

RECONNUE PAR L'ÉTAT

PYTHON

SÉANCE NUM 2



Enseignant : GHANEM

Plan de la Séance 2

Sommaire :

Partie 1 : Tutoriel : mon premier code en python
(comparaison interprété VS compilé)

Partie 2 : Types de variables en Python

Partie 3 : Structures conditionnelles et Structures itératives

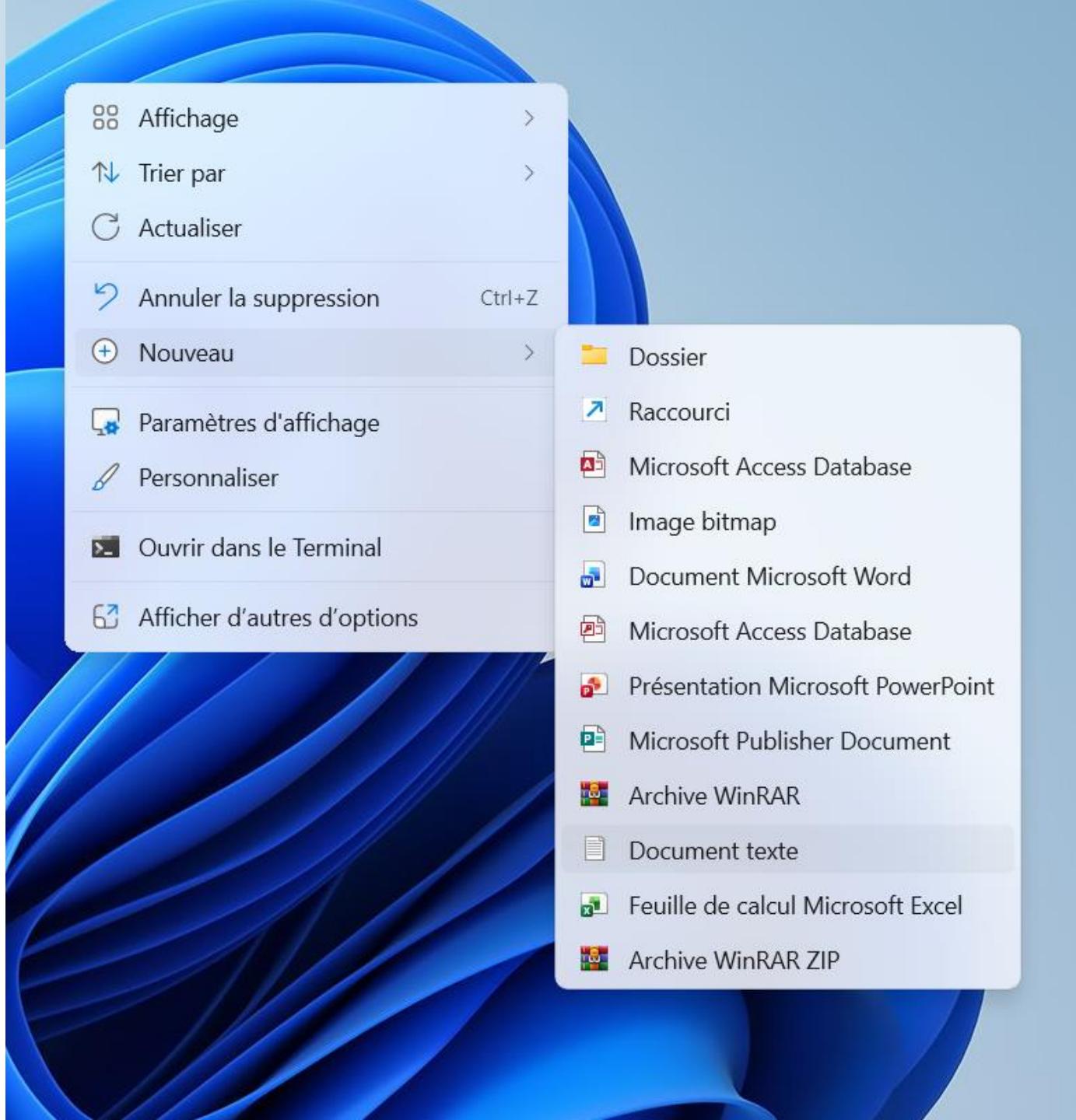
Partie 4 : Fonctions

Partie 5 : l'équivalent des tableaux en Python ?

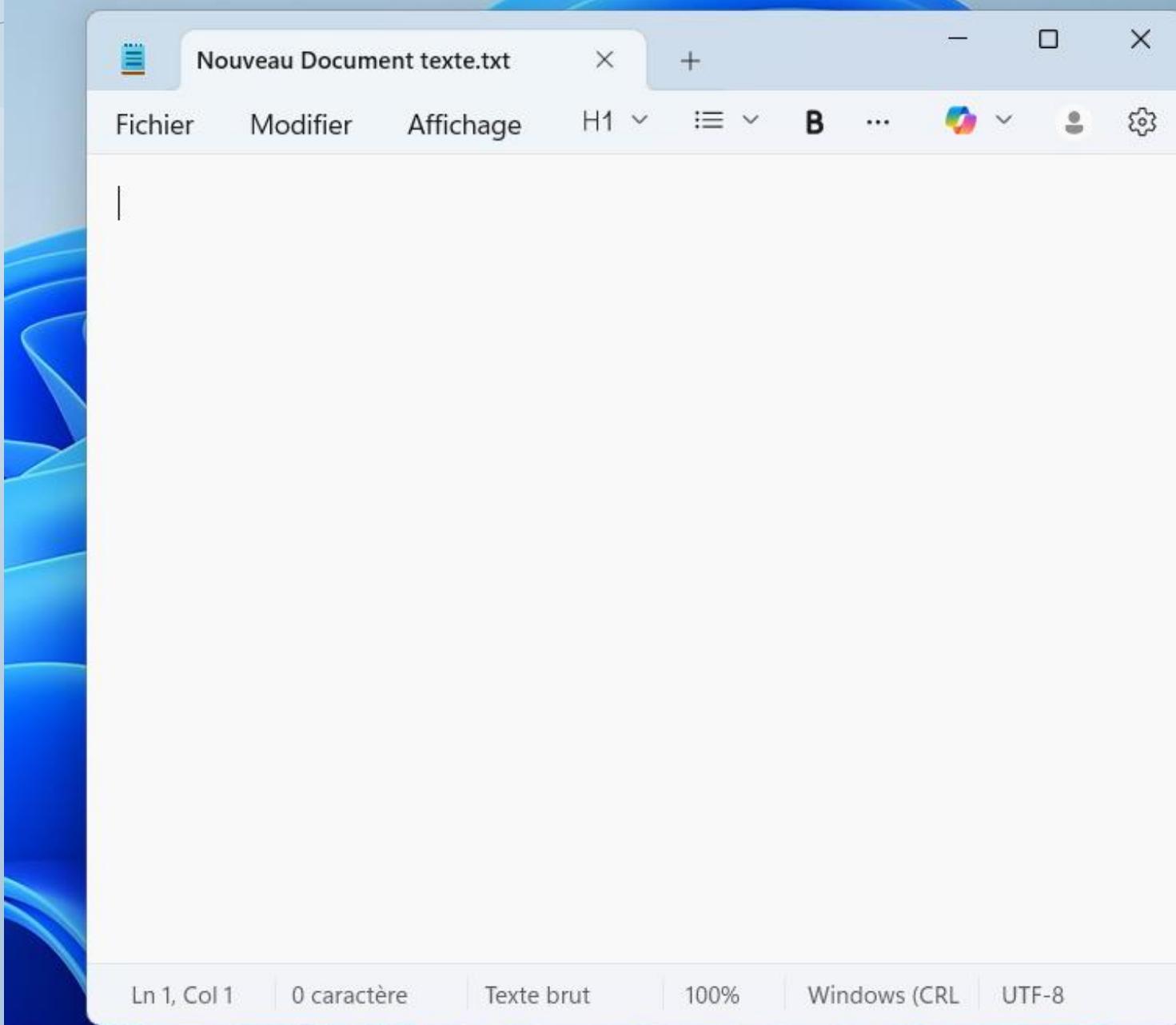
Partie 6 : l'équivalent des structures en Python ?

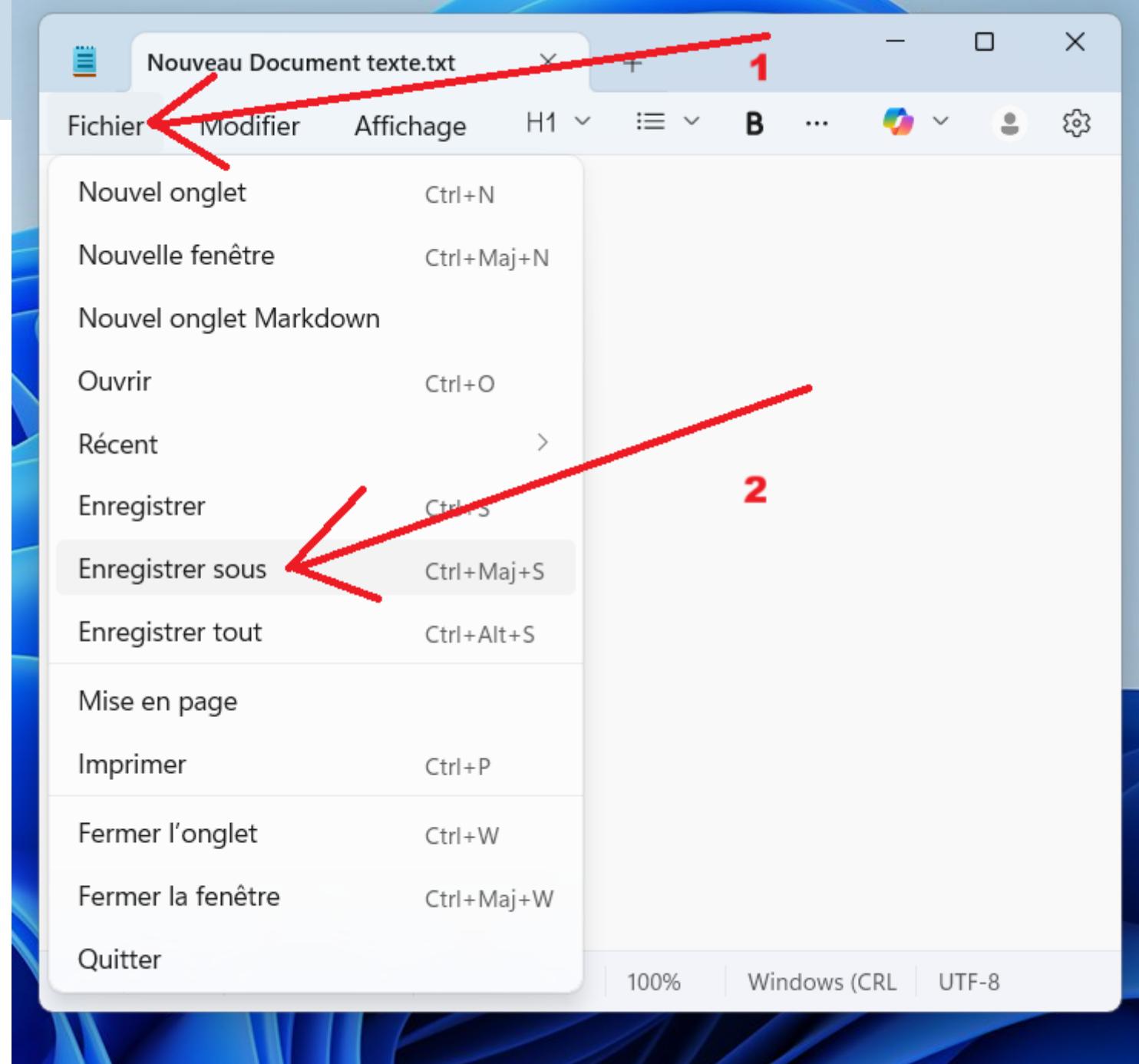
Partie 7 : Définition simple d'une classe et son utilisation

Partie 1 : Tutoriel : mon premier code en python (comparaison interprété VS compilé)









 Enregistrer sous

X

← → ▼ ▲ Bureau >▼ 

Rechercher dans : Bureau



Organiser ▾

Nouveau dossier

 Galerie Bureau Téléchargem Documents Images

Nom du fichier : Nouveau Document texte



Type : Fichiers texte (*.txt)



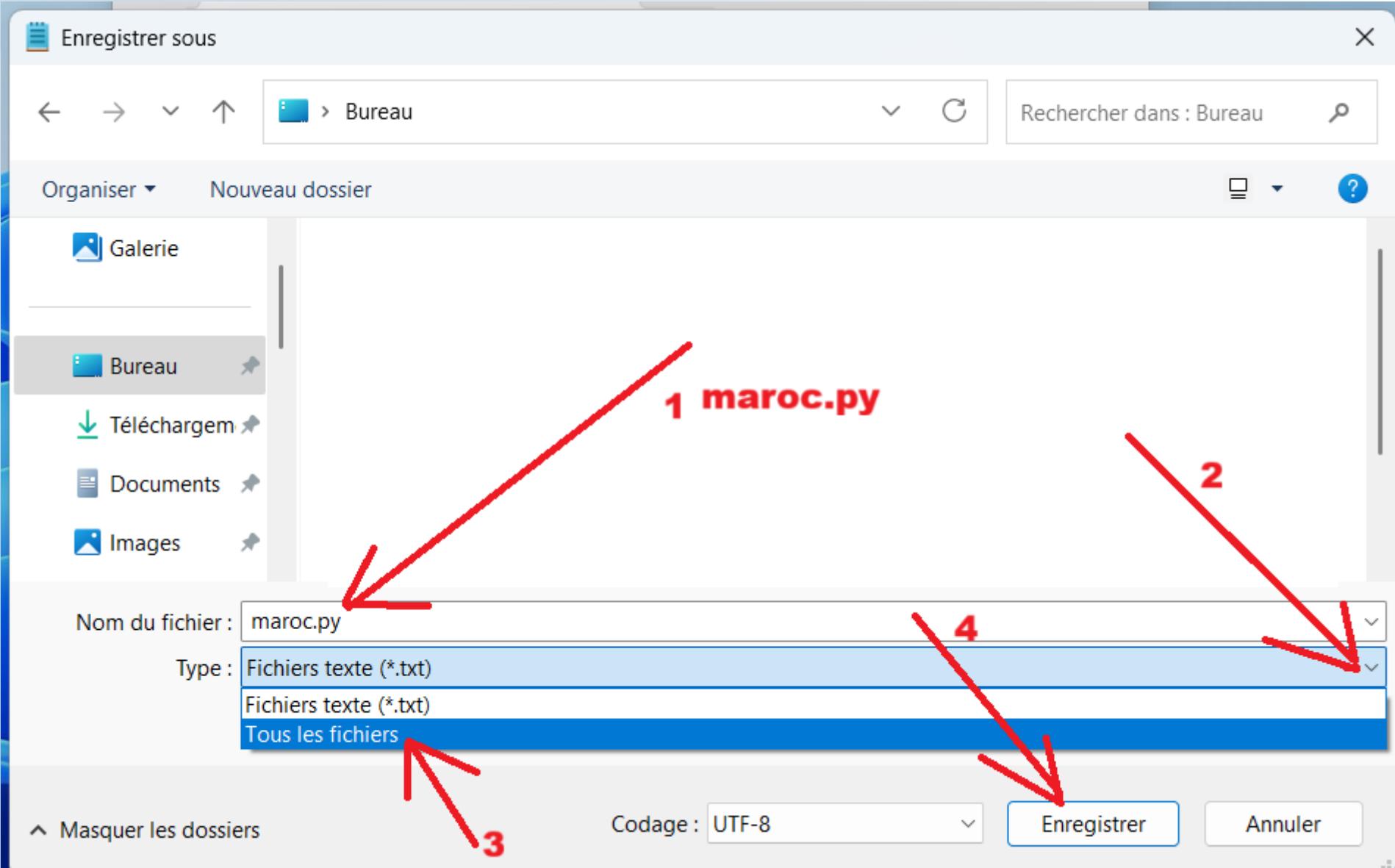
▲ Masquer les dossiers

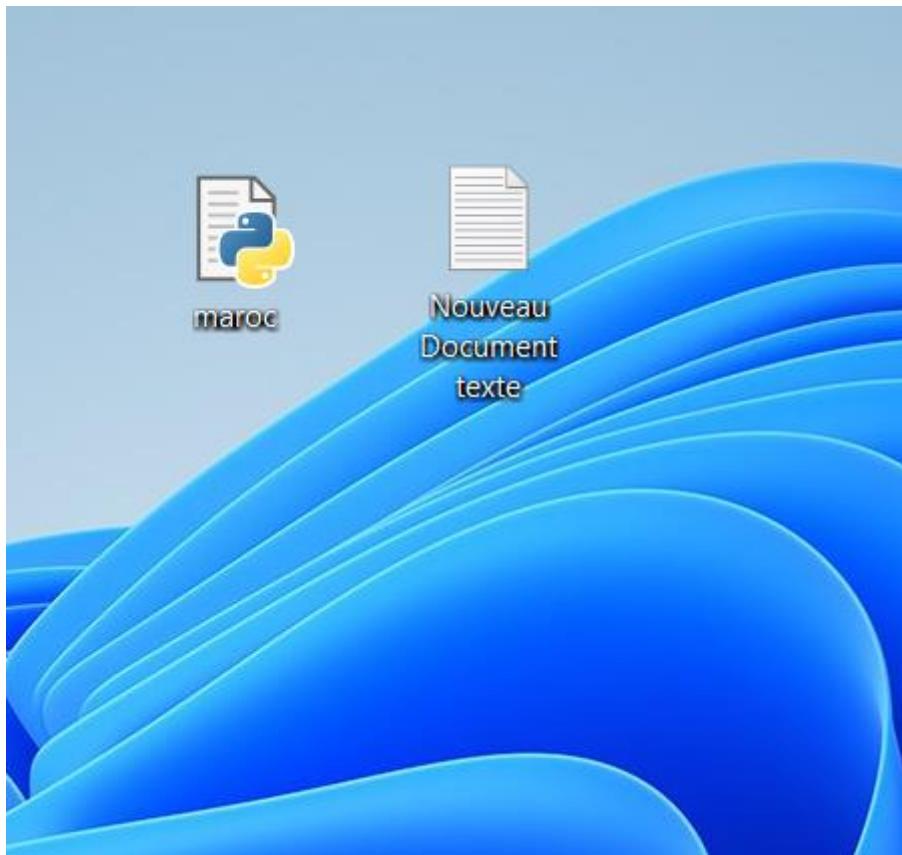
Codage : UTF-8

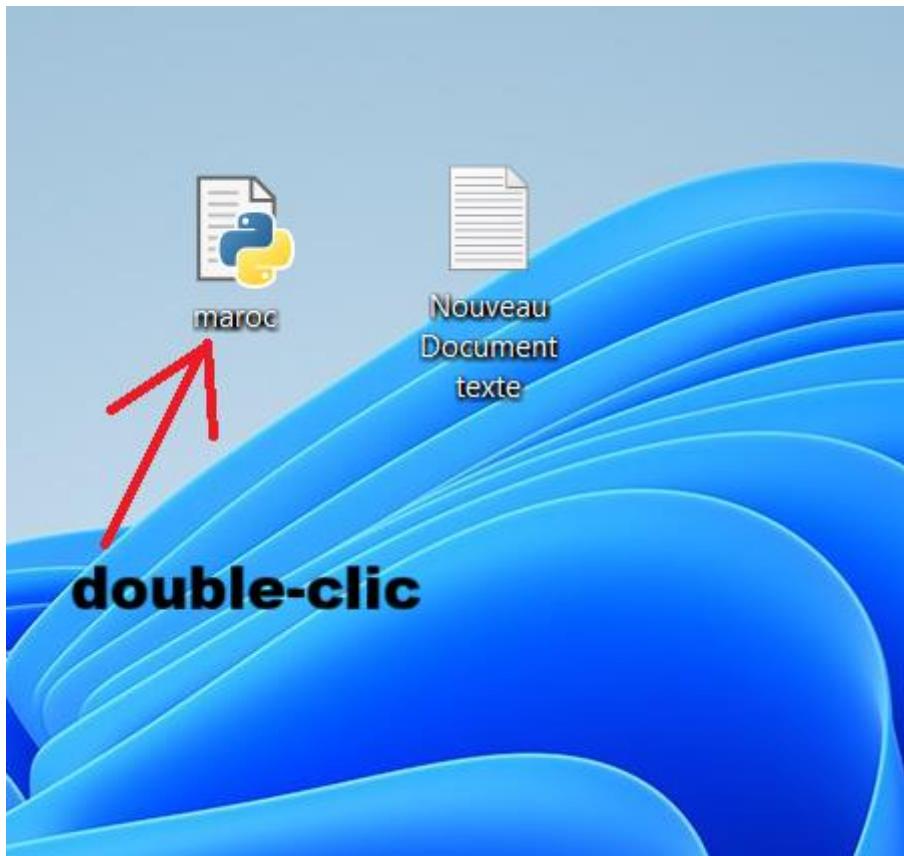
Enregistrer

Annuler



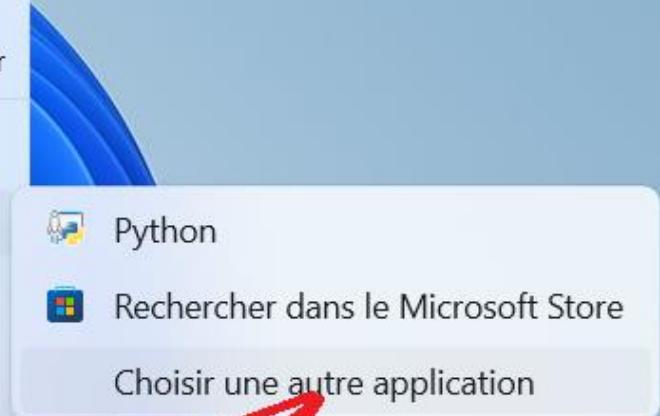
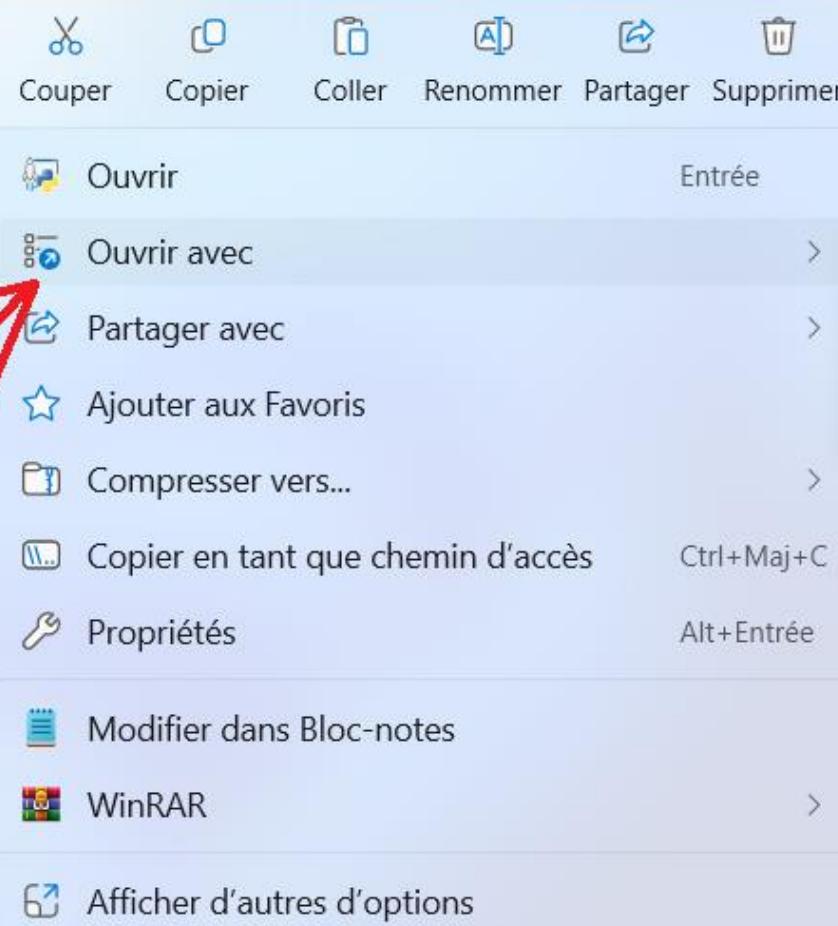






Malheureusement, l'écran d'exécution apparaît et disparaît rapidement.

clic droit



3



Sélectionner une application pour ouvrir ce fichier .py

Application par défaut



Python

double-clic

Autres options



Adobe Acrobat



Bloc-notes



Firefox



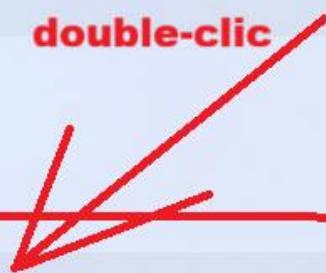
Lecteur multimédia Windows (ancienne génération)



MendeleyDesktop

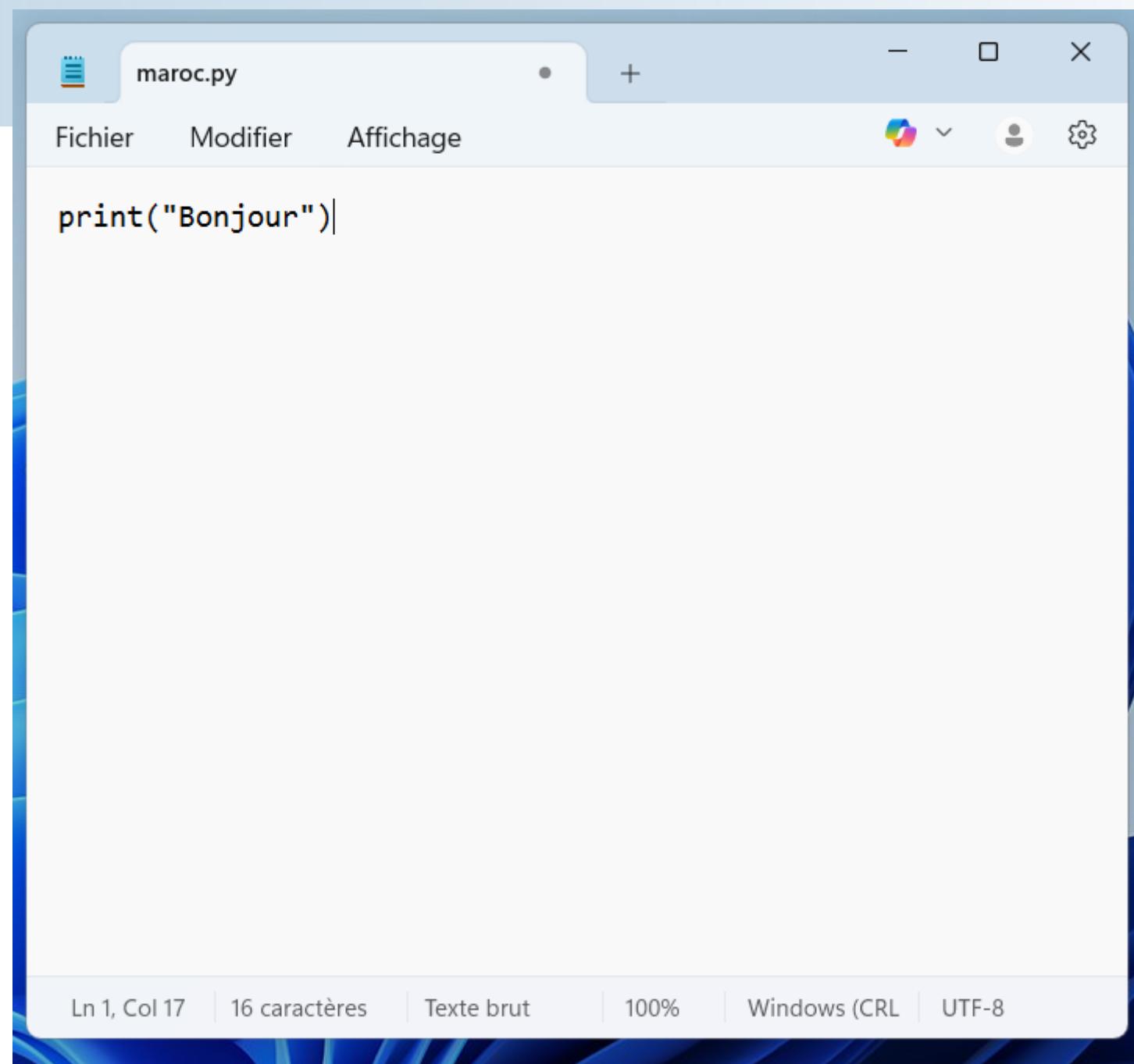


PDF Architect 9



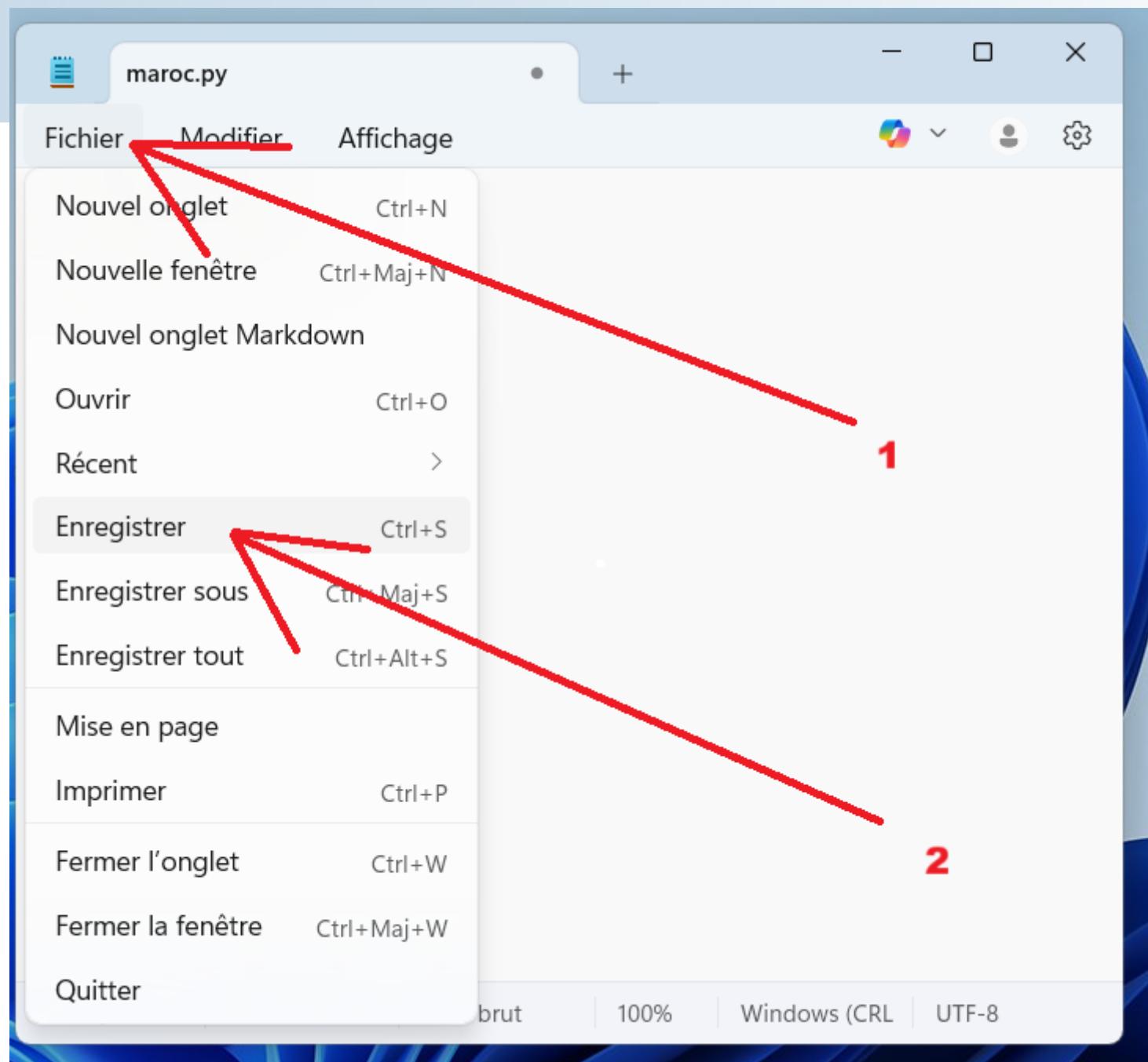
Toujours

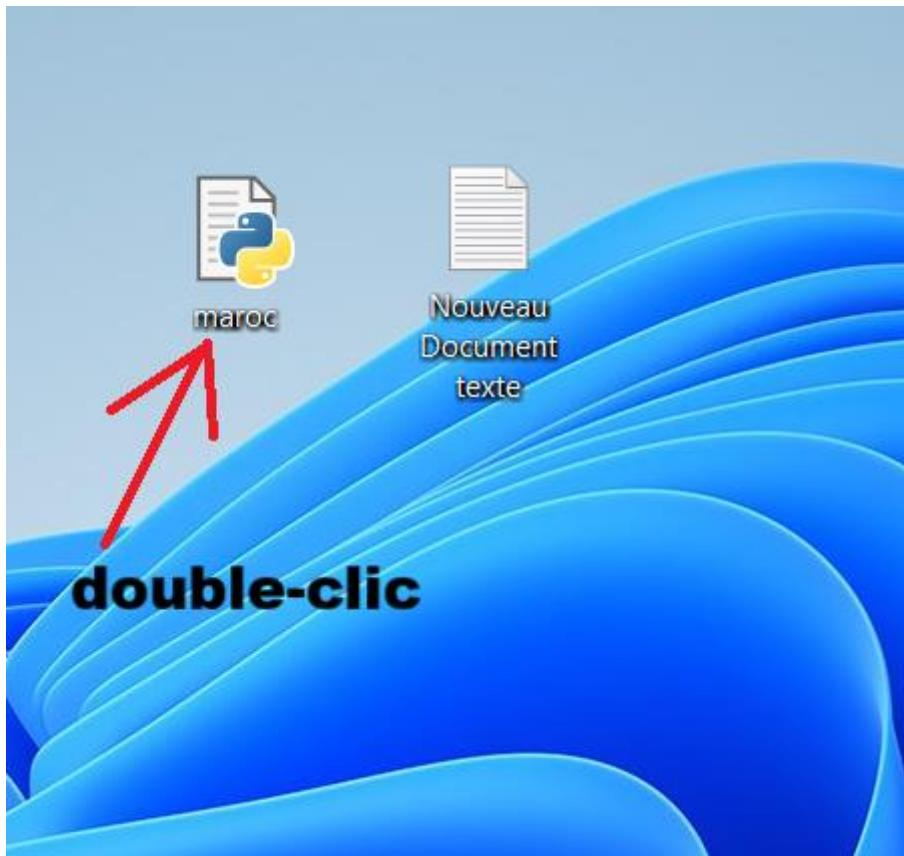
Une seule fois



The image shows a screenshot of a code editor window. The title bar reads "maroc.py". The menu bar includes "Fichier", "Modifier", and "Affichage". The toolbar features icons for file operations, a color palette, user profile, and settings. The main text area contains the Python code: `print("Bonjour")`. The status bar at the bottom displays "Ln 1, Col 17", "16 caractères", "Texte brut", "100%", "Windows (CRL)", and "UTF-8".

```
print("Bonjour")
```



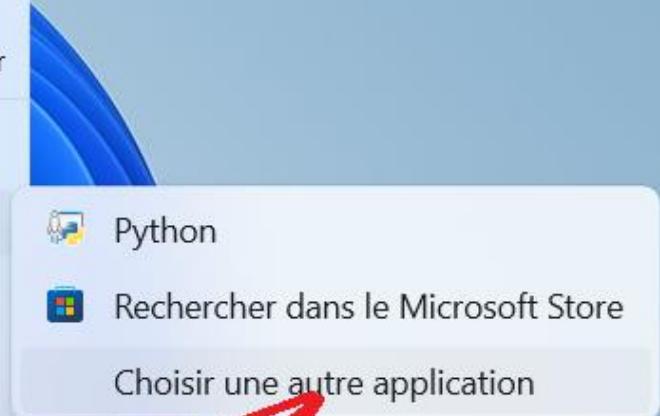
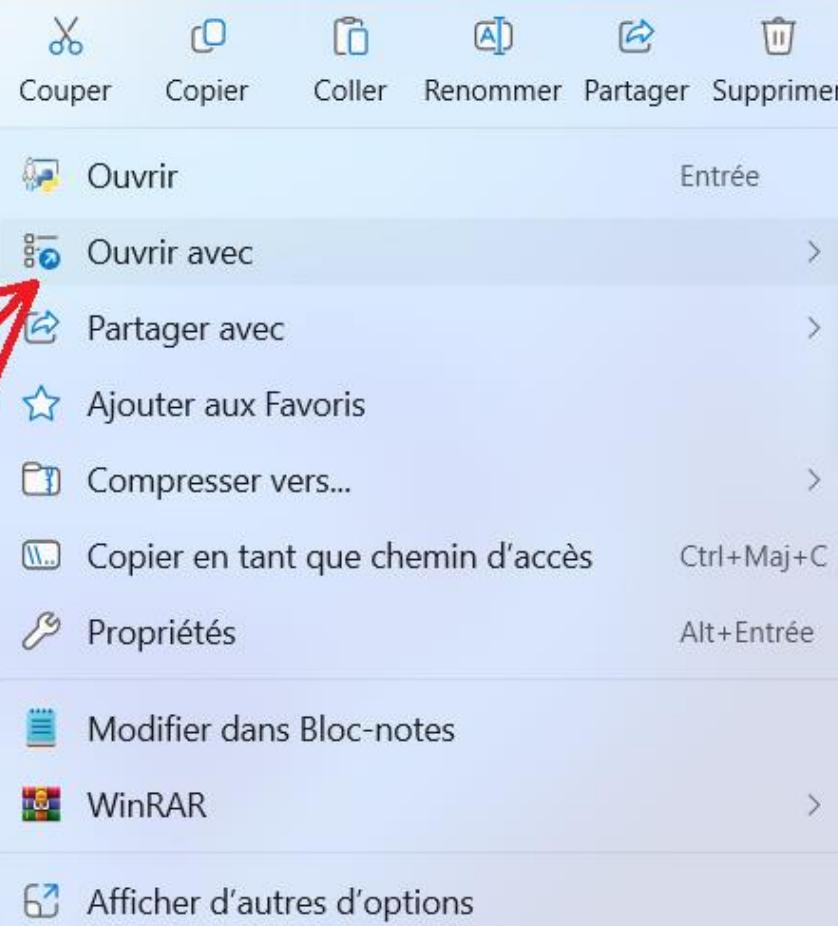


Encore une fois , l'écran d'exécution apparaît et disparaît rapidement !!!!

comment bloquer l'écran ?

On va déclarer un entier k , et on appelle la fonction qui attend la saisie d'un nombre sur le clavier :

clic droit



3



Sélectionner une application pour ouvrir ce fichier .py

Application par défaut



Python

double-clic

Autres options



Adobe Acrobat



Bloc-notes



Firefox



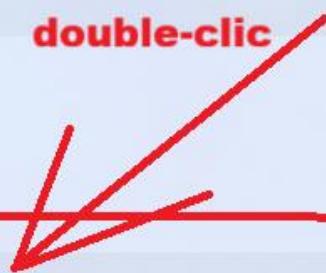
Lecteur multimédia Windows (ancienne génération)



MendeleyDesktop

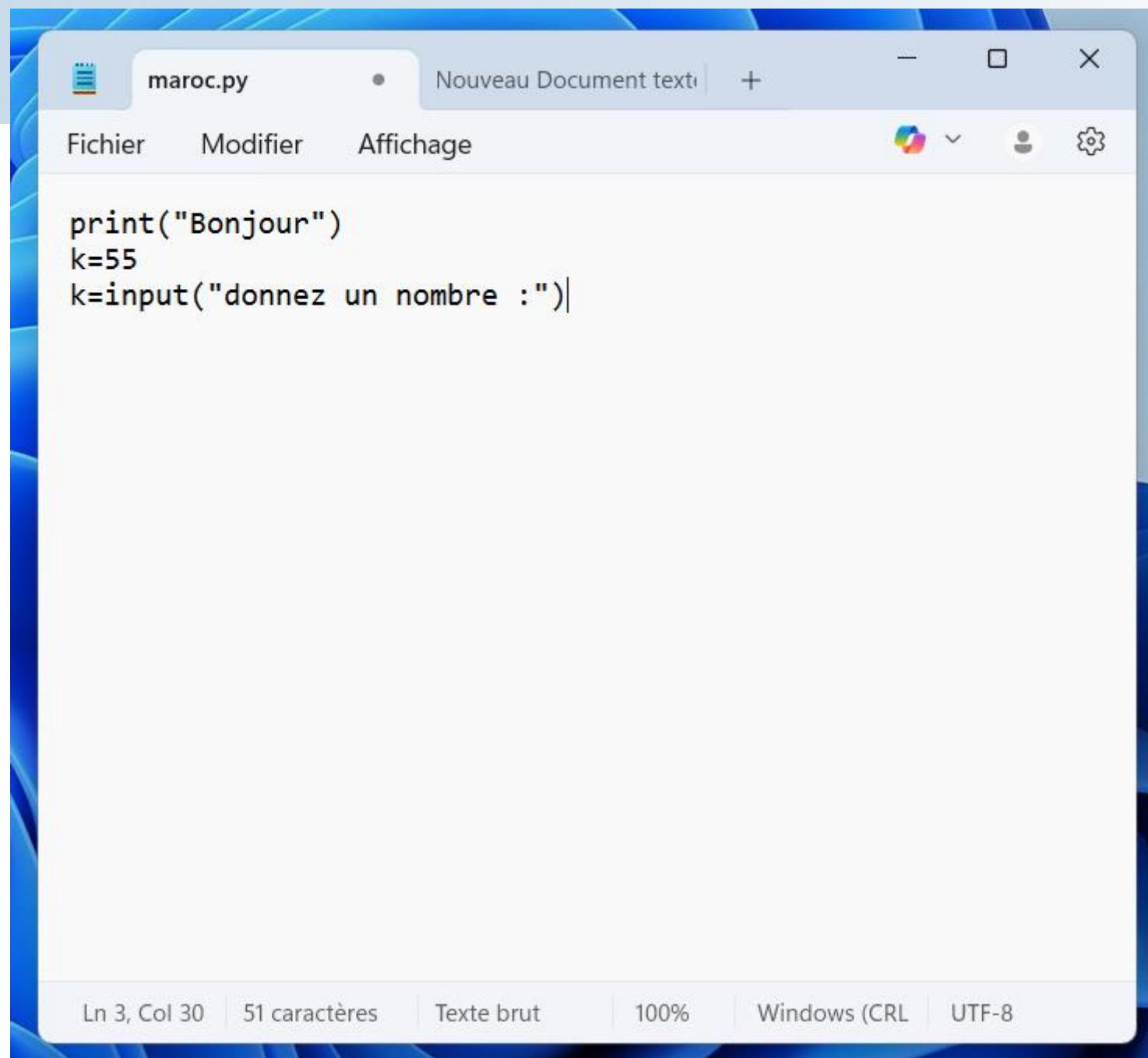


PDF Architect 9



Toujours

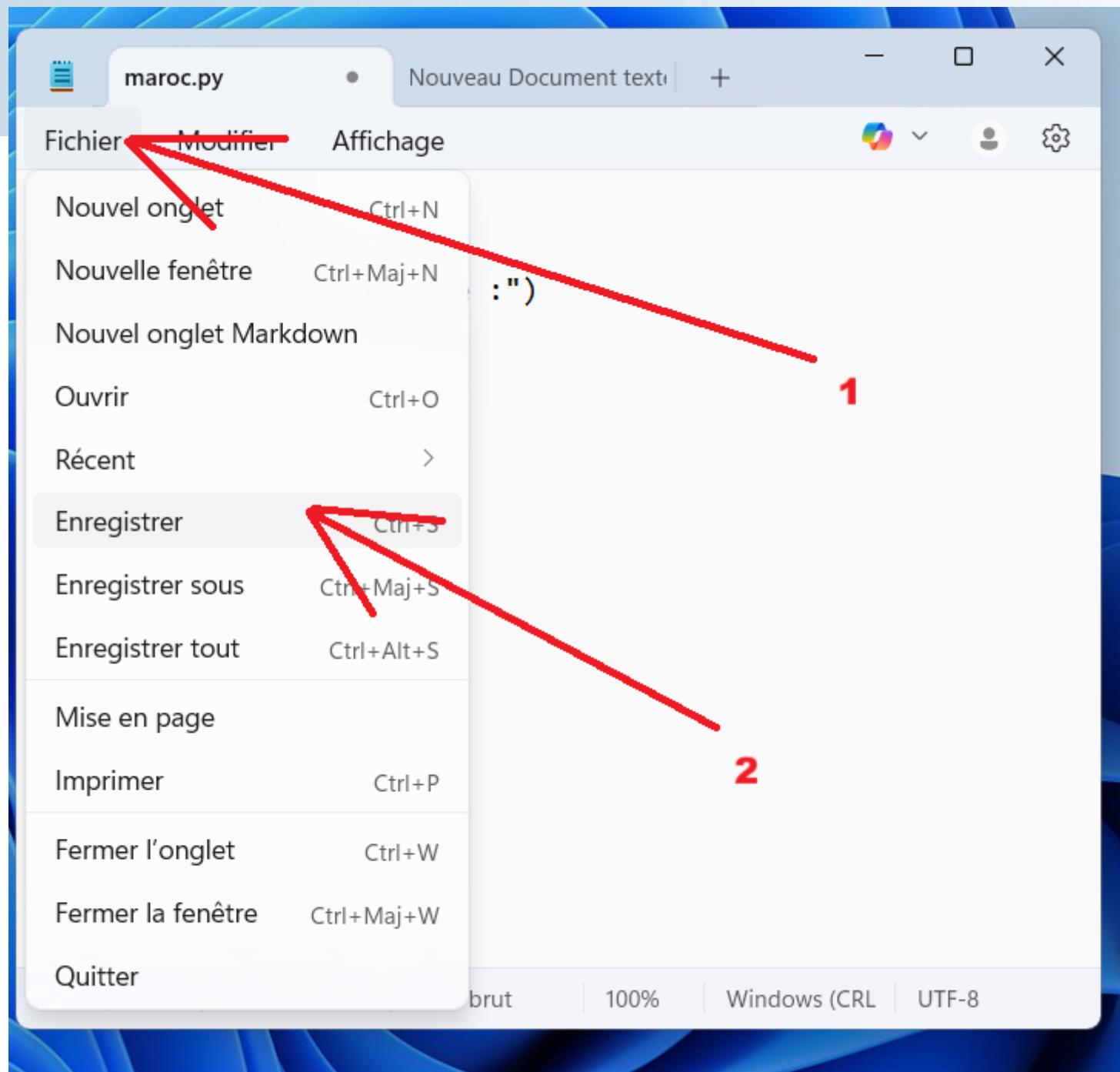
Une seule fois

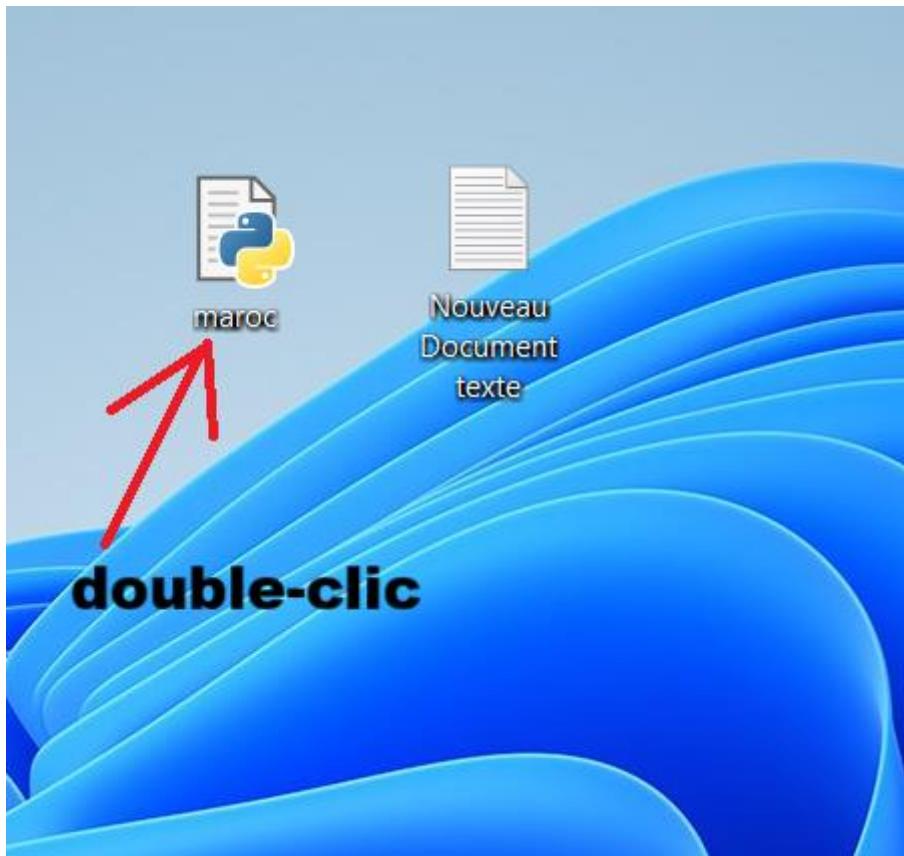


The screenshot shows a text editor window titled "maroc.py". The menu bar includes "Fichier", "Modifier", and "Affichage". The toolbar features icons for file operations, a color palette, user profile, and settings. The main text area contains the following Python code:

```
print("Bonjour")
k=55
k=input("donnez un nombre :")|
```

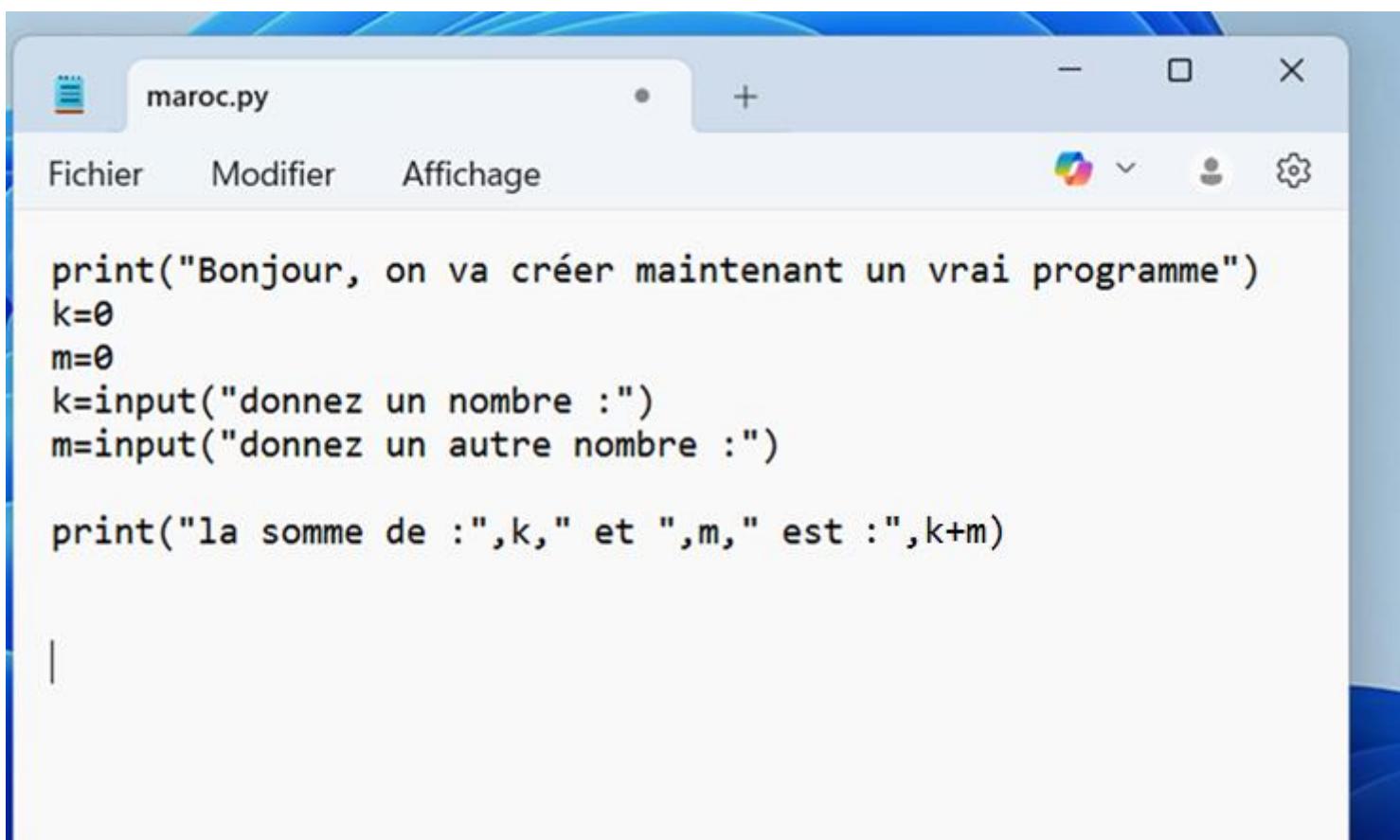
The status bar at the bottom displays: "Ln 3, Col 30 | 51 caractères | Texte brut | 100% | Windows (CRL | UTF-8".





C:\Users\DELL\AppData\Local X

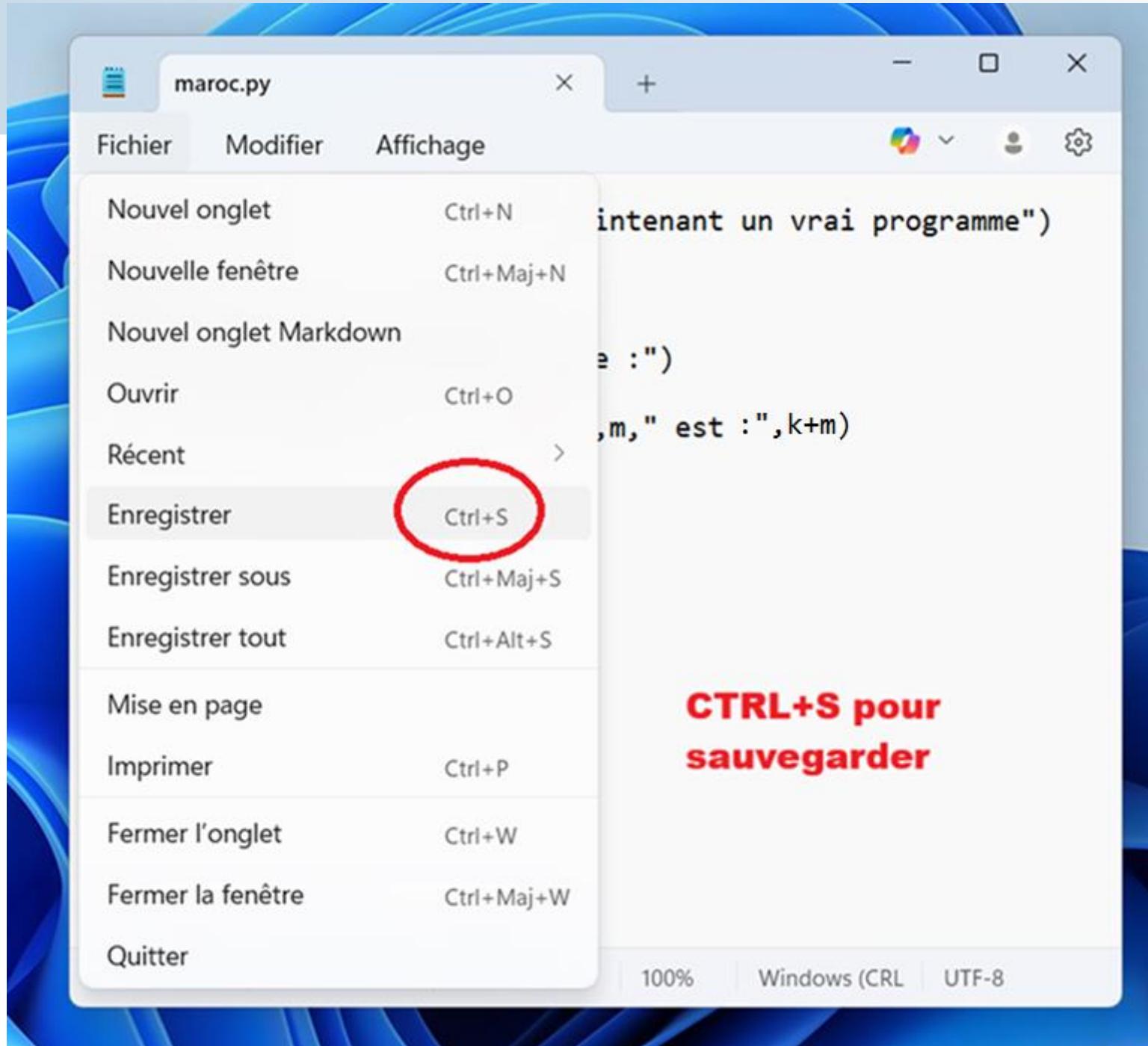
Bonjour
donnez un nombre :

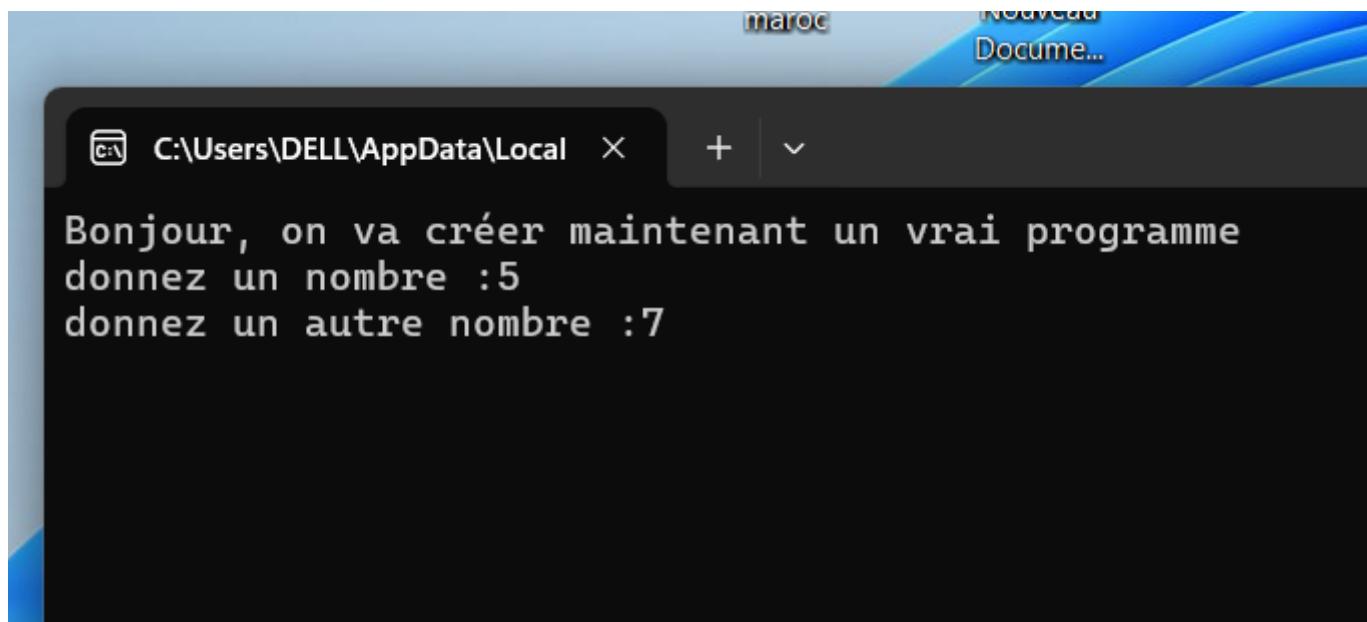


The screenshot shows a Python code editor window titled "maroc.py". The menu bar includes "Fichier", "Modifier", and "Affichage". The toolbar features icons for file operations and settings. The code itself is as follows:

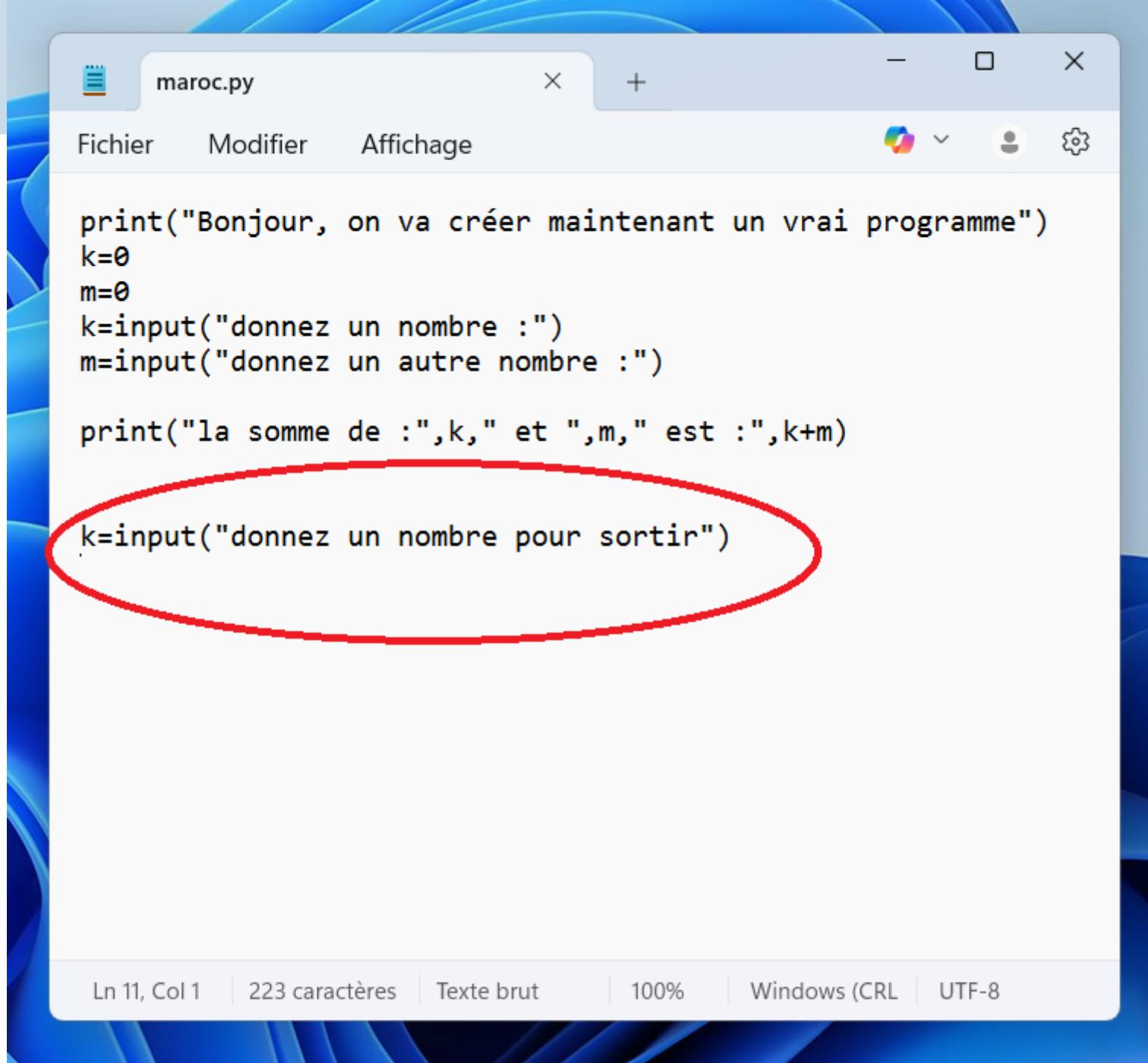
```
print("Bonjour, on va créer maintenant un vrai programme")
k=0
m=0
k=input("donnez un nombre :")
m=input("donnez un autre nombre :")

print("la somme de :",k," et ",m," est :",k+m)
```





on a oublié de bloquer l'écran pour voir le résultat !



The screenshot shows a Python code editor window titled "maroc.py". The menu bar includes "Fichier", "Modifier", and "Affichage". The toolbar features a color palette, user profile, and settings icon. The code itself is as follows:

```
print("Bonjour, on va créer maintenant un vrai programme")
k=0
m=0
k=input("donnez un nombre :")
m=input("donnez un autre nombre :")

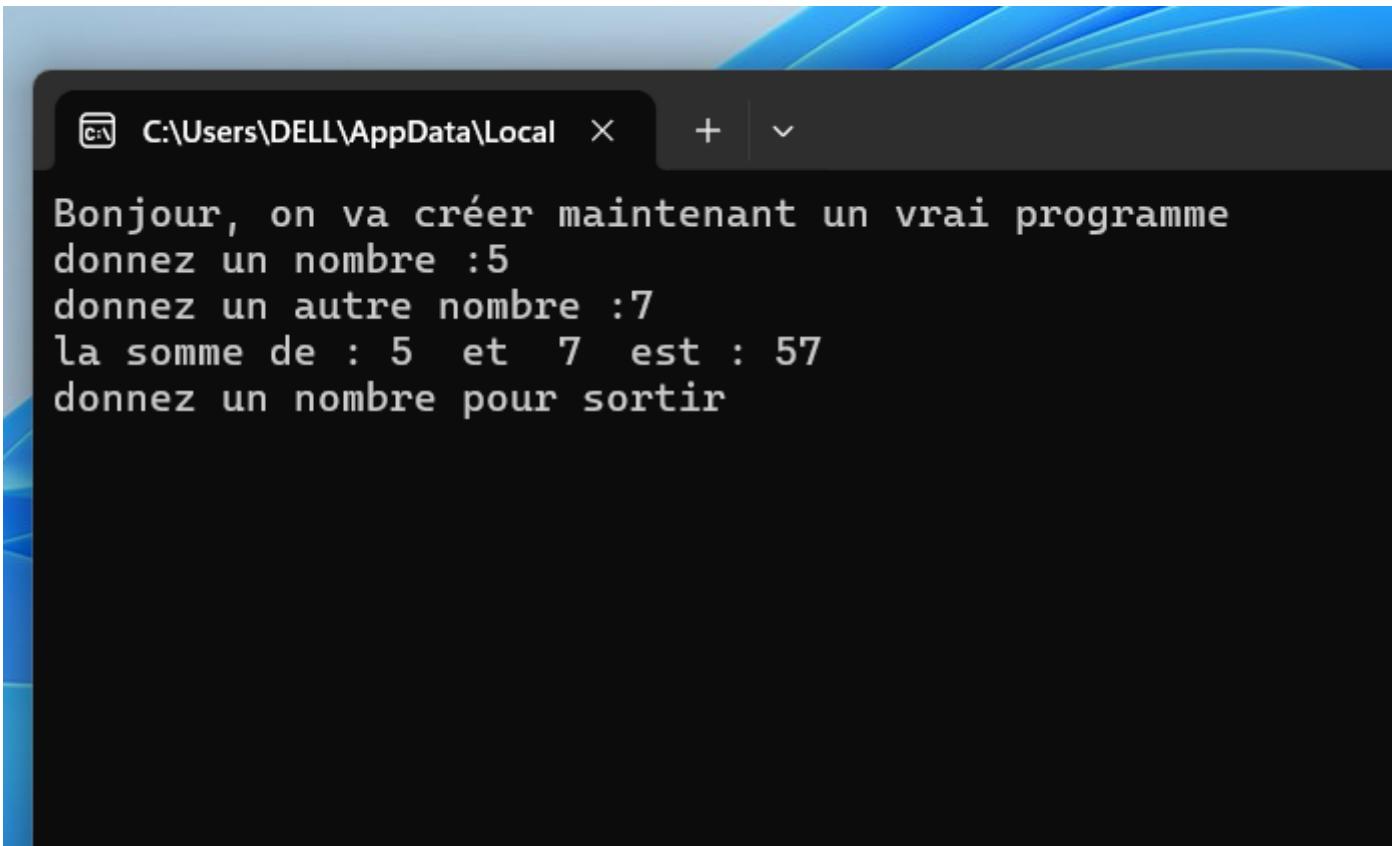
print("la somme de :",k," et ",m," est :",k+m)

k=input("donnez un nombre pour sortir")
```

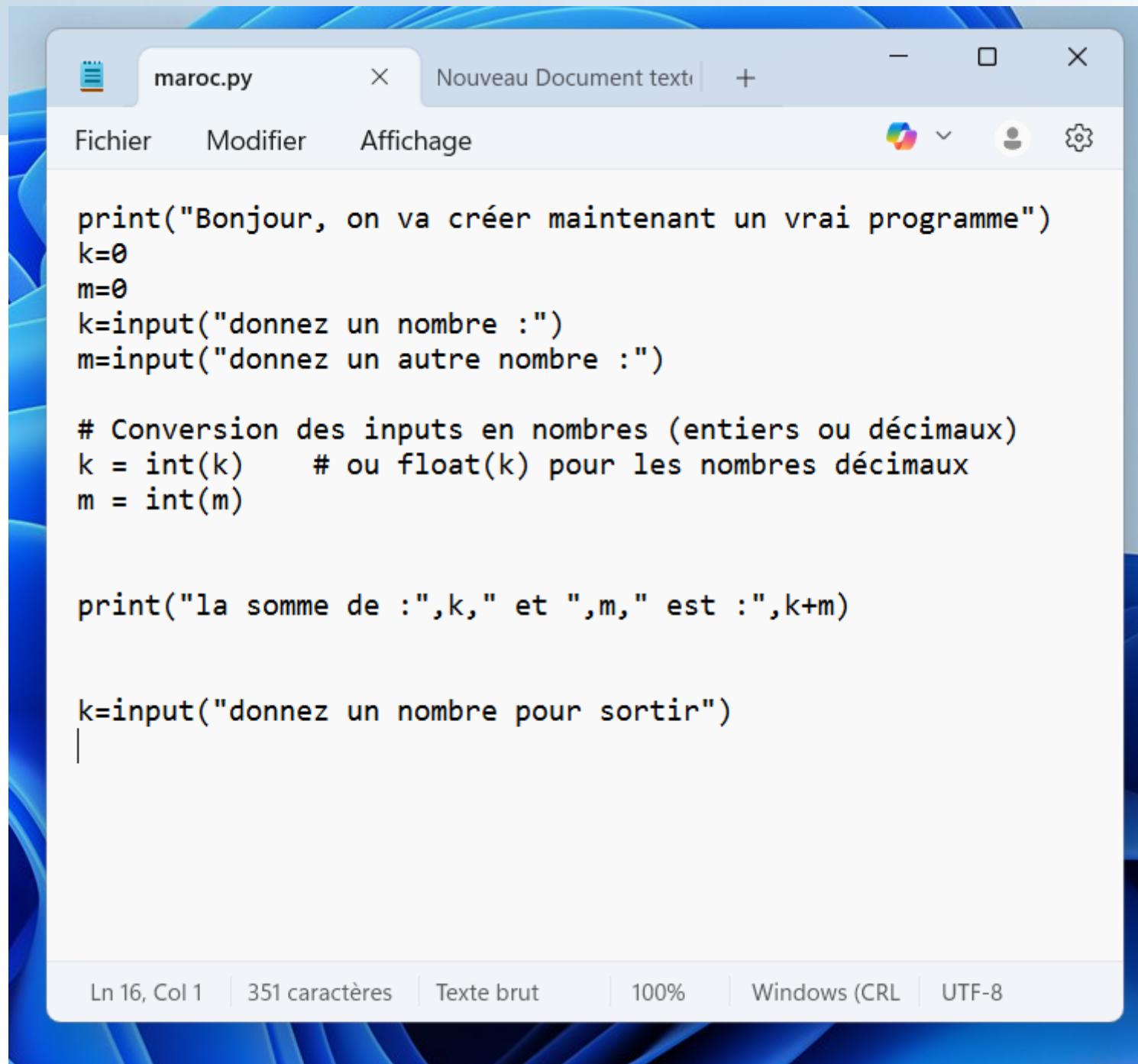
A red oval highlights the final line of code, which is `k=input("donnez un nombre pour sortir")`.

At the bottom of the editor, status information is displayed: "Ln 11, Col 1", "223 caractères", "Texte brut", "100%", "Windows (CRL)", and "UTF-8".

Résultat incorrect !!!!



```
C:\Users\DELL\AppData\Local + ▾
Bonjour, on va créer maintenant un vrai programme
donnez un nombre :5
donnez un autre nombre :7
la somme de : 5 et 7 est : 57
donnez un nombre pour sortir
```



The screenshot shows a text editor window with the following details:

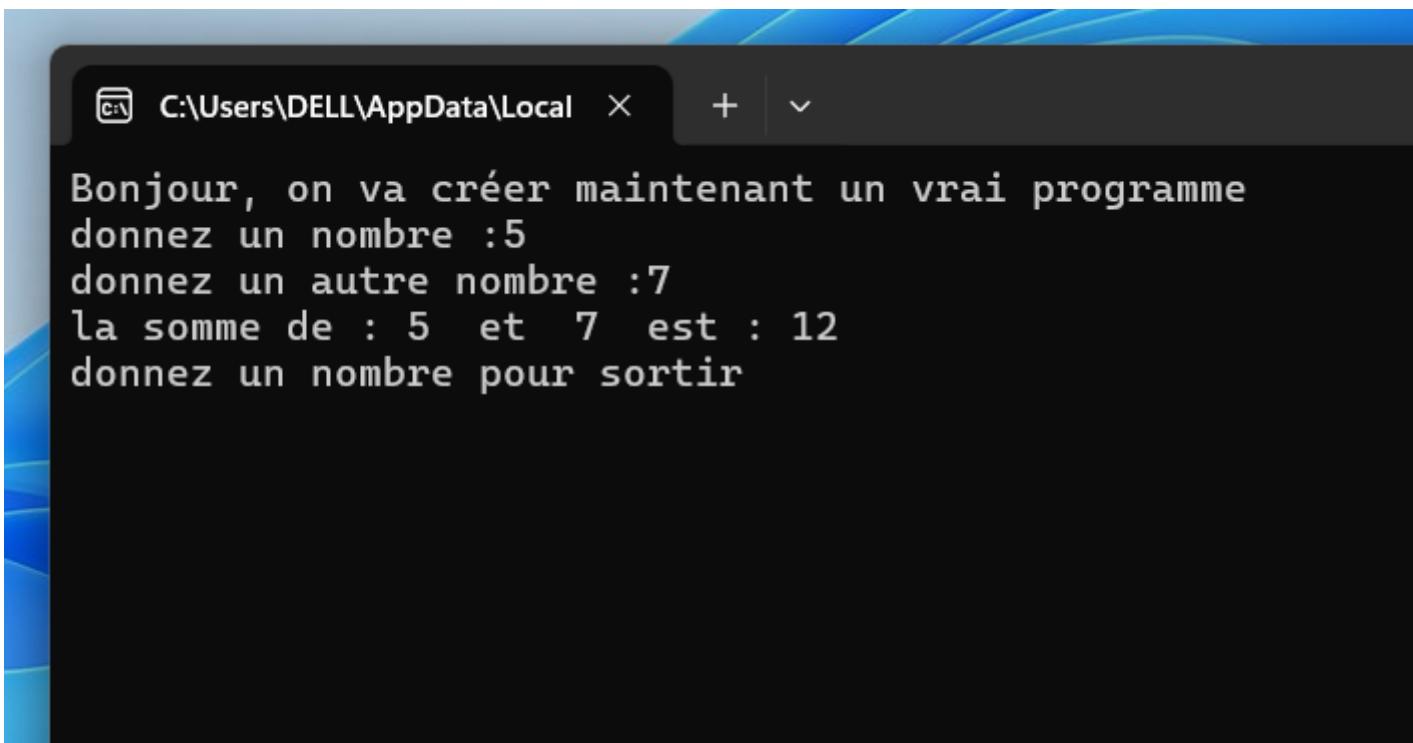
- Title Bar:** The title bar displays "maroc.py" and "Nouveau Document texte".
- Menu Bar:** The menu bar includes "Fichier", "Modifier", and "Affichage".
- Toolbar:** The toolbar features icons for file operations, a color palette, user profile, and settings.
- Code Area:** The main area contains the following Python code:

```
print("Bonjour, on va créer maintenant un vrai programme")
k=0
m=0
k=input("donnez un nombre :")
m=input("donnez un autre nombre :")

# Conversion des inputs en nombres (entiers ou décimaux)
k = int(k)      # ou float(k) pour les nombres décimaux
m = int(m)

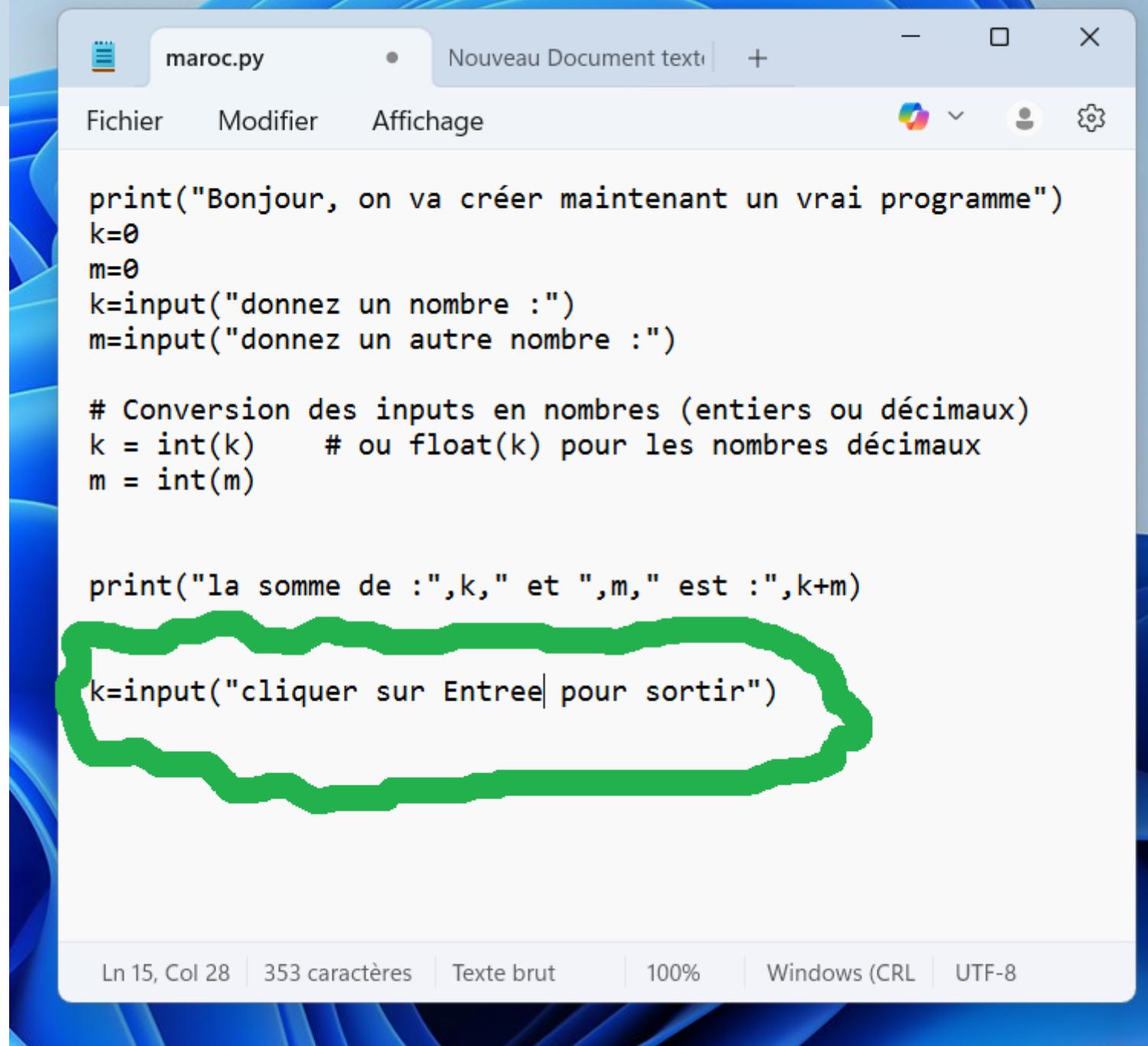
print("la somme de :",k," et ",m," est :",k+m)

k=input("donnez un nombre pour sortir")
```
- Status Bar:** The status bar at the bottom provides information about the document: "Ln 16, Col 1", "351 caractères", "Texte brut", "100%", "Windows (CRL)", and "UTF-8".



C:\Users\DELL\AppData\Local

```
Bonjour, on va créer maintenant un vrai programme
donnez un nombre :5
donnez un autre nombre :7
la somme de : 5 et 7 est : 12
donnez un nombre pour sortir
```



maroc.py • Nouveau Document texte +

Fichier Modifier Affichage

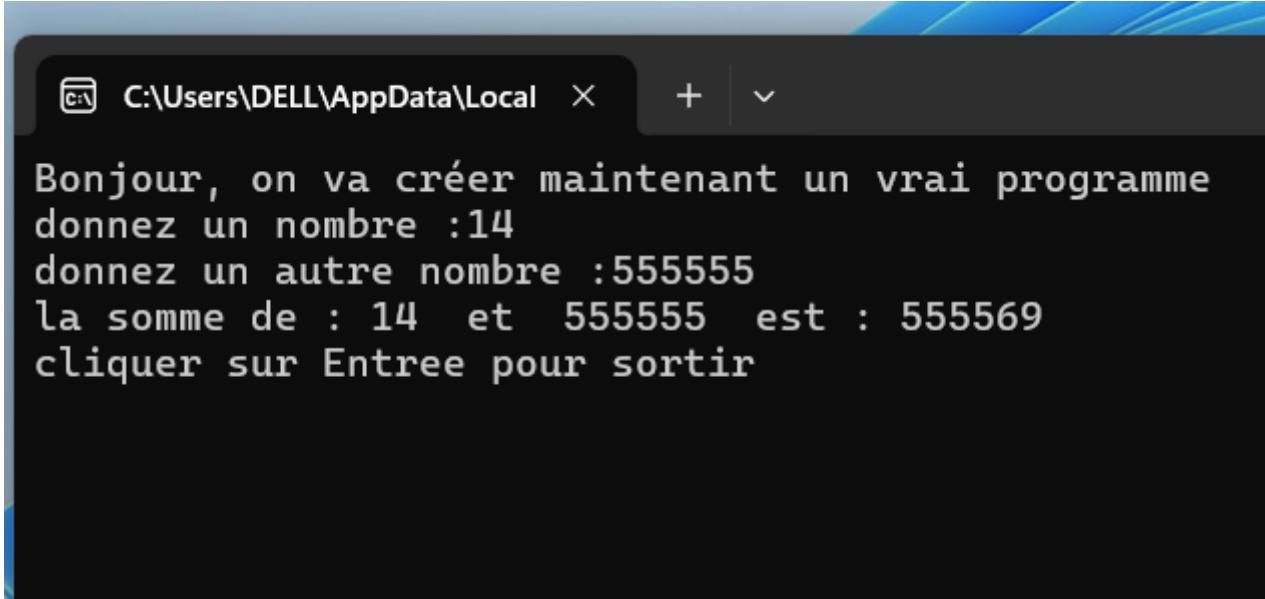
```
print("Bonjour, on va créer maintenant un vrai programme")
k=0
m=0
k=input("donnez un nombre :")
m=input("donnez un autre nombre :")

# Conversion des inputs en nombres (entiers ou décimaux)
k = int(k)      # ou float(k) pour les nombres décimaux
m = int(m)

print("la somme de :",k," et ",m," est :",k+m)

k=input("cliquer sur Entrée pour sortir")
```

Ln 15, Col 28 | 353 caractères | Texte brut | 100% | Windows (CRL | UTF-8



```
C:\Users\DELL\AppData\Local
```

Bonjour, on va créer maintenant un vrai programme
donnez un nombre :14
donnez un autre nombre :555555
la somme de : 14 et 555555 est : 555569
cliquer sur Entrée pour sortir

- On peut faire la même chose en utilisant langage C?



```
#include<stdio.h>

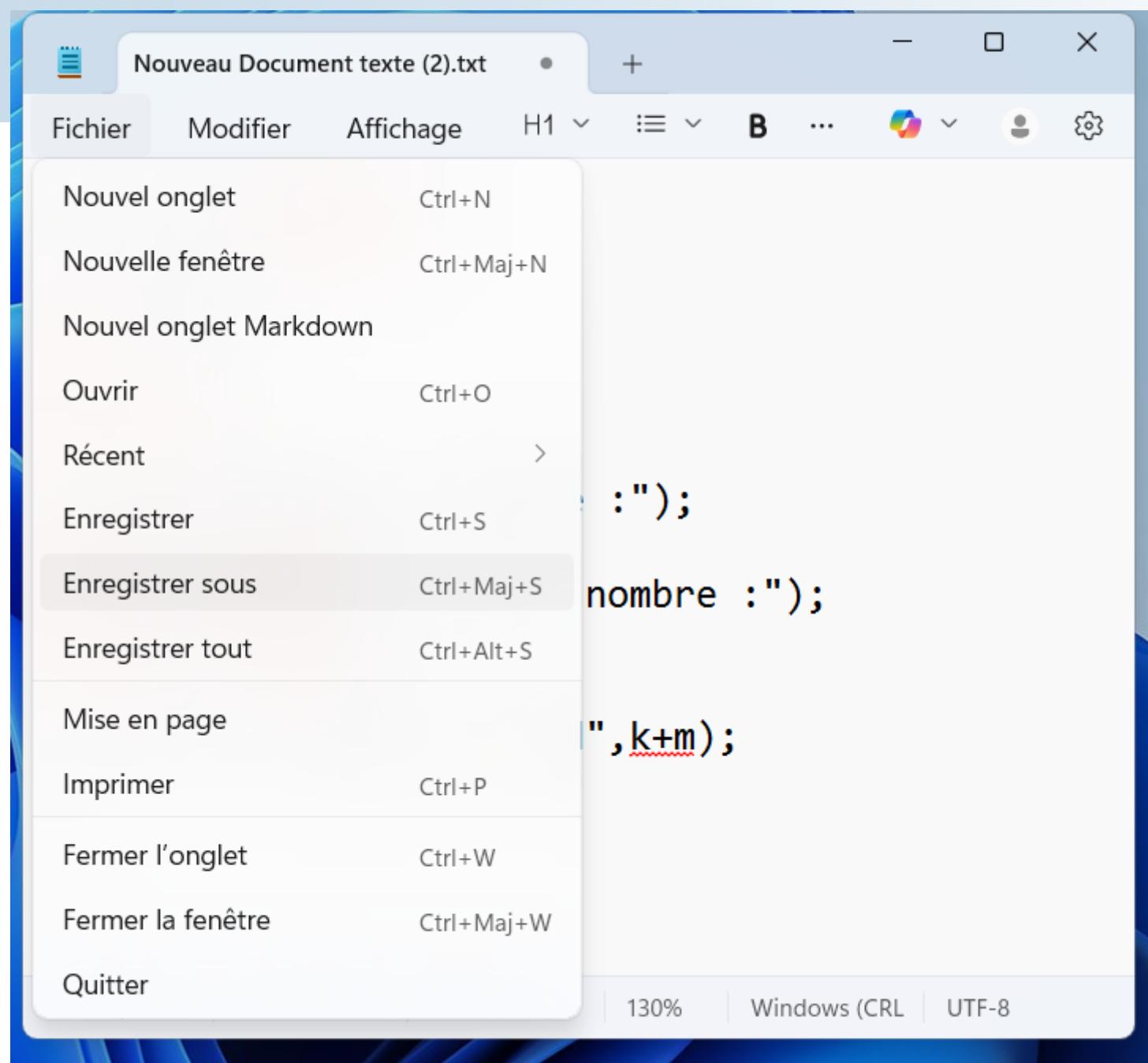
int main()
{
int k,m;

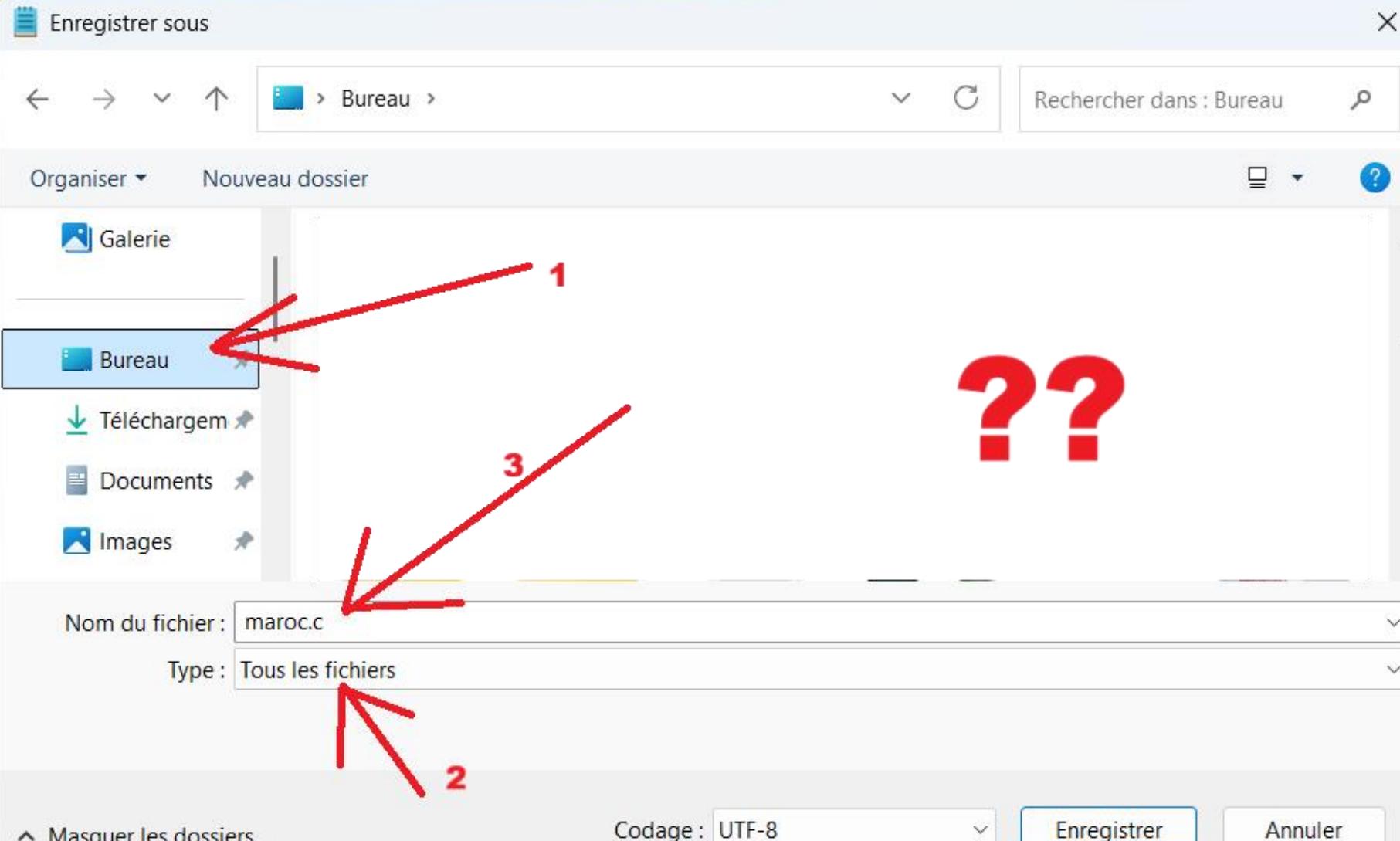
printf("donner un nombre :");
scanf("%d",&k);
printf("donner un autre nombre :");
scanf("%d",&m);

printf("la somme est :%d",k+m);

}
```

Ln 14, Col 2 | 175 caractères | Texte brut | 130% | Windows (CRL) | UTF-8





Ln 14, Col 2

175 caractères

Texte brut

130%

Windows (CTRL)

UTF-8

Enregistrer

Annuler





!!!??

- Donc ?
- Un code en C ne peut pas être interprété directement
- On doit le compiler pour pouvoir l'exécuter

Dev-C++ 4.9.9.2

File Edit Search View Project Execute Debug Tools CVS Window Help

Compile Ctrl+F9
Compile current file Shift+Ctrl+F9
Run Ctrl+F10
Parameters...
Compile & Run F9
Rebuild All Ctrl+F11
Syntax Check
Clean
Profile analysis
Program Reset Alt+F2

```
int k, m;  
  
printf("donner un nombre :");  
scanf("%d", &k);  
printf("donner un autre nombre :");  
scanf("%d", &m);  
  
printf("la somme est :%d", k+m);  
}|
```

Partie 2

Types de variables en Python

- **Types numériques**
 - Type int – Entier
 - Type float – Flottant
 - Type complex - Complexe
- **Type texte**
 - Type str - Chaîne de caractères
- **Types séquentiels**
 - Type list – Liste
 - Type tuple - Tuple
- **Types de mapping**
 - Type dict - Dictionnaire
- **Types d'ensembles**
 - Type set - Ensemble
- **Type booléen**
 - Type bool - Booléen
- **Types binaires**
 - Type bytes – Bytes
 - Type bytearray - Tableau de bytes

Types numériques

Les types suivants représentent des données numériques:

- Type int – Entier
- Type float – Flottant
- Type complex - Complexe

Type int - Entier

Nombre entier sans virgule

age = 25

quantite = -10

resultat = age + quantite

Type float - Flottant

Nombre à virgule flottante

prix = 19.99

temperature = -5.5

moyenne = (prix + temperature) / 2

Type complex - Complexe

Nombre complexe avec partie réelle et imaginaire

$$z1 = 2 + 3j$$

$$z2 = 4 - 1j$$

$$\text{somme} = z1 + z2$$

Les créateurs de Python ont bien pensé aux nombres complexes mathématiques.

En Python, le type `complex` représente exactement l'ensemble \mathbb{C} des nombres complexes des mathématiques.

La seule différence est la notation :

Mathématiques : $a + bi$ (avec i comme unité imaginaire)

Python : $a + bj$ (avec j comme unité imaginaire)

Ils ont choisi ' j ' car c'est la notation utilisée en ingénierie électrique et en programmation.

Mais les opérations sont les mêmes :

Addition, soustraction, multiplication, division

Pour l'exemple précédent on avait un nombre ‘somme’ donc

Partie réelle (somme.real)

Partie imaginaire (somme.imag)

C'est bien le même concept mathématique !

Le conjugué d'un nombre complexe $z=a+ib$ est le nombre $a-ib$, obtenu en changeant le signe de la partie imaginaire.

Géométriquement, le conjugué est le symétrique du nombre complexe par rapport à l'axe des réels dans le plan complexe

Conjugué (somme.conjugate())

Type texte

Le type suivant représente du texte

- Type str - Chaîne de caractères

Type str - Chaîne de caractères

Séquence de caractères entre guillemets

nom = "Ali"

message = 'Bonjour'

salutation = message + " " + nom

Types séquentiels

Les types suivants représentent des collections ordonnées

- Type list – Liste
- Type tuple - Tuple

Type list - Liste

Collection ordonnée et modifiable

```
fruits = ["pomme", "banane", "orange"]
```

```
fruits.append("kiwi")
```

```
premier = fruits[0]
```

Type tuple - Tuple

Collection ordonnée et non modifiable

```
coordonnees = (10, 20)
```

```
couleur_rgb = (255, 0, 0)
```

```
x = coordonnees[0]
```

Types de mapping

Le type suivant représente des associations clé-valeur

- Type dict - Dictionnaire

Type dict - Dictionnaire

Collection de paires clé-valeur

```
etudiant = {"nom": "Pierre", "age": 20}
```

```
etudiant["note"] = 15
```

```
nom_etudiant = etudiant["nom"]
```

Types d'ensembles

Les types suivants représentent des collections d'éléments uniques

- Type set - Ensemble

Type set - Ensemble

Collection non ordonnée d'éléments uniques

```
nombres = {1, 2, 3, 3, 2}
```

```
nombres.add(4)
```

```
nombres.remove(1)
```

Type booléen

Le type suivant représente des valeurs logiques

- Type bool - Booléen

Type bool - Booléen

Valeur logique True ou False

`est_actif = True`

`est_vide = False`

`resultat = est_actif and est_vide`

Types binaires

Les types suivants représentent des données binaires

- Type bytes – Bytes
- Type bytearray - Tableau de bytes

Type bytes - Bytes

Séquence d'octets immuable

```
data1 = b"SUPMTI"
```

```
data2 = bytes([83, 85, 80, 77, 84, 73])
```

Explication :

- S → code ASCII 83
- U → code ASCII 85
- P → code ASCII 80
- M → code ASCII 77
- T → code ASCII 84
- I → code ASCII 73

Donc data1 = data2

Type **bytearray** - Tableau de bytes

Séquence d'octets modifiable

```
buffer = bytearray(5)
```

```
buffer[0] = 65
```

```
buffer[1] = 66
```

Partie 3 : Structures conditionnelles et Structures itératives

Structures conditionnelles

Les instructions qui permettent d'exécuter du code selon des conditions

Structure if - Si simple

Exécute un bloc si la condition est vraie

age = 18

if age >= 18:

 print("Majeur")

Structure if-else - Si sinon

Exécute un bloc si vrai, un autre si faux

```
note = 12
```

```
if note >= 10:
```

```
    print("Validé")
```

```
else:
```

```
    print("Non validé")
```

Structure if-elif-else - Si sinon si

Teste plusieurs conditions successivement

note = 15

if note >= 16:

 print("Très bien")

elif note >= 14:

 print("Bien")

elif note >= 12:

 print("Assez bien")

else:

 print("Passable")

Structures itératives

Les instructions qui répètent l'exécution d'un bloc de code

Boucle for - Pour

Répète pour chaque élément d'une séquence

```
fruits = ["pomme", "banane", "orange"]
```

```
for fruit in fruits:
```

```
    print(fruit)
```

Boucle while - Tant que

Répète tant qu'une condition est vraie

```
compteur = 0
```

```
while compteur < 5:
```

```
    print(compteur)
```

```
    compteur += 1
```

Partie 4 : Fonctions

Fonctions

Blocs de code réutilisables avec des paramètres

Définition de fonction

Crée une fonction avec def et des paramètres

```
def addition(a, b):
```

```
    resultat = a + b
```

```
    return resultat
```

Appel de fonction

Utilise une fonction avec des arguments

```
somme = addition(5, 3)
```

```
print(somme) # Affiche 8
```

Partie 5 : l'équivalent des tableaux en Python ?

Tableau en C

```
#include <stdio.h>

int main() {
    int nombres[5] = {10, 20, 30, 40, 50};

    printf("%d\n", nombres[0]);

    nombres[1] = 25;

    for(int i = 0; i < 5; i++) {
        printf("%d ", nombres[i]);
    }

    return 0;
}
```

L'équivalent en Python (liste) :

```
nombres = [10, 20, 30, 40, 50]
```

```
print(nombres[0])
```

```
nombres[1] = 25
```

```
for nombre in nombres:
```

```
    print(nombre, end=" ")
```

Partie 6 : l'équivalent des structures en Python ?

Structure en C :

```
#include <stdio.h>

struct Etudiant {
    char nom[50];
    int age;
    float note;    };

int main() {
    struct Etudiant etudiant = {"Mohamed", 20, 15.50};

    printf("%s\n", etudiant.nom);

    return 0;    }
```

les dictionnaires Python sont la structure qui se rapproche le plus des struct en C.

Python (dictionnaire) :

```
etudiant = {"nom": " Mohamed ", "age": 20, "note": 15}  
print(etudiant["nom"])
```

Partie 7 : Définition simple d'une classe et son utilisation

ATTENTION : ce n'est pas un cours, c'est juste un rappel + un exemple de classe, vous n' allez pas comprendre quelques mots clés mentionnés dans le prochain exemple, pas grave, on le verra dans le cours prochainement

Voici la définition la plus simple d'une classe en Python :

Définition simple d'une classe :

```
class NomDeLaClasse:  
  
    def __init__(self, parametres):  
        # Initialisation des attributs  
        self.attribut = valeur  
  
    def methode(self):  
        # Code de la méthode  
        pass
```

exemple avec la classe Etudiant

```
class Etudiant:
```

```
    def __init__(self, nom, age, note):      # nom, age et note sont des attributs
        self.nom = nom
        self.age = age
        self.note = note
```

```
    def afficherMesInfos(self):
```

```
        print(f"Nom: {self.nom}, Age: {self.age}, Note: {self.note}")
```

Création de 2 étudiants (instanciation de 2 étudiants)

```
etud1 = Etudiant("Ali", 20, 15.5)
```

```
etud2 = Etudiant("Rym", 22, 14.0)
```

Appel de leurs méthodes

```
etud1.afficherMesInfos()
```

```
etud2.afficherMesInfos()
```