



Initiation à la programmation - 42

Ruby - Jour 3

Staff 42 pedago@42.fr

Résumé: Ce document est le sujet du jour 3 du programme d'initiation à Ruby.

Table des matières

I	Consignes	2
II	Procédure de rendu : Git	3
III	Préambule	5
IV	Exercice 00 : my_first_method	6
V	Exercice 01 : greetings_for_all	7
VI	Exercice 02 : help_your_professor	8
VII	Exercice 03 : family_affairs	10
VIII	Exercice 04 : persons_of_interest	11

Chapitre I

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vous serez corrigés par les autres participants du programme d'initiation. C'est le "peer-correcting" de la pédagogie 42 !
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google / man / Internet /`
- Lisez attentivement les exemples. Les exercices pourraient bien requérir des choses qui y sont précisées, et non dans le sujet...

Chapitre II

Procédure de rendu : Git

- Votre rendu est collecté via un serveur distant. Cela signifie que vous devrez envoyer sur ce serveur votre repertoire de rendu avec vos exercices suivant une arborescence précise.
- Le logiciel utilisé pour ce faire s'appelle **git**.
- Git permet de faire beaucoup de choses, vous trouverez ci-dessous les principales commandes :
- **Git clone** vous permet de **cloner** un repertoire présent sur un serveur directement dans votre repertoire courant. Cela signifie que tous les fichiers présents que vous y aviez mis depuis un autre poste se retrouveront sur votre ordinateur et pourront être modifiés et réenvoyés avec les commandes git suivantes. Cette commande est aussi importante quand vous souhaitez vérifier ce que vous avez "pushé" sur le serveur, n'hésitez pas à en abuser.

```
?> git clone vogsphere@vogsphere.42.fr:piscine/truc/machin/creme
Cloning into 'piscine'[...]
?>
```

- **Git add** vous permet d'ajouter à une liste de fichiers surveillés un ou plusieurs fichiers/repertoires. Ils seront ajoutés en l'état ce qui veut dire que si vous les modifiez par la suite, il faudra les ajouter à nouveau pour mettre à jour la liste.

```
?> git add file directory
?>
```

- **Git commit -m** vous permet de "fixer" votre liste de fichiers surveillés afin de préparer leur envoi sur le serveur. Le message est obligatoire et permet d'avoir un historique des créations et modifications que vous avez effectué au fur et à mesure de votre travail.

```
?> git commit -m "Ajout des exercices X Y Z"
[master (root-commit) 4e8b2aa] Ajout des exercices X Y Z
3 file changed, 500 insertions(+)
create mode 100755 X Y Z
?>
```

- **Git push origin master**, la commande ultime, celle qui vous permet d'envoyer sur le serveur vos exercices qui y seront collectés par la Moulinette. Vérifiez bien son retour, le moindre message d'erreur signifie que vous n'avez rien pu envoyer et que par conséquent votre répertoire sur le serveur est vide (ce qui serait dommage pour votre note). Si vous avez le moindre doute, contactez un membre du staff 42.

```
?> git push origin master
[...]  
X files written.  
?>
```



L'adresse git de votre dépôt se trouve sur l'intranet et dans le fichier config du répertoire `.git` de votre dossier de rendu, c'est avec elle que vous pourrez "git clone".

Chapitre III


Préambule

Voici la liste des posters de motivation que l'on peut trouver dans le bureau de Barney Stinson au fil des saisons de *How I Met Your Mother* :

- Awesomeness : “When I get sad, I stop being sad and be Awesome Instead. True Story. Barney Stinson”
- Conformity : “It’s the one who is different that gets left out in the cold.”
- Courage : “True greatness comes when you’re tested. Theodore Roosevelt”
- Challenge : “We either find a way or we make one”
- Opportunity : “You will always miss 100% of the shots you don’t take.”
- Teamwork : “Coming together is the beginning. Keeping together is progress. Working together is success. Henry Ford.”
- Teamwork : “The chain is only as strong as the weakest link”
- Perseverance : “Continuous effort is the key to unlocking your potential. Sir Winston Churchill”
- Perfection : “It is not good enough to win, everybody else should lose”
- Strength : “What the mind can conceive, it can achieve”

Chapitre IV

Exercice 00 : my_first_method

	Exercice : 00
Baladons-nous dans l'array	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <code>my_first_method.rb</code>	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `my_first_method.rb` qui contient une méthode `upcase_it` qui prend une chaîne de caractère comme argument. Cette méthode doit retourner une version majuscule de la string, si et seulement si la chaîne de caractères fait plus de 10 caractères. Si la chaîne fait 10 caractères ou moins, la méthode renvoie `nil`.
- Vous appliquerez cette méthode, et afficherez son retour s'il n'est pas vide, sur les paramètres passés en ligne de commande. S'il n'y a aucun paramètre, affichez `none` suivi d'un retour à la ligne.


```
?> ./my_first_method.rb | cat -e
none$
?> ./my_first_method.rb "eheh" | cat -e
?> ./my_first_method.rb "hello world" "I'm happy to be here" | cat -e
HELLO WORLD$
I'M HAPPY TO BE HERE$
?>
```



Google ruby methods.

Chapitre V

Exercice 01 : greetings_for_all

	Exercice : 01
On dit bonjour à la dame!	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : greetings_for_all.rb	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script **greetings_for_all.rb** qui contient une méthode **greetings** qui prend un nom en paramètre et affiche un message de bienvenue avec ce nom. Si la méthode est appelée sans argument, son paramètre par défaut sera “noble stranger”. Si la méthode est appelée avec un argument qui n’est pas une chaîne de caractères, un message d’erreur devra être affiché à la place du message de bienvenue.
- Ainsi le script suivant :

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


```
greetings lucie
greetings
greetings 22
```

aura la sortie :

```
?> ./greetings_for_all.rb | cat -e
Hello, lucie.$
Hello, noble stranger.$
Error ! This was no name.$
?>
```


Chapitre VI

Exercice 02 : help_your_professor

	Exercice : 02
Un petit coup de pouce au prof...	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : help_your_professor.rb	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script `help_your_professor.rb`. Il contiendra une methode `average_mark` qui à partir d'un hash associant le prénom des étudiants à leur note à un devoir, calcule la moyenne de la classe pour ce devoir.
- Ainsi le script suivant :

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

BTS_3B = {
  "marine" => 18,
  "jean" => 15,
  "coline" => 8,
  "luc" => 9
}
BTS_3A = {
  "quentin" => 17,
  "julie" => 15,
  "marc" => 8,
  "stephanie" => 13
}
puts "Average mark for the 3B group: #{average_mark BTS_3B}."
puts "Average mark for the 3A group: #{average_mark BTS_3A}."
```

aura la sortie :


```
?> ./help_your_professor.rb | cat -e
Average mark for the 3B group: 12.$
Average mark for the 3A group: 13.$
?>
```



Google ruby hashes.

Chapitre VII

Exercice 03 : family_affairs

	Exercice : 03
Des histoires de famille.	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : family_affairs.rb	
Fonctions Autorisées : Toutes	
Remarques : n/a	

- Créez un script **family_affairs.rb**. Il contiendra une méthode `find_the_gingers` qui prend en paramètre un hash représentant les membres d'une famille avec leur prénom comme clef et leur couleur de cheveux comme attribut. La méthode doit rassembler les prénoms des personnes rousses dans un array et les retourner.
- 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 find_the_gingers famille_Dupont
```

aura la sortie :

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

Chapitre VIII

Exercice 04 : persons_of_interest

	Exercice : 04
Des personnes qui valent la peine d'être connues.	
Dossier de rendu : <i>ex04/</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 `great_births` qui prend en paramètre un hash représentant des personnes de l'histoire, chaque entrée du hash étant elle-même un hash avec les clefs `:name` et `:year_of_birth`, qui les trie dans l'ordre des dates de naissance et les affiche ainsi triées.
- Ainsi le script suivant :

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

women_in_science = {
  :ada => { :name => "Ada Lovelace", :year_of_birth => "1815" },
  :cecilia => { :name => "Cecila Payne", :year_of_birth => "1900" },
  :lise => { :name => "Lise Meitner", :year_of_birth => "1878" },
  :grace => { :name => "Grace Hopper", :year_of_birth => "1906" }
}

great_births women_in_science
```

aura la sortie :

```
?> ./persons_of_interest.rb | cat -e
Ada Lovelace is a great person born in 1815.$
Lise Meitner is a great person born in 1878.$
Cecila Payne is a great person born in 1900.$
Grace Hopper is a great person born in 1906.$
?>
```