



# Ruby - Cellule 03

## Les boucles

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

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

# 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 : to25</b>	<b>4</b>
<b>IV</b>	<b>Exercice 01 : table_mult</b>	<b>5</b>
<b>V</b>	<b>Exercice 02 : i_got_that</b>	<b>6</b>
<b>VI</b>	<b>Exercice 03 : advanced mult</b>	<b>7</b>

# Chapitre I

## 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
- 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 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 : to25

	Exercice : 00
Allons jusqu'à 25 !	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>to25.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme `to25.rb`.
- Ce programme doit être exécutable.
- Ce programme accepte un input de l'utilisateur. Cet input est un nombre, vous le stockerez dans une variable sous forme numérique.
- Vous allez ensuite faire une boucle qui affiche tous les nombres, du nombre fourni jusqu'à 25.
- Si l'input est supérieur à 25, vous afficherez "Erreur" suivi d'un retour à la ligne.


```
?> ./to25.rb
Entrez un nombre inferieur a 25
45
Erreur
?> ./to25.rb
Entrez un nombre inferieur a 25
20
Dans la boucle, ma variable vaut 20
Dans la boucle, ma variable vaut 21
Dans la boucle, ma variable vaut 22
Dans la boucle, ma variable vaut 23
Dans la boucle, ma variable vaut 24
Dans la boucle, ma variable vaut 25
?>
```



While.

# Chapitre IV

## Exercice 01 : table\_mult


	Exercice : 01
Le retour des tables de multiplication	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>table_mult.rb</b>	
Fonctions Autorisées : <b>Toutes</b>	

- Créez un programme `table_mult.rb`.
- Ce programme doit être exécutable.
- Ce programme accepte un input de l'utilisateur. Cet input est un nombre, vous le stockerez dans une variable sous forme numérique.
- Ce nombre est la table de multiplication que vous allez afficher. (Par exemple, si l'input est 2, il faut afficher la table de 2)

```
?> ./to25.rb
Entrez un nombre
8
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
?>
```

# Chapitre V

## Exercice 02 : i\_got\_that

	Exercice : 02
Do you got it ?	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <i>i_got_that.rb</i>	
Fonctions Autorisées : Toutes	

- Créez un programme `i_got_that.rb`.
- Ce programme doit être exécutable.
- Ce script doit contenir une boucle `while` qui accepte un input de l'utilisateur, écrit une phrase en retour, et s'arrête uniquement lorsque l'utilisateur a entré "STOP".
- Chaque tour de boucle doit accepter un input de l'utilisateur.


```
?> ./i_got_that.rb
What you gotta say ? : Hello
I got that ! Anything else ? : I like ponies
I got that ! Anything else ? : stop...
I got that ! Anything else ? : STOP
?>
```



While, break.

# Chapitre VI

## Exercice 03 : advanced mult

	Exercice : 03
Le retour du retour des tables de multiplication	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <b>advanced_mult.rb</b>	
Fonctions Autorisées : Toutes	

- Créez un programme `advanced_mult.rb`.
- Ce programme doit être exécutable.
- Ce programme va afficher toutes les tables de multiplication sous la forme suivante

```
?> ./advanced_mult.rb "yolo" | cat -e
none$
?> ./advanced_mult.rb
Table de 0: 0 0 0 0 0 0 0 0 0 0 0
Table de 1: 0 1 2 3 4 5 6 7 8 9 10
Table de 2: 0 2 4 6 8 10 12 14 16 18 20
Table de 3: 0 3 6 9 12 15 18 21 24 27 30
Table de 4: 0 4 8 12 16 20 24 28 32 36 40
Table de 5: 0 5 10 15 20 25 30 35 40 45 50
Table de 6: 0 6 12 18 24 30 36 42 48 54 60
Table de 7: 0 7 14 21 28 35 42 49 56 63 70
Table de 8: 0 8 16 24 32 40 48 56 64 72 80
Table de 9: 0 9 18 27 36 45 54 63 72 81 90
Table de 10: 0 10 20 30 40 50 60 70 80 90 100
?>
```

- Vous n'avez le droit qu'à deux boucles while.