromper ! Faites des tests, tatonnez - en informatique, c'est en faisant des erreurs qu'on apprend !

Chapitre II

Exercice 00 : get_names

	Exercice : 00
Créer un dico des noms	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <code>get_names.rb</code>	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `get_names.rb`.
- Il va demander à l'utilisateur d'entrer des noms, et mettre ces noms dans un array.
- Quand l'utilisateur entrera "STOP", la boucle s'arrête, et l'array est affiché.
- Si le nombre de paramètres passés au script est différent de 0, le script affichera simplement "none".


```
?> ./get_names.rb "on avait dit pas d'argument !"
none
?> ./get_names.rb
Donne-moi un nom : Laurie Mezard
Merci, un autre nom ? Fred Grati
Merci, un autre nom ? Lydie Le Blevenec
Merci, un autre nom ? stop
Merci, un autre nom ? STOP
["Laurie Mezard", "Fred Grati", "Lydie Le Blevenec", "stop"]
?>
```



Vous avez déjà vu toutes ces notions !

Chapitre III

Exercice 01 : your_namebook

	Exercice : 01
Organise-moi un peu cette liste de noms !	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <code>your_namebook.rb</code>	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `your_namebook.rb`.
- Comme dans l'exercice précédent, vous allez construire un array avec des noms, mais cette fois à partir d'un hash.
- Vous allez d'abord déclarer un hash associant des prénoms à des noms.


```
personnes = {  
  "jean" => "valjean",  
  "grace" => "hopper",  
  "frederic" => "grati",  
  "fifi" => "brindacier"  
}
```

- Ensuite, votre script va construire l'array avec le nom complet des personnes, avec la première lettre en majuscule. Vous afficherez ensuite l'array.

```
?> ./your_namebook.rb  
["Jean Valjean", "Grace Hopper", "Frederic Grati", "Fifi Brindacier"]  
?>
```

Chapitre IV

Exercice 02 : family_affairs

	Exercice : 02
Des histoires de famille.	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : family_affairs.rb	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `family_affairs.rb`.
- Il contiendra une méthode `trouver_les_roux`.
- Cette méthode prend en paramètre un hash représentant les membres d'une famille avec leur prénom comme clef et leur couleur de cheveux comme attribut.
- Cette méthode utilisera la méthode `select` pour rassembler les prénoms des personnes rousses dans un nouvel array, qu'elle retournera.
- Ainsi le script suivant :

```
?> cat family_affairs.rb | cat -e
# your method definition here

famille_Dupont = {
  "mathieu" => :roux,
  "marie" => :blond,
  "virginie" => :brun,
  "gaetan" => :roux,
  "fred" => :roux
}

p trouver_les_roux(famille_Dupont)
```

aura la sortie :


```
?> ./family_affairs.rb | cat -e
["mathieu", "gaetan", "fred"]$
?>
```



Google ruby hashes, select, each_key, to_a.

Chapitre V

Exercice 03 : persons_of_interest

	Exercice : 03
Des personnes qui valent la peine d'être connues.	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : persons_of_interest.rb	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `persons_of_interest.rb`.
- Il contiendra une méthode `naissances_celebres`.
- Cette méthode prend en paramètre un hash représentant des personnes de l'histoire. Chaque entrée du hash est elle-même un hash avec les clefs `:nom` et `:date_de_naissance`.
- La méthode va trier le hash passé en paramètre dans l'ordre des dates de naissance, puis afficher chaque entrée (regardez l'exemple ci-dessous).
- Ainsi le script suivant :

```
?> cat persons_of_interest.rb | cat -e
# your method definition here

femmes_scientifiques = {
  :ada => { :nom => "Ada Lovelace", :date_de_naissance => "1815" },
  :cecilia => { :nom => "Cecila Payne", :date_de_naissance => "1900" },
  :lise => { :nom => "Lise Meitner", :date_de_naissance => "1878" },
  :grace => { :nom => "Grace Hopper", :date_de_naissance => "1906" }
}

naissances_celebres femmes_scientifiques
```

aura la sortie :

```
?> ./persons_of_interest.rb | cat -e
Ada Lovelace est une grande scientifique nee en 1815.$
Lise Meitner est une grande scientifique nee en 1878.$
Cecila Payne est une grande scientifique nee en 1900.$
Grace Hopper est une grande scientifique nee en 1906.$
?>
```




Google ruby hashes, sort_by.



Vous pouvez aussi googler les noms sus-cités, elles le valent bien !