

Initiation à Python

L'émulateur se trouve dans le répertoire `Ma classe\Documents en consultation\maths\Python1-5\` et ouvrez le fichier `simulator.html` dans Firefox ou Chrome.

Le langage BF

Imaginez un ruban de cases sur lequel se déplace une tête de lecture/écriture. Les cases contiennent des nombres entiers. Un programme contrôle la tête :

- + ajoute 1 à la cellule courante et - lui soustrait 1.
- > déplace la tête d'une case vers la droite et < l'inverse.
- : (deux points) affiche le nombre dans la case.
- ! affiche l'état du ruban et la position de la tête.

Il y a six autres commandes qu'on n'expliquera pas ici. Je ne demande pas d'expliquer les programmes en BF.

Tapez `bf("+++:", bf("++++!>+++++! [<+>-]<!"))` et `bf("+++++! [>++++++<-!])`.

Qu'avez-vous calculé dans les deux derniers ?

En bonus plus difficile, tapez sans erreur :

```
bf("+++++[[>]+[<]>-]>>[++]<[<]>!>[<]>+<-]>>[+]<[<]>!>])
```

Qu'avez-vous calculé ?

Remboursement d'emprunt

M. X contracte un emprunt de 150000 € au taux annuel de 1,36 %. Il rembourse une mensualité de 921,69 €.

Le script `emprunt.py` contient la fonction `emprunt` qui a trois arguments : `montant`, `taux` et `mensualite`.

Elle s'utilise ainsi : `emprunt(mensualite=999.1)` et renvoie le nombre de mois nécessaires pour rembourser l'emprunt avec une mensualité de 999,10 € et 0.

Vous pouvez omettre les valeurs par défaut (comme ci-dessus où `taux=1.36` et `montant=150000` sont omises).

Déterminez au bout de combien de mois il aura remboursé son emprunt.

En tâtonnant, déterminez une mensualité à payer pour qu'il rembourse l'emprunt de 150000 € au taux de 1,55 % sur 20 ans (240 mois exactement).

M. X aimerait savoir combien lui ont coûté les intérêts, c'est-à-dire combien il a payé en plus des 150000 €.

À l'aide du dernier `restantdu`, déterminer la dernière mensualité (ce n'est pas 921,69 €).

À l'aide du calcul précédent, modifiez le script pour qu'`emprunt()` affiche aussi le coût du crédit et renvoie (180, 15792.15) au lieu de (180, 0).

Pour arrondir valeur à deux chiffres après la virgule, on utilise `round(valeur, 2)`.