

PARCOURS 7 :

Utiliser les listes

Utiliser le module graphique

EXERCICE 39 : Parcourir une liste

Enregistrer la liste : $L=['b','ou','n','j','o','u','r',11,2,3]$

Compléter le tableau :

Séquence à taper	Résultat
	'j'
	['j','o','u','r',11,2,3]
	['n','j']
	['b','ou','n']
	['ou','n','j']
	['b','a','ou','n','j','o','u','r',11,2,3]
	['b','n','j','o','u','r',11,2,3]

EXERCICE 40 : Listes en compréhension

1°) Créer la liste des multiples de 13 inférieurs à 300.

2°) a) Créer la liste des cubes d'entiers inférieurs à 15.

b) Adapter la liste précédente afin de ne garder que les cubes des entiers impairs.

3°) a) Créer une liste de nombres aléatoires de 20 lancers d'un dé à 6 faces parfaitement équilibré.

b) Afficher le résultat le plus fréquent.

EXERCICE 41 :

Soit u la suite définie par :
$$\begin{cases} u_0 = 0,5 \\ u_{n+1} = u_n(1 - u_n) \end{cases}$$

1°) Donner la liste des 10 premiers termes.

2°) Représenter le nuage de points $(n ; u_n)$ lorsque $n < 50$.

EXERCICE 42

1°) Créer une fonction : `disparition_e()` qui supprime les e d'un texte choisi par l'utilisateur.

2°) Créer une fonction qui renvoie toutes les occurrences communes de deux listes.

EXERCICE 43 :

Créer une liste des coefficients directeurs des sécantes à la fonction carrée, pour des valeurs allant de - 4 à 4 avec un pas de 0,1.

EXERCICE 44 :

La suite de Fibonacci est définie par :
$$\begin{cases} F_0 = 1 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n \end{cases}$$

1°) Créer une fonction qui renvoie la liste des n premiers nombres de la suite de Fibonacci.

2°) Créer la liste des rapports $\frac{F_{n+1}}{F_n}$.

3°) Représenter le nuage de points formé par les rapports des termes de la suite de Fibonacci.

Pour aller plus loin

EXERCICE 45 : Approximation de π par la méthode de Monte-Carlo

On dispose d'un carré de côté 1 ; son aire vaut 1. À l'intérieur de ce carré, on inscrit un quart de cercle de rayon 1 et de centre le coin inférieur gauche du carré, son aire vaut $\frac{\pi}{4}$.

Si on choisit au hasard un point du carré, la probabilité qu'il soit à l'intérieur du quart de cercle est donc de $\frac{\pi}{4}$.

Nous pouvons en déduire que $\frac{\text{nbre de points à l'intérieur du cercle}}{\text{nbre de points tirés}}$ va nous donner une approximation de $\frac{\pi}{4}$.

Écrire une fonction qui tire au hasard un point du carré et qui l'affiche sur la figure, en rouge s'il est à l'intérieur du cercle en bleu s'il est à l'extérieur et qui renvoie l'approximation de π .

EXERCICE 46 : Syracuse

La suite de Syracuse est une suite définie par :

$$\begin{cases} u_0 \in \mathbb{N} \\ u_{n+1} = \frac{u_n}{2}, \text{ si } n \text{ est pair} \\ u_{n+1} = 3u_n + 1, \text{ si } n \text{ est impair} \end{cases}$$

L'énoncé de la conjecture de la suite de Syracuse est : quel que soit le premier terme choisi, en appliquant l'algorithme de Syracuse, nous finissons toujours par obtenir le nombre 1.

On utilise le vocabulaire suivant :

- le vol, c'est l'ensemble des étapes de la suite
- l'altitude maximale du vol, c'est le plus grand nombre obtenu
- le temps du vol : c'est le nombre d'étapes qui sépare le nombre du début de l'apparition du premier 1

1°) Créer la fonction suivant(n), qui renvoie la première étape de la suite de Syracuse de n.

2°) Créer les fonctions vol(n), altitude_max(n), temps_de_vol(n).

3°) Représenter les points (n ; u_n) ils pourront être liés ou sous forme de nuage de points.

Pour les exercices de codages, nous serons amenés à utiliser les fonctions **chr**(n) qui renvoie le caractère qui a pour code ASCII n et **ord**(caractère) qui fait le travail en sens inverse.

EXERCICE 47 : Codage affine

1°) Écrire une fonction qui code un message en utilisant le codage affine : $y \equiv 11x + 7$ où x est le code ASCII d'une lettre claire et y celui d'une lettre codée.

On pourra ne travailler qu'avec des lettres majuscules non accentuées.

2°) Écrire une fonction qui code un message en utilisant un codage affine $y \equiv ax + b$ où a et b sont des entiers naturels qui feront partie des paramètres de la fonction.

EXERCICE 48 : Codage inversé

Codage inversé (a \leftrightarrow z, b \leftrightarrow y ..., m \leftrightarrow n)

Créer une fonction : codage_inverse(), qui, connaissant une phrase, la renvoie en utilisant un codage inversé.

Attention à l'espace qui doit rester un espace.