# Programmation sous python – 2025-2026

Deuxième chance
25 juin, 2026 — 12:45-14:45

À lire. Quelques consignes importantes pour la notation :

— Les scripts doivent être commentés et syntaxiquement corrects : un script qui génère une erreur de syntaxe est systématiquement sanctionné (pas de " petite croix rouge " sous Spyder !)
— Internet est autorisé ainsi que vos travaux en séances. Les échanges de mails et l' utilisation d'intelligences artificielles (type ChatGPT) sont interdits.
— Le non-respect des consignes est aussi pénalisé (si on demande une fonction `somme(n)`, on ne veut pas trouver une fonction `SOMME(n)` ou encore `Somme(N)` comme réponse).
— Votre script `cc3_25-26.py` est à déposer sur Moodle dans l'espace approprié dans `UE 4 : Programmation sous Python`.
— Le barème est sur 24 : **il n'est pas nécessaire de tout faire pour avoir la note maximale**. Les question avec (*) sont plus difficiles et chacune vaut deux points.

---

**Exercice 1** (Transformation itérée de chaînes et invariants numériques)**.** Le but de cet exercice est de transformer une chaîne de caractères en séparant ses caractères selon la parité de leurs indices, puis d'étudier la dynamique induite par cette transformation.

1) Écrire une fonction `separer_parite(stg)` qui prend en entrée une chaîne de caractères `stg` et renvoie une nouvelle chaîne `stgm` construite ainsi :

- Si la longueur de `stg` est $\leq 2$, la fonction renvoie `stg` inchangée.
- Sinon, elle concatène d'abord tous les caractères situés à des indices **pairs** $(0, 2, 4, \dots)$, suivis de tous les caractères situés à des indices **impairs** $(1, 3, 5, \dots)$.

(*Indication:* `"python"` (indices : p :0, y :1, t :2, h :3, o :4, n :5) doit devenir `"ptoyhn"`.) Vérifier votre fonction sur `"python"` et sur `"algorithme"`.

2) On applique cette fonction de manière itérative. En partant de $\texttt{stg}_0 =$ `"rattrapage"`, on définit la suite par $\texttt{stg}_{k+1} = \texttt{separer\_parite}(\texttt{stg}_k)$.

Écrire un script permettant de déterminer s'il existe un entier $1 \leq n \leq 100$ tel que $\texttt{stg}_n =$ `"rattrapage"`. Si oui, afficher ce plus petit entier $n$ (appelé période).

**Exercice 2.** Soit un entier $A \geq 10$.

1) Écrire une fonction $\texttt{c10}(A)$ qui renvoie la liste des chiffres de $A$ en base 10.

2) Écrire une fonction $\texttt{produit\_des\_sommes}(A)$ qui

a) sépare les chiffres de $A$ selon que leur **valeur** est un carré parfait $(0, 1, 4, 9)$ ou non $(2, 3, 5, 6, 7, 8)$

b) calcule $Q$, la somme des carrés des carrés parfaits d'un côté et $L$, la somme des autres de l'autre

c) renvoie le produit $(Q + 1)(L + 1)$. (*Indication:* Pour $A = 143$, les carrés parfaits sont 1 et 4 (somme $= 1^2 + 4^2 = 17$), l'autre est 3 (somme $= 3$). Le produit renvoyé est $(17 + 1) \times (3 + 1) = 72$.)

3) On définit la suite $(u_n)_n$ par $u_0 = A$, et

$$u_{n+1} = \texttt{produit\_des\_sommes}(u_n).$$

Pour $A = 1359$ calculer les 20 premiers termes de la suite $(u_n)_n$. La suite (finie obtenue) est-elle périodique à partir d'un certain rang ? Afficher votre réponse sur le terminal et la période si c'est le cas.

4*) Écrire une fonction $\texttt{periode}(A)$ qui retourne la période précédente pour un $A \geq 2$ quelconque.

**Exercice 3** (Suites de produit et seuils numériques — Échelle semi-log)**.** Soit $x \in [0, 1[$ un paramètre fixé. On s'intéresse au produit infini convergent ou divergent défini par passage à la limite sur $n \geq 1$ par

$$P_n(x) = \prod_{k=1}^{n} \left(1 + x^k\right) = (1 + x)(1 + x^2) \cdots (1 + x^n).$$

On pourrait prendre $P_0(x) = 1$ pour tout $x$ si on a besoin.

On rappelle qu'en Python, la fonction exponentielle et le logarithme népérien s'obtiennent via le module `numpy` (ou `math`) avec `numpy.exp` et `numpy.log`.

1) Écrire une fonction `produit(n, x)` qui prend en entrée un entier $n$ et un réel $x$ et qui renvoie la valeur de $P_n(x)$. Que renvoie la fonction pour $(n, x) = (50, 0.5)$ et pour $(100, 0.95)$ ?

2) Pour $x$ proche de 1, le produit grandit rapidement. Écrire une fonction `indice_pour_seuil(S, x)` qui prend en entrée deux réels $S$ (un seuil, avec $S > 1$) et $x \in [0, 1[$, et qui renvoie $-1$ ou un entier $N$ selon les cas suivants :

- $N$ est le premier entier $\leq 10000$ tel que $P_N(x) \geq S$ ;
- $-1$ sinon.

Que renvoie la fonction pour $(S, x) = (10, 0.90)$ et pour $(S, x) = (100, 0.98)$ ?

3) On considère la liste de paramètres $X = [0.80, 0.82, 0.84, \ldots, 0.98]$ (allant de 0.80 à 0.98 inclus avec un pas de 0.02).

a) Construire la liste $Y$ contenant les entiers $N_{S,x}$ obtenus pour un seuil fixé $S = 50$ lorsque $x$ parcourt la liste $X$.

b) En utilisant la bibliothèque `matplotlib`, représenter sur un même graphique :

- en bleu, la ligne polygonale définie par $X$ et $Y$
- en rouge, la même ligne polygonale obtenue pour le seuil $S = 500$.

On veillera à configurer une grille, un titre explicite et une légende adaptée.

4*) Représenter graphiquement la valeur du produit comme fonction de $x \in [0, 0.98]$ pour $n = 5$ en bleu et pour $n = 30$ en rouge. Afin d'éviter l'écrasement des données provoqué par les valeurs de $x$ très proches de 0.98, reprendre les représentations graphiques en appliquant $\log_{10}$ aux ordonnées. Comment analysez-vous visuellement l'effet de ce changement d'échelle ?

**Barème indicatif : 8 — 8 — 8**