

## Projet « BankNum »

Vous allez créer un programme simulant une banque numérique, en revoyant des bases de programmation.

Pour cela : - Lisez les « Point info » et tester les exemples avec "Edupython" ou l'éditeur en ligne [Python3trinket](#)

- Faites les versions successives du programme «BankNum» et les enregistrer.

### A. Création d'un compte !

#### Point info

**Un algorithme** est une suite d'instructions simples amenant à la résolution d'un problème.

On peut le décomposer en 3 phases : - **L'initialisation des variables**

- **Le traitement du problème**

- **L'affichage des résultats.**

**La programmation** est la traduction d'un algorithme en langage informatique : Scratch, Python, C++, ...

#### Exemple à tester :

Voici un premier programme Python : analysez le.

Répondez aux questions ci-dessous même si des détails vous échappent encore (nous y reviendront \*) :

- Que fait ce premier programme ?

- Associer chaque ligne de code à la bonne phase d'algorithme.

```
1 cube=0
2 nom=input('Prénom?')
3 nombre=int(input('Bonjour '+nom+'. Donne un nombre'))
4
5 cube=nombre**3
6
7 print('Le cube de '+str(nombre)+' est '+str(cube))
```

Réponses

Vérifier vos réponses en tapant et en exécutant le programme.

#### Point info\*

**Une variable** est un emplacement de mémoire informatique, avec un nom, contenant principalement :

- **des nombres** entiers ou décimaux, notés **int** et **float** en Python.

- **du texte** ou chaîne de caractères, reconnaissable aux **guillemets**, notés **str** en Python

Le contenu d'une variable, peut **varier** au fil programme.

**Le signe =** affecte une valeur à une variable : var=12, var=5.3 , var='bonjour'

**input()** affiche une demande à l'utilisateur, la réponse est considérée par défaut comme du texte.

**int(), ou float()** converti du texte en nombre entier, ou décimal : int('12') vaut 12, et float('12') vaut 12.0

**str()** converti un nombre en texte.

**L'opération +** Peut **additionner** deux nombres : *print(2+3) affichera 5.*

Peut **additionner** du texte par collage : *print("You"+"pi") affichera Youpi.* On dit "**concaténer**".

Confusion fréquente : *print('2'+ '3')* affiche 23 : concaténation de deux caractères '2' et '3'

**print()** affiche le contenu d'une variable

**Pour la suite :** Les noms **en gras** désignent les variables à utiliser. Quand ils sont **en italique** il s'agit de leur contenu.

#### Programme BankNum.A :

Un utilisateur est client de la banque BankNum dans laquelle il a initialement 1000€ sur un **compte** courant.

Vous devez faire un programme qui affiche «Bienvenu chez BankNum : Quel est votre nom ? ».

Celui-ci doit enregistrer la réponse dans une variable **nom** et afficher : « Bonjour **nom**, quelle somme voulez vous déposer ? »

Cette nouvelle réponse doit être enregistrée dans une variable **somme**.

Le programme doit alors ajouter la **somme à compte** et afficher « Vous avez **compte** € sur votre compte »

Enregistrer votre programme BankNum.A

Réponse

## B. Proposer un Choix

« BankNum » veut proposer à l'utilisateur le choix entre un dépôt ou bien un retrait d'argent.

### Point Info

Une instruction conditionnelle	Algorithme	Python
est une condition qui peut être <b>vraie ou fausse</b> et selon le résultat on effectue un traitement ou un autre.	<b>SI</b> condition vraie <b>ALORS</b> traitement 1 <b>SINON</b> traitement 2 <b>FIN SI</b>	<b>if</b> condition vraie : traitement 1 <b>else :</b> traitement 1

### Exemple à tester :

<p>Deviner ce que fait ce programme, juste en le lisant, puis vérifier en le exécutant.</p> <p>Des détails peuvent vous être inconnus mais nous y reviendront *</p>	<pre> 1 numero=7 2 nom=input('Nom?') 3 proposition=int(input('Deviner un chiffre entre 1 et 10')) 4 5 if proposition==numero: 6     print('Bravo', nom, '!') 7     print('Le numero était bien ', numero ) 8 elif proposition &gt;numero: 9     print("C'est moins, réessayer une autre fois.") 10 else : 11     print("C'est plus, réessayer une autre fois.") </pre>
---	--

Réponse

### Point Info\*

**Il faut deux points « : »** après **if** et **else**. Ils correspondent au mot « **alors** ».

**Il faut 4 espaces** en dessous de **if**, **else** ou **elif**. Cela délimite le bloc d'instructions concerné. C'est l'**indentation**.

**Il faut un double égal « == »** dans une condition. Le **=** simple est réservé à l'affectation des variables.

« **elif** » est la contraction de « **else** » et « **if** ». On peut imbriquer plusieurs conditions.

### Programme BankNum.B :

BankNum veut proposer à l'utilisateur le choix entre déposer ou retirer de l'argent sur un compte courant.

Reprenez votre programme BankNum.A où l'utilisateur a déjà 1000€ sur son compte.

Ajoutez des instructions pour demander à l'utilisateur : « *Pour un dépôt ( taper d) ou retrait ( taper r)* »

et enregistre la réponse dans une variable **action**.

Selon l'**action** choisie faites afficher : « Bonjour **nom**, quelle somme voulez vous déposer ? »

ou bien « Bonjour **nom**, quelle somme voulez vous retirer ? ».

Enregistrez la réponse dans une variable **somme**

Le programme doit alors afficher « Vous avez **compte** € sur votre compte »

Enregistrez votre programme BankNum.B

Réponse

### Bonus facultatifs :

- Faites afficher « *Découvert non autorisé* » si le solde du compte devient négatif.
- Faites demander un **code** client, pour que l'utilisateur puisse accéder la demande. Sinon le programme répond « code erroné. » et se termine.

### C. Compte épargne

Dans un «Compte épargne » l'argent déposé rapporte des intérêts tous les mois.

#### Point Info

Une boucle **bornée** répète des instructions **un nombre de fois n connu** à l'avance.

Algorithme	Python
<p><b>Pour</b> <i>i</i> <b>variant de 1 à n faire</b>  <i>bloc d'instruction</i></p> <p><b>Fin Pour</b></p>	<p><b>for</b> <i>i</i> <b>in range (1,n+1) :</b>  <i>bloc d'instruction</i></p>

*i* est appelé le "**compteur**" ou «*itérateur* » : *i* augmente automatiquement de 1 jusqu'à n.

#### Exemple à tester :

Voici un court programme.  
 Deviner ce qu'il fait,  
 puis vérifier en l'exécutant.

```
table=int(input('Donner un chiffre entre 1 et 10'))
for i in range(1,11):
    print(i,'x',table,'=',i*table)
```

Réponse :

#### Programme BankNum.C :

a) L'établissement BankNum veut proposer un livret d'épargne un taux de 1 % par an calculé sur la base des **intérêts composés** : les intérêts s'ajoutent chaque année au capital initial, pour calculer les intérêts des années suivantes.

Pour un capital initial de 1000 € quel sera le capital au bout ,

- d'un an d'épargne ?
- de 2 ans d'épargne ?
- de 5 ans d'épargne ?

Vérifier vos affirmations avec le simulateur en ligne suivant : [Lafinancepourtous.com](http://Lafinancepourtous.com)

Épargne périodique :

Durée du placement :

Taux annuel de rendement espéré :


b) L'établissement BankNum veut proposer son propre simulateur de «Compte épargne » , en fonction de la somme déposée et selon la durée choisie

Pour cela, reprendre votre programme BankNum.B lui ajouter des instructions pour qu'il demande à l'utilisateur :

« Pour un dépôt ( taper d), un retrait ( taper r), une simulation d'épargne ( taper e) »

et enregistre la réponse dans une variable **action**.

Ajouter au code du programme BankNum.B, la condition si l'**action** choisie est 'e', et demander alors la **somme** à épargner, ainsi que la **durée** en mois.

Le programme doit afficher l'**épargne** totale à la fin de la durée choisie, sachant que la banque un taux de 1 % par an.

Tester votre programme avec les résultats du a) . Enregistrer votre programme BankNum.C

Réponse :

**D : Quitter ( supplément pour les rapides)**

Une boucle **non bornée** répète des instructions tant qu'une condition est satisfaite.

Algorithme	Python
<b>Tant que</b> <i>condition vraie</i> <b>faire</b> <i>bloc d'instruction</i>	<b>while</b> <i>condition vraie</i> : <i>bloc d'instruction</i>
<b>Fin tant que</b>	

NB : Les deux points : et le décalage en dessous de while sont obligatoires

**Exemple à tester :**

Voici un programme assez compréhensible,  
Deviner ce qu'il fait,  
puis vérifier en l'exécutant.

```
proposition=0
essai=0

while proposition!=7:
    proposition=int(input('Combien font 2 + 5 ?'))
    print('Essaye encore !')
    essai=essai+1

print('Bravo ! Trouvé en ', essai,'essai(s)')
```

Réponse :

**Programme BankNum.D :**

Reprendre un de vos programme BankNum

lui ajouter des instructions pour qu'il demande en plus à l'utilisateur s'il veut « quitter » :

« Pour un dépôt ( taper d), ....., **pour Quitter( taper q)** » et enregistre la réponse dans une variable **action**.

Englober une bonne partie du code du programme, par une boucle « tant que l'utilisateur n'a pas tapé q »

et dans ce cas, proposer les divers services, mais aussi de nouveau Quitter.

Réponse :

c) Établissement d'une formule directe, introduction aux suites.

*Activité 2 p158 du manuel « Mathématiques premières STI2D/STL. Collection Sigma ».*