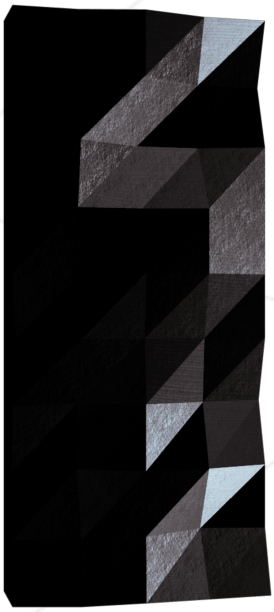

Initiation à la programmation - 42

Jour 01

Staff 42 bocal@42.fr

Résumé: Ce document est le sujet du jour 01 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 ce que `Wikipédia` a à dire à propos de la loutre :

`La Loutre d'Europe ou Loutre européenne (Lutra lutra), souvent qualifiée de loutre commune dans les pays d'Europe où elle est présente, est un mammifère carnivore semi-aquatique et principalement nocturne, de la famille des Mustélidés (sous-famille Lutrinés). Elle est l'une des trois espèces de loutres se rattachant au genre Lutra. En France, on ne trouve que cette seule espèce de loutre.`

`Sa hauteur est d'environ 30 cm au garrot. Son pelage, brun foncé, est composé de deux couches : le poil de bourre, court, très fin, dense et laineux ; et le poil de jarre, long, lisse, brillant et imperméable.`

`Excellente nageuse, elle dispose de pattes palmées, d'un corps allongé (60 à 80 cm en moyenne, auquel il faut ajouter une queue épaisse à la base et s'effilant vers l'extrémité de 30 à 40 cm de longueur), pour un poids allant de 5 à 15 kg.`

`Elle vit au bord des cours d'eau (ruisseaux, rivières et même fleuves), jusqu'à une altitude de 1300 m, dans les marais et parfois sur les côtes marines. Elle est habituellement solitaire, occupant un territoire de 5 à 15 km de rives le long d'un cours d'eau (parfois davantage) ou de 20 à 30 km2 en zone de marais. Elle emprunte régulièrement les mêmes passages sur la berge pour se mettre à l'eau : les "coulées". Lorsqu'elle sort de l'eau, elle se roule dans l'herbe pour essuyer sa fourrure, sur des zones reconnaissables à l'herbe couchée et appelées "places de ressui".`

`Elle fait sa tanière (qu'on appelle une "catiche", de l'ancien français "se catir" = se blottir, se cacher) entre les racines des arbres des berges des cours d'eau ou dans d'autres cavités (cavité rocheuse, tronc creux, terrier d'une autre espèce). La catiche contient souvent une entrée plus ou moins dissimulée au-dessous du niveau d'eau et un conduit d'aération.`

C'est mignon, une loutre.

Chapitre III

Exercice 00 : 42

	Exercice : 00
Les vraies savent afficher 42	
Dossier de rendu : $ex00/$	
Fichiers à rendre : `42.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `42.rb` qui, lorsqu'on l'exécute, affiche "42" suivi d'un retour à la ligne.

```
?> ./42.rb | cat -e
42$
?>
```

Ca vous parait simple ? C'est normal.

Le ruby est un langage très proche de l'anglais, cela rend les recherches plus faciles.

Chapitre IV

Exercice 01 : name

	Exercice : 01
Afficher un nom	
Dossier de rendu : $ex01/$	
Fichiers à rendre : `name.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `name.rb` dans lequel vous définirez une variable first_name et une variable last_name, initialisées respectivement avec votre prénom et votre nom, et qui les affichera suivies d'un retour à la ligne.

```
?> ./name.rb | cat -e
Frederic Grati$
?>
```

Chapitre V

Exercice 02 : name++

	Exercice : 02
Afficher un nom, en mieux	
Dossier de rendu : *ex*02/	
Fichiers à rendre : `name.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Modifiez le script name.rb : votre programme affichera la même chose, mais cette fois votre nom et votre prénom seront d'abord concaténés et affectés à une troisième variable.

```
?> ./name.rb | cat -e
Frederic Grati$
?>
```

`Google ruby strings.`

Chapitre VI

Exercice 03 : what's your name

	Exercice : 03
Comment que tu t'appelles ?	
Dossier de rendu : $ex03/$	
Fichiers à rendre : `whatsyourname.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez le script whatsyourname.rb qui demande d'abord à l'utilisateur d'entrer son nom, puis son prénom, et enfin affiche les deux.

```
?> ./whatsyourname.rb
Hey, what's your first name ? : Laurie
And your last name ? : Mezard
Well, pleased to meet you Laurie Mezard.
?>
```

`Google gets, chomp.`

Chapitre VII

Exercice 04 : aff_first_param

	Exercice : 04
Afficher un paramètre	
Dossier de rendu : *ex*04/	
Fichiers à rendre : `aff_first_param.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `aff_first_param.rb` qui, lorsqu'on l'exécute, affiche la première chaîne de caractères passée en paramètre suivie d'un retour à la ligne. S'il n'y a aucun paramètre, affichez `none` suivi d'un retour à la ligne.

```
?> ./aff_first_param.rb | cat -e
none$
?> ./aff_first_param.rb "WiFilles" "Numerique" "Enpowerment" | cat -e
WiFilles$
?>
```

`Google ARGV, array, condition if.`

Chapitre VIII

Exercice 05 : age

	Exercice : 05
Recupérer et modifier un nombre	
Dossier de rendu : $ex05/$	
Fichiers à rendre : `age.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `age.rb` qui demande à l'utilisateur d'entrer son âge, puis affiche quel âge aura l'utilisateur dans 10 ans, 20 ans et 30 ans.

```
?> ./age.rb
Please tell me your age : 15
You are currently 15 years old.
In 10 years, you'll be 25 years old.
In 20 years, you'll be 35 years old.
In 30 years, you'll be 45 years old.
?>
```

`Google string to_i.`

Chapitre IX

Exercice 06 : UPCASE_IT

	Exercice : 06
Voir en grand	
Dossier de rendu : $ex06/$	
Fichiers à rendre : `upcase_it.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `upcase_it.rb` qui prend une chaîne de caractères en paramètre. Lorsqu'on l'exécute, le script affiche la chaîne de caractères en majuscules suivie d'un retour à la ligne. Si le nombre de paramètres est différent de 1, affichez `none` suivi d'un retour à la ligne.

```
?> ./upcase_it.rb | cat -e
none$
?> ./upcase_it.rb "wi-filles" | cat -e
WI-FILLES$
?> ./upcase_it.rb 'CeT eXeRcIcE eSt AsSeZ fAcIlE !' | cat -e
CET EXERCICE EST ASSEZ FACILE !$
?>
```

`Google upcase.`

Chapitre X

Exercice 07 : downcase_it

	Exercice : 07
Voir plus petit	
Dossier de rendu : $ex07/$	
Fichiers à rendre : `downcase_it.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `downcase_it.rb` qui prend une chaîne de caractères en paramètre. Lorsqu'on l'exécute, le script affiche la chaîne de caractères en minuscules suivie d'un retour à la ligne. Si le nombre de paramètres est différent de 1, affichez `none` suivi d'un retour à la ligne.

```
?> ./downcase_it.rb | cat -e
none$
?> ./downcase_it.rb "PONEY" | cat -e
poney$
?> ./downcase_it.rb 'CeT eXeRcIcE eSt AsSeZ fAcIlE !' | cat -e
cet exercice est assez facile !$
?>
```

Vous devriez y arriver seules.

Chapitre XI

Exercice 08 : scan_it

	Exercice : 08
Scanner un texte	
Dossier de rendu : *ex*08/	
Fichiers à rendre : `scan_it.rb`	
Fonctions Autorisées : `Toutes`	
Remarques : `n/a`	

- Créez un script `scan_it.rb` qui prend deux paramètres. Le premier est un mot clé à chercher dans une chaîne. Le deuxième est la chaîne à parcourir. Lorsqu'on l'exécute, le programme affiche le nombre d'occurences du mot clé dans la chaîne. Si le nombre de paramètres est différent de 2 ou que la première chaîne n'apparaît pas dans la deuxième, affichez `none` suivi d'un retour à la ligne.

```
?> ./scan_it.rb | cat -e
none$
?> ./scan_it.rb "les" | cat -e
none$
?> ./scan_it.rb "les" "les exercices du J01 ne sont pas les plus difficiles" | cat -e
3$
?>
```

Essayez "string.scan(string)".