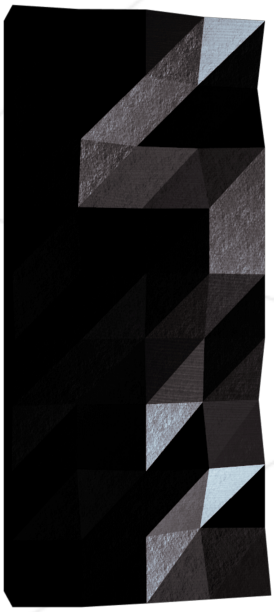

Initiation à la programmation - 42

Jour 04

Staff 42 bocal@42.fr

Résumé: Ce document est le sujet du jour 04 de la piscine d'initiation à la programmation.

Table des matières

Chapitre I

Consignes

- La procédure de correction se déroulera durant la dernière heure de la journée. Chaque personne corrigera une autre personne selon le model de peer-correcting.
- Vous avez une question ? Demandez à votre voisine de droite. Sinon, essayez avec votre voisine de gauche.
- Lisez attentivement les exemples. Les exercices pourraient bien requérir des choses qui y sont précisées, et non dans le sujet...
- Votre manuel de référence s'appelle `Google / man / Internet / ...`.

Chapitre II

Préambule

Voici une recette de `Choucroute à l'alsacienne` :

- Ingrédients (4 personnes)
  - 1kg de choucroute
  - 350g de lard fumé
  - 350g de palette
  - 1 càs de saindoux
  - 2 gousses d'ail
  - 1 feuille de laurier
  - 10 grains de genièvre
  - 1 oignon piqué avec 2 clous de girofle
  - 25cl de Riesling
  - 350g de pommes de terre
  - 4 saucisses de Strasbourg
- Préparation
  - Rincer la choucroute sous l'eau froide. L'égoutter, et en verser la moitié dans un faitout.
  - Incorporer le lard fumé et la palette, puis recouvrir du reste de choucroute.
  - Ajouter le saindoux, l'ail (Non pelé !), le genièvre, le laurier et l'oignon piqué.
  - Arroser de Riesling, et laisser cuire à couvert et à feu doux pendant 1 heure.
  - Ajouter les pommes de terre, et poursuivre la cuisson 50 minutes.
  - Incorporer les saucisses, puis laisser cuire encore 10 minutes.
  - Servir avec une quantité déraisonnable de bière.

Ce projet est plus facile si vous le réalisez après avoir mangé de la `Choucroute à l'alsacienne`

Chapitre III

Exercice 00 : table_mult

	Exercice : 00
Le retour des tables de multiplication	
Dossier de rendu : $ex00/$	
Fichiers à rendre : `table_mult.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `table_mult.rb` qui prend un paramètre.
- Ce paramètre est la table de multiplication que vous allez afficher. (Par exemple, si le paramètre est 2, il faut afficher la table de 2)
- Si le nombre de paramètres est différent de 1, vous afficherez "none" suivi d'un retour à la ligne.

```
?> ./table_mult.rb | cat -e
none$
?> ./table_mult.rb 8 | cat -e
0 x 8 = 0$
1 x 8 = 8$
2 x 8 = 16$
3 x 8 = 24$
4 x 8 = 32$
5 x 8 = 40$
6 x 8 = 48$
7 x 8 = 56$
8 x 8 = 64$
9 x 8 = 72$
?>
```

Comment faire l'exercice ? Commencez par stocker le paramètre sous forme numérique dans une variable. Faites ensuite une boucle while avec une autre variable qui prend les valeurs de 0 à 9...

Chapitre IV

Exercice 01 : play_with_arrays

	Exercice : 01
Manipuler des arrays	
Dossier de rendu : $ex01/$	
Fichiers à rendre : `play_with_arrays.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `play_with_arrays.rb`.
- Votre script va itérer sur un array de nombres (que vous définirez) et construire un nouvel array en ajoutant 2 à chaque valeur de l'array d'origine.
- Vous devez donc avoir deux arrays dans votre programme, celui d'origine et le nouveau que vous avez créé.
- A la fin, affichez les deux arrays à l'ecran en utilisant la méthode p plutôt que puts.
- Par exemple, si votre array d'origine est [2, 8, 9, 48, 8, 22, -12, 2], vous aurez la sortie suivante :

```
?> ./play_with_arrays.rb | cat -e
[2, 8, 9, 48, 8, 22, -12, 2]$
[4, 10, 11, 50, 10, 24, -10, 4]$
?>
```

`Google "méthode p en ruby", each`

Chapitre V

Exercice 02 : play_with_arrays++

	Exercice : 02
Manipuler des arrays, encore	
Dossier de rendu : *ex*02/	
Fichiers à rendre : `play_with_arrays.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Reprenez le script précédent, mais cette fois vous ne traiterez que les valeurs supérieures à 5 de l'array d'origine.
- Par exemple, si votre array d'origine est [2, 8, 9, 48, 8, 22, -12, 2], vous aurez la sortie suivante :

```
?> ./play_with_arrays.rb | cat -e
[2, 8, 9, 48, 8, 22, -12, 2]$
[10, 11, 50, 10, 24]$
?>
```

`Google "méthode p en ruby", each`

Chapitre VI

Exercice 03 : my_first_method

	Exercice : 03
Baladons-nous dans l'array	
Dossier de rendu : $ex03/$	
Fichiers à rendre : `my_first_method.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `my_first_method.rb`.
- Vous devez définir dans ce script une méthode. Cette méthode s'appelle `upcase_it`.
- La méthode `upcase_it` prend une chaîne de caractère comme argument. Elle doit retourner cette chaîne de caractère en majuscules.
- Vous appliquerez cette méthode, et afficherez son retour, 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 "hello world" "I'm happy to be here" | cat -e
HELLO WORLD$
I'M HAPPY TO BE HERE$
?>
```

`Cherchez sur Google : "définition d'une méthode en ruby", each`

Procédez en deux étapes. D'abord, regardez sur internet comment on définit une méthode, et créez la méthode upcase_it. Une fois que vous avez défini votre méthode, itérez avec each sur le tableau ARGV et appliquez votre méthode à chaque paramètre.

Chapitre VII

Exercice 04 : greetings_for_all

	Exercice : 04
On dit bonjour à la dame !	
Dossier de rendu : $ex04/$	
Fichiers à rendre : `greetings_for_all.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `greetings_for_all.rb`.
- Vous devez définir dans ce cripte une méthode. Cette méthode s'appelle `greetings`.
- La méthode `greetings` 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 inconnue".
- 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
# definition de votre methode ici

greetings lucie
greetings
greetings 22
```

affichera :

```
?> ./greetings_for_all.rb | cat -e
Hello, lucie.$
Hello, noble inconnue$
Erreur ! Ce n'etait pas un nom.$
?>
```

`Cherchez sur Google: is_a, paramètre par défaut.`