



## Ruby - Cellule 02

### Les conditions

Staff Pédago 42 [pedago@42.fr](mailto:pedago@42.fr)

*Résumé: Dans cette cellule, nous voyons comment utiliser les conditions.*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Consignes générales</b>	<b>3</b>
<b>III</b>	<b>Exercice 00 : Je suis null ?</b>	<b>4</b>
<b>IV</b>	<b>Exercice 01 : Je suis négatif ?</b>	<b>5</b>
<b>V</b>	<b>Exercice 02 : Mots de passe</b>	<b>6</b>
<b>VI</b>	<b>Exercice 03 : Multiplication</b>	<b>7</b>

# Chapitre I

## Préambule

Le jeu du Sirop selon Perceval, tiré de la série Kaamelott :  
"Bon, j'veais vous apprendre les règles simplifiées, parce que les vraies règles, elles sont velues. Bon, le seul truc, c'est que normalement, ça se joue à trois. Mais c'est pas grave on va se débrouiller.

Le principe, c'est de faire des valeurs. Donc là, mettons, on est trois, il y a trois valeurs à distribuer. On va dire, sirop de huit, sirop de quatorze et sirop de vingt-et-un. Vous occupez pas des sirops tout de suite. Ce qu'il faut comprendre d'abord, c'est les valeurs. Si vous lancez une valeur en début de tour, mettons un sirop de huit, pour commencer petit, les autres ont le choix entre laisser filer la mise ou relancer un sirop de quatorze. On tourne dans le sens des valeurs. C'est pour ça, il faut bien comprendre le système des valeurs ; après, ça va tout seul.

Bon alors mettons que j'ouvre avec un sirop de huit.  
Si c'est vous qu'avez siroté au tour d'avant, ça tourne dans votre sens. Alors soit vous laissez filer, vous dites "file-sirop", soit vous vous sentez de relancer et vous annoncez un sirop de quatorze. Comme on a commencé les annonces, le second joueur a pas le droit de laisser filer. Vous pouvez soit relancer un sirop de vingt-et-un, soit vous abandonnez le tour et vous dites "couche-sirop" ou "sirop Jeannot", ça dépend des régions. Et après, soit on fait la partie soit je fais un "contre-sirop" ! Et à partir de là, sirop de pomme sur vingt-et-un donc on fait la partie en quatre tours jusqu'à qu'il y en ait un qui sirote.

À la gagne, il n'y a que trois possibilités : soit vous faites votre sirop de huit, vous dites "beau sirop" et on recompte, soit vous faites votre sirop de quatorze, vous dites "beau sirop, sirop gagnant" et on vous rajoute la moitié, soit vous faites votre sirop de vingt-et-un et vous dites "beau sirop, mi-sirop, siroté, gagne-sirop, sirop-grelot, passe-montagne, sirop au bon goût".

Normalement ça se joue avec des cartes mais si vous avez que des dés, vous pouvez aussi jouer avec des dés puisque ce qui compte c'est les valeurs."

Au moins un des exercices suivants n'a aucun rapport avec le jeu du Sirop.

# Chapitre II


## Consignes générales

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vos exercices seront évalués par vos camarades de Piscine.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre répertoire lors de la peer-évaluation.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les **man** ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

# Chapitre III

## Exercice 00 : Je suis null ?


	Exercice : 00
Je suis null ?	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>iszero.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme **iszero.rb**
- Ce programme doit être exécutable. (pensez aux permissions notamment)
- Lorsqu'on l'exécute, le programme doit demander de rentrer un nombre.
- Si le nombre est égal à zéro, le programme doit afficher "Ce nombre est egal a zero".
- si le nombre est différent de zéro, le programme doit afficher "Ce nombre est different de zero"

```
?> ./iszero.rb
42
Ce nombre est different de zero
?> ./iszero.rb
0
Ce nombre est egal a zero
?>
```

# Chapitre IV

## Exercice 01 : Je suis négatif?


	Exercice : 00
Je suis négatif?	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>isneg.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme **isneg.rb**
- Ce programme doit être exécutable.
- Lorsqu'on l'exécute, le programme doit demander de rentrer un nombre.
- Si le nombre est négatif, le programme doit afficher "Ce nombre est negatif".
- Si le nombre est positif, le programme doit afficher "Ce nombre est positif"
- Si le nombre est égale a zéro, le programme doit afficher "Ce nombre est positif et negatif"

```
?> ./isneg.rb
42
Ce nombre est positif
?> ./isneg.rb
-42
Ce nombre est negatif
?> ./isneg.rb
0
Ce nombre est positif et negatif
?>
```

# Chapitre V

## Exercice 02 : Mots de passe

	Exercice : 02
Access denied	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <b>password.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme **password.rb**
- Ce programme doit être exécutable.
- Le programme doit avoir une variable contenant un mots de passe.


```
password = "Le ruby c'est cool"
```

- Lorsqu'on l'exécute, le programme doit demander de rentrer un mots de passe.
- Si le mot de passe est correct, le programme doit affiché "ACCESS GRANTED" sinon "ACCESS DENIED"

```
?> ./password.rb
1234
ACCESS DENIED
?> ./password.rb
Le ruby c'est cool
ACCESS GRANTED
?>
```

# Chapitre VI

## Exercice 03 : Multiplication

	Exercice : 03
Multiplication	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <b>mult.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme **mult.rb**
- Ce programme doit être exécutable.
- Lorsqu'on l'exécute, le programme doit demander de rentrer 2 nombres.
- Le programme va afficher si le résultat de la multiplication des 2 nombre est positif, négatif ou nul
- Puis le programme va afficher le résultat de la multiplication.

```
$ ./mult.rb
Entrez un premier nombre:
42
Entrez un deuxieme nombre:
42
42 x 42 = 1764
le resultat est positif

$ ./mult.rb
Entrez un premier nombre:
78
Entrez un deuxieme nombre:
-1
78 x -1 = -78
le resultat est negatif

$ ./mult.rb
Entrez un premier nombre:
72
Entrez un deuxieme nombre:
0
72 x 0 = 0
le resultat est positif et negatif
```