

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

Informatique

17

Rédaction

Cours

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

<i>Ecrire un code commenté et lisible</i>	3
1.I. Conseils	3
1.II. Exemples	3
1.II.1 Mauvaise rédaction	3
1.II.2 Bonne rédaction	4
1.III. Pourquoi tout cela	5
1.III.1 Compréhension	5
1.III.2 Débogage.....	7

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

Ecrire un code commenté et lisible

Maintenant que vous savez programmer, il est nécessaire que vous appreniez à rédiger proprement un code.

1.I. Conseils

- Utiliser des noms de variables clairs et parlant
- Aérez votre code avec lignes vides et espaces
- Organiser les lignes logiquement les unes après les autres
- Commenter :
 - o ## Pour séparer des grandes parties
 - o # Pour préciser localement à quoi sert votre code
- A l'écrit, marquer l'indentation par des traits verticaux

1.II. Exemples

Voici deux exemples de fonctions réalisant le même calcul, vous comprendrez vite la différence.

1.II.1 Mauvaise rédaction

```

from math import cos
from math import pi
N = 100
a = []
c = []
for i in range(N):
    b = (2*pi/N)*i
    a.append(abs(cos(b)))
    c.append(b)
import matplotlib.pyplot as plt
plt.close('all')
plt.plot(c,a)
plt.show()
def f(a):
    b = 0
    for i in range(len(a)):
        b += a[i]
    return b/len(a)
d = f(a)
e = [d for i in range(N)]
plt.plot(c,e)
plt.show()

```

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

1.II.2 Bonne rédaction

```
## Code proposé

'''
Discrétisation de la fonction abs(cos(x)) sur N points et calcul de sa
moyenne
Auteur: Denis DEFAUCHY
'''

## Import des librairies

from math import cos
from math import pi
import matplotlib.pyplot as plt

## Définition des fonctions

def f_Moyenne(Liste):
    Nb_Termes = len(Liste)
    Somme = 0
    for i in range(len(Liste)): # Calcul de la somme des Nb_Termes de la
liste Liste
        Terme = Liste[i]
        Somme += Terme
    Moyenne = Somme/Nb_Termes # Calcul de la moyenne de la liste Liste
    return Moyenne

## Programme

N = 100 # Nombre de points de la discrétisation
Liste_X = []
Liste_abs_cos = []

for i in range(N): # Création de la liste des valeurs de la fonction
abs(cos(x))
    X = (2*pi/N)*i # Création de la liste des abscisses sur une période
    Liste_X.append(X)
    abs_cos = abs(cos(X))
    Liste_abs_cos.append(abs_cos)

Moyenne = f_Moyenne(Liste_abs_cos) # Calcul de la moyenne de la fonction
abs(cos(x))
Liste_Moyenne = [Moyenne for i in range(N)] # Création de la liste pour
tracer la moyenne

# tracé des courbes

plt.close('all') # Fermeture des courbes déjà ouvertes
plt.plot(Liste_X, Liste_abs_cos) # Ajoute de la courbe de abs(cos(x))
plt.plot(Liste_X, Liste_Moyenne) # Ajout de la courbe de la moyenne
plt.show() # Affichage du graphique
```

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

1.III. Pourquoi tout cela

- En prépa
 - o Vous allez être évalués à l'écrit... **Vous serez donc lus par des HUMAINS** et non par des machines
 - o Vous ne serez pas là pour expliquer ce que fait votre code
 - o Les correcteurs partiront du principe que les enseignants vous apprennent à bien rédiger – S'ils se cassent la tête à comprendre un code, ils passeront sans mettre de points même si c'est juste, partant de ce principe
- Ensuite
 - o Dans le milieu professionnel, votre code doit être lisible par les autres
 - o Et surtout, par vous-même plusieurs mois ou années après...

Je vais souvent insister sur la nécessité d'écrire en plusieurs lignes ce que vous faites en une seule. Diverses raisons y sont liées, en particulier :

- Cela rend le code **compréhensible** à la première lecture
- Cela permet de trouver bien plus facilement des erreurs : **débogage**

1.III.1 Compréhension

Aux concours d'abord par les correcteurs de vos copies, par vous et vos futurs collègues ensuite, vos codes seront lus et doivent être compris très rapidement.

Livrons-nous d'abord à un petit exercice. Soit le code suivant :

```
Liste = ["conseille ", "codes", "vous ", "variables "], ["créer", "intermédiaires ", "des ", "vos ", "de "], ["je ", "dans "]

Phrase =
Liste[2][0]+Liste[0][2]+Liste[0][0]+Liste[1][4]+Liste[1][0]+Liste[1][2]+L
iste[0][2]+Liste[1][1]+Liste[2][1]+Liste[1][3]+Liste[0][1]

print(Phrase)
```

Y-a-t-il une erreur dans ma phrase ? Si oui, où est-elle ? Jouez le jeu !!!!

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

Et dans ce code :

```
Liste = ["conseille ", "codes", "vous ", "variables "], ["créer",
"intermédiaires ", "des ", "vos ", "de "], ["je ", "dans "]

conseille = Liste[0][0]
codes =     Liste[0][1]
vous =     Liste[0][2]
variables = Liste[0][2]

creer =     Liste[1][0]
intermediaires = Liste[1][1]
des =      Liste[1][2]
vos =      Liste[1][3]
de =       Liste[1][4]

je = Liste[2][0]
dans = Liste[2][1]

Phrase =
je+vous+conseille+de+creer+des+variables+intermediaires+dans+vos+codes

print(Phrase)
```

Y-a-t-il une erreur ? Ou est-elle ?

Vous avez j'espère tout compris ! Alors, lorsque vous écrirez des codes illisibles, je ne manquerai pas de ne pas vous mettre tous les points, même s'ils sont justes !

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

1.III.2 Débogage

Concernant les erreurs, voici un exemple qui permet de déboguer immédiatement le code s'il est écrit en plusieurs lignes au lieu d'une :

```
Alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

def f_Cesar_1(Phrase,Cle):
    Taille_Phrase = len(Phrase)
    Resultat = []
    for i in range(Taille_Phrase):
        Lettre = Phrase[i]
        Ind = Alphabet.index(Lettre)
        Ind_New = Ind + Cle
        Ind_New = Ind_New % 26
        Resultat.append(Alphabet[Ind_New])
    return f_Regrouper(Resultat)

def f_Cesar_2(Phrase,Cle):
    Resultat = []
    for i in range(len(Phrase)):
        Resultat.append(Alphabet[(Alphabet.index(Phrase[i]) + Cle)%26])
    return f_Regrouper(Resultat)
```

Ce code transforme un mot en décalant ses lettres d'une valeur contenue dans Cle.

```
>>> f_Cesar_1('COUCOU',2)
'EQWEQW'

>>> f_Cesar_2('COUCOU',2)
'EQWEQW'
```

Les deux fonctions proposées réalisent le même travail. Autant dire, dès le départ, que la fonction 1 est bien plus claire que la fonction 2.

Mais au-delà de ce problème, le gros avantage de rédiger le code de la fonction 1 est qu'en cas d'erreur, on identifie immédiatement l'origine du problème.

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

Ajoutons la même erreur aux deux codes :

```
def f_Cesar_1(Phrase,Cle):
    Taille_Phrase = len(Phrase)
    Resultat = []
    for i in range(Taille_Phrase):
        Lettre = Phrase(i)
        Ind = Alphabet.index(Lettre)
        Ind_New = Ind + Cle
        Ind_New = Ind_New % 26
        New_Lettre = Alphabet[Ind_New]
        Resultat.append(New_Lettre)
    return f_Regrouper(Resultat)

def f_Cesar_2(Phrase,Cle):
    Resultat = []
    for i in range(len(Phrase)):
        Resultat.append(Alphabet[(Alphabet.index(Phrase(i)) + Cle)%26])
    return f_Regrouper(Resultat)
```

L'avez-vous trouvée ? On exécute chacune de ces fonctions, et voici ce que l'on obtient :

```
>>> f_Cesar_1('COUCOU',2)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "C:\Users\Denis DEFAUCHY\Dropbox\Privé\Professionnel\Cours\CPGE\Chapitres\6 - Info IPT\IPT 1\Cours\Exemple_Moche.py", line 28, in f_Cesar_1
    Lettre = Phrase(i)
TypeError: 'str' object is not callable

>>> f_Cesar_2('COUCOU',2)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "C:\Users\Denis DEFAUCHY\Dropbox\Privé\Professionnel\Cours\CPGE\Chapitres\6 - Info IPT\IPT 1\Cours\Exemple_Moche.py", line 38, in f_Cesar_2
    Resultat.append(Alphabet[(Alphabet.index(Phrase(i)) + Cle)%26])
TypeError: 'str' object is not callable
```

Dans la seconde exécution, l'erreur est à la ligne 38. Celle-là :

```
38 | Resultat.append(Alphabet[(Alphabet.index(Phrase(i)) + Cle)%26])
```

Ce n'est pas évident de savoir où elle est !

Dans le premier code, elle est là :

```
28 | Lettre = Phrase(i)
```

C'est évident, il faut corriger les parenthèses en crochets.

Dernière mise à jour	Informatique	Denis DEFAUCHY
30/03/2021	17 – Rédaction	Cours

Un dernier exemple encore plus difficile à corriger.

On utilise le code sans erreurs, et on l'exécute. On obtient alors :

```
>>> f_Cesar_1('Coucou',2)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
    File "<tmp 1>", line 8, in f_Cesar_1
      Ind = Alphabet.index(Lettre)
ValueError: substring not found

>>> f_Cesar_2('Coucou',2)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
    File "<tmp 1>", line 17, in f_Cesar_2
      Resultat.append(Alphabet[(Alphabet.index(Phrase[i]) + Cle)%26])
ValueError: substring not found
```

Comment trouver d'où vient l'erreur ? Dans le second cas, on a la ligne, mais où est-elle ?

Dans le second code, on y voit un peu plus clair. L'erreur est dans la création de l'indice Ind. Pour déboguer, on ajoute alors au code un « print » de la lettre en argument de la ligne qui pose problème :

```
print(Lettre)
Ind = Alphabet.index(Lettre)
```

On réexécute la fonction 1, ce qui donne :

```
>>> f_Cesar_1('Coucou',2)
C
o
Traceback (most recent call last):
  File "<console>", line 1, in <module>
    File "<tmp 1>", line 9, in f_Cesar_1
      Ind = Alphabet.index(Lettre)
ValueError: substring not found
```

On peut voir que le code a fonctionné pour « C », mais pas pour « o ». Mon code est donc censé fonctionner puisqu'il est passé une fois. Je regarde de plus près, et je trouve : il y a un « o » qui pose problème. Eh oui, il est en minuscule, alors que dans l'alphabet, il est en majuscule... Il n'est donc pas possible de trouver cette lettre...

Lorsque je passe derrière vos ordinateurs, si vous n'avez pas codé proprement, je peux passer des heures à chercher l'erreur. Souvent, j'utilise des « print » pour identifier son origine, si les variables existent. Je refuse de prendre le temps de corriger des codes avec des lignes trop remplies et pas claires !

VOUS DEVREZ DONC REDIGER VOS CODES CLAIREMENT EN PLUSIEURS LIGNES !