

NSI 1ere programmation

Présentation de REPL.it

qkzk

REPL

L'acronyme **Read-Eval-Print loop** désigne une interface dans laquelle on peut écrire une commande, l'exécuter, voir s'afficher une réponse et recommencer.

En programmation on parle souvent **console** et dans l'utilisation d'un système unix de **shell** ou d'interface en ligne de commande **CLI**.

repl.it est un site fondé par la startup américaine du même nom que nous allons utiliser cette année pour les exercices sur Python.

Certains d'entre vous connaissent déjà repl et vont à nouveau pester contre son interface. . .

Nous allons principalement utiliser les exercices en ligne sans trop nous intéresser aux autres fonctionnalités de repl.

repl permet aussi de réaliser un programme en ligne dans de nombreux langages.

Si vous avez accès à internet, vous pouvez développer dans repl.

Exemple de programme en ligne

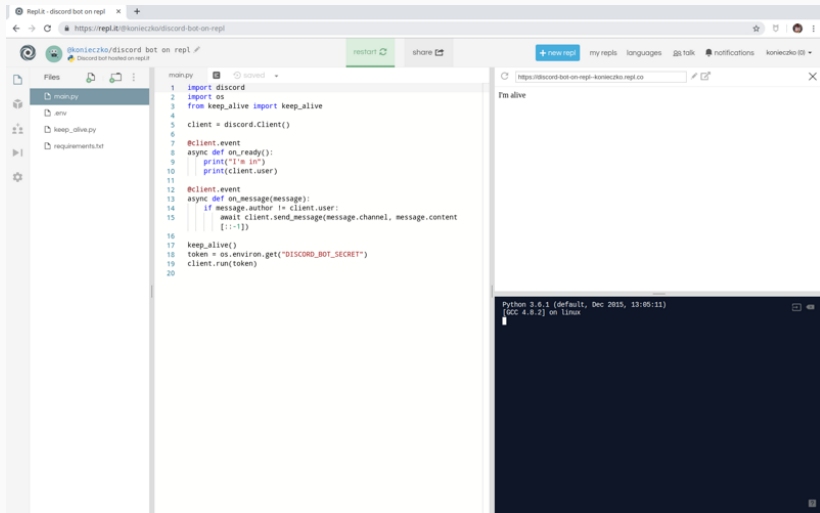


Figure 1: myRepl

Exemple de programme en ligne

La fenêtre se découpe en 4 parties :

1. tout à gauche Files présente les différents fichiers
2. La grande colonne blanche présente les sources du fichier actuellement ouvert
3. A droite on trouve les sorties. Ici ce programme comporte à la fois un site web (blanc) et une sortie console (noir).

Que fait ce programme ?

C'est en fait un bot discord (qui ne fait pas grand chose).

Le site web force repl.it à laisser le bot tourner tant qu'un utilisateur se connecte toutes les heures.

J'utilise donc un autre outil en ligne pour rafraîchir une page toutes les 30 minutes et le programme tourne presque indéfiniment.

Il se relance chaque fois que repl.it installe une mise à jour majeure.

Repl permet de partager des exercices auto corrigés. Vous avez un énoncé, tentez de répondre et soumettez votre réponse. Dès qu'elle est validée, vous pouvez passer au suivant.

Exemple

Interface

The screenshot displays the repl.it web interface for a student. The browser address bar shows the URL `https://repl.it/teacher/assignments/3205982/preview`. The page title is "1.1. Input/print: somme de trois nombres (preview)". The interface is divided into three main sections:

- Code Editor:** Contains a Python script with the following code:

```
1- # Ce programme lit deux nombres en entrée et imprime leur somme :
2 a = int(input())
3 b = int(input())
4 # c =
5
6 # Pouvez-vous le modifier afin qu'il imprime la somme de trois nombres ??
7
8 print(a + b)
9
```
- Terminal:** Shows the output of the Python interpreter:

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
> |
```
- Instructions Panel:** Contains the following text:
 - Due: --** with a green "submit" button.
 - Instructions from your teacher:**
 - Statement:** "Ecrire un programme qui prenne trois nombres et imprime leur somme. Chaque nombre est donné sur une ligne à part."
 - Exemple d'entrée:** A text input field containing the numbers "2", "3", and "6" on separate lines.
 - Exemple de sortie:** A text input field containing the number "11".
 - Théorie:** "Si vous ne savez pas par où aborder cette consigne, lisez un résumé de la leçon : https://snakify.org/lessons/print_input_numbers/. Vous pouvez aussi lire un détail étape par étape."

Figure 2: repl_student

L'interface comporte donc 3 parties et 2 boutons importants

- Haut gauche : le code source de votre programme
- Bas gauche : la console Python dans laquelle vous pouvez le tester
- Droite : la consigne
- le bouton RUN : permet d'exécuter votre script pour l'essayer
- le bouton vert SUBMIT : envoie votre programme et le soumet à des tests. S'ils passent tous, vous pouvez avancer. Sinon on vous dit pourquoi ils échouent.

L'intérêt de repl est manifeste : vous progressez sans que je doive valider chacune de vos réponses. . .

L'inconvénient est que nous allons devoir utiliser deux fonctions qu'il faut “normalement” éviter : `input` et `print`

Ce ne sont des fonctions qui ne servent qu'à **lire depuis la console** et **afficher dans la console**.

Les entrées et sorties d'un programme traditionnel ne sont pas “`input`” et “`print`”

Les consignes présentent **toujours un exemple issu des tests.**

S'inscrire et rejoindre la classe

1. Dans le navigateur rendez-vous sur repl.it
2. Si vous n'avez pas encore de compte, cliquez sur **sign up**.
Utilisez google et le mail fourni pour classroom
3. Si vous avez déjà un compte : **login**

Vous devez vous inscrire dans une **classe** en mode **student**. La dernière page du diaporama contient le lien d'invitation vers la classe.

Il vous suffit de le copier dans votre navigateur Chrome.

Vous aurez alors accès à cette page dans laquelle vous pouvez cliquer sur la classe et choisir un exercice.

Réaliser un exercice ensemble

la consigne est simple :

Statement

Ecrire un programme qui prenne trois nombres et imprime leur somme. Chaque nombre est donné sur une ligne à part.

Exemple d'entrée

2

3

6

Exemple de sortie

11

Réaliser un exercice ensemble

Le code source se présente ainsi :

```
# Ce programme lit deux nombres  
# en entrée et imprime leur somme :  
a = int(input())  
b = int(input())  
# c =  
  
# Pouvez-vous le modifier afin qu'il imprime  
# la somme de trois nombres ??  
  
print(a + b)
```

Si on soumet maintenant les tests vont échouer

Réaliser un exercice ensemble

Le code source se présente ainsi :

```
# Ce programme lit deux nombres  
# en entrée et imprime leur somme :  
a = int(input())  
b = int(input())  
c = int(input())  
  
# Pouvez-vous le modifier afin qu'il imprime  
# la somme de trois nombres ??  
  
print(a + b + c)
```

Réaliser un exercice ensemble

Cliquer sur **run**, taper 3 nombres dans la console :

- 1 ENTREE
- 2 ENTREE
- 3 ENTREE

On voit alors apparaître leur somme. C'est juste !

On clique sur **submit** et les tests sont validés.

1. Lire la consigne
2. Comprendre la consigne (c'est mieux)
3. Essayer jusqu'à réussir
4. Soumettre

Lien d'invitation vers la classe

Lien d'invitation :

<https://repl.it/classroom/invite/9jALoq8>