

# TP8 : Révisions et manipulations de fichiers

Démarrer le logiciel Spyder puis, dans le menu « Fichier » créer un nouveau fichier puis le sauvegarder avec un nom sous la forme TP8\_votrenom. Dans la suite, on sauvegardera régulièrement son travail.

## I Quelques révisions

### Exercice 1 Suite récurrente

---

On considère la suite  $(u_n)$  définie par récurrence par  $u_0 \in [0, 1]$  puis pour tout  $n \in \mathbb{N}$  :

$$u_{n+1} = \sin(u_n).$$

1. Écrire une fonction Python `suite_iter(n,u0)` calculant le  $n$ -ième terme de la suite de façon itérative.
  2. Écrire une fonction Python `suite_recur(n,u0)` calculant le  $n$ -ième terme de la suite de façon récursive.
  3. Écrire une fonction `graph()` qui représente graphiquement  $\sin$  et  $x \mapsto \sin x - x$  sur  $[0, 1]$  (le fenêtre ira de 0 à 1 pour les abscisses et les ordonnées).
  4. Dédurre de la question précédente que  $(u_n)_{n \in \mathbb{N}}$  est décroissante et tend vers 0.
- 

### Exercice 2 Palindromes

---

Un nombre palindrome est un nombre entier naturel qui se lit de la même manière dans les deux sens. Le plus grand palindrome qui est un produit de deux entiers naturels à deux chiffres est

$$9009 = 91 \times 99.$$

1. Tester `list(95421)` dans la console et examiner le résultat.
  2. Écrire une fonction `palindrome(n)` qui pour un entier  $n \in \mathbb{N}$  teste s'il s'agit d'un palindrome.
  3. Écrire une fonction `liste_pal(k)` qui retourne la liste de tous les palindromes inférieurs ou égaux à  $k$ .
  4. Déterminer le plus grand palindrome qui est le produit de deux entiers à trois chiffres.
- 

## II Suite implicite

### Exercice 3 Dichotomie

---

On rappelle que  $\sqrt[3]{5}$  est l'unique racine réelle de  $x \mapsto x^3 - 5$ . Écrire une fonction Python `dichotomie(erreur)` où `erreur` est un flottant qui avec une méthode de dichotomie donne une valeur approchée de  $\sqrt[3]{5}$  à `erreur` près sans utiliser `5**(1.0/3)`. On pourra revoir la méthode de dichotomie.

---

### Exercice 4 Représentation de suite implicite

---

On considère, pour tout  $n \in \mathbb{N}^*$  l'équation d'inconnue  $x \in ]0, +\infty[$  :

$$x + \ln x = n. \tag{E_n}$$

1. Montrer que pour tout  $n \in \mathbb{N}^*$ ,  $(E_n)$  a une unique solution qu'on notera  $u_n$ .
  2. Écrire une fonction `suite(n)` qui retourne une valeur approchée à 0.01 près de  $u_n$ . On s'inspirera de l'exercice précédent.
  3. Représenter, en écrivant une fonction `graph()`, sur un même graphique les suites  $(u_n)_{n \in \mathbb{N}^*}$  et  $(n)_{n \in \mathbb{N}^*}$  pour  $n = 1, \dots, 100$ . Que peut-on conjecturer ?
  4. *Question facultative* : Démontrer la conjecture précédente.
-

### III Manipulation de fichiers

Dans cette partie, on va voir comment utiliser Python pour écrire dans un fichier texte (**.txt** lorsqu'on travaille sous Windows). Avant de commencer l'exercice, on suivra attentivement la procédure mentionnée par l'enseignant. On appelle désormais « Répertoire de travail » le dossier dont l'adresse est mentionné en haut à droite de Spyder.

#### Exercice 5 *Création et écriture dans un fichier*

---

1. Taper dans l'éditeur de texte les lignes de code :

```
fichier_ecriture = open('test.txt','w')
fichier_ecriture.write('ceci est une phrase')
fichier_ecriture.close()
```

puis les exécuter (mettre en surbrillance, clic droit puis « exécuter... »). Ouvrir dans l'explorateur Windows le répertoire de travail et expliquer les lignes précédentes. Fermez ensuite le fichier texte que vous avez ouvert.

2. Taper dans l'éditeur de texte les lignes de code :

```
fichier_ecriture = open('test_1.txt','a')
fichier_ecriture.write('ceci est une autre phrase')
fichier_ecriture.close()
```

puis les exécuter (mettre en surbrillance, clic droit puis « exécuter... »). Interpréter puis exécuter à nouveau les lignes de code précédentes et interpréter à nouveau.

3. Créer une fonction `ecriture(n)` qui crée un fichier `test_2.txt` comportant les lignes

```
ceci est la phrase 1
:
:
ceci est la phrase n
```

On n'oubliera pas de convertir les entiers en `str` et l'instruction `'\n'` permet de sauter un ligne.

---

#### Exercice 6 *Lecture d'un fichier*

---

1. Télécharger sur le site internet <http://www.pcsi-troyes.fr/> sous l'onglet **Informatique** puis **Informatique** le fichier `decimales.txt` qui contient les mille premières décimales de  $\pi$  et le placer dans le répertoire de travail (on fera un clic et « enregistrer la cible sous » puis exécuter

```
fichier_lecture = open('decimales.txt','r')
```

Ceci permet de lire le fichier sans pouvoir écrire dans celui-ci.

2. Exécuter `s = fichier_lecture.read()` puis afficher le contenu de `s`.
  3. Créer une fonction `conversion()` qui renvoie la liste des mille premières décimales de  $\pi$ . On éliminera notamment les `'\n'` de la variable `s` précédente et on convertira les chaînes de caractères en entiers.
  4. Écrire une fonction `frequence(n)` qui déterminer la fréquence d'apparition de `n` dans les mille premières décimales de  $\pi$ .
- 

### IV Calendrier

#### Exercice 7 *Années bissextiles*

---

On rappelle qu'une année est bissextile (comporte 366 jours au lieu de 365) lorsque

1. l'année est divisible par 4 et non divisible par 100 ;
2. l'année est divisible par 400.

Écrire une fonction Python `bissextile(n)` qui renvoie `True` si l'année est bissextile et `False` sinon.

---

---

**Exercice 8** *Premier janvier et jour*

---

1. Sachant que le premier janvier 2013 est tombé un dimanche, écrire une fonction `jour(n)` retournant le jour dans la semaine où tombe le premier janvier d'une année. On prendra comme convention : 1 pour lundi, 2 pour mardi... jusqu'à 7 pour dimanche. Vérifier avec les exemples suivants : le premier janvier de 2042 tombe un dimanche et le premier janvier 2000 est tombé un samedi.
  2. Écrire une fonction prenant en entrée une date (triplet d'entiers jour/mois/année) et retournant le jour dans la semaine correspondant à cette date.
- 

**Exercice 9** *Age*

---

Écrire une fonction Python qui prend en entrée deux liste à trois éléments de la forme `[annee,mois,jour]` (une pour la date du jour et une pour la date de naissance) et qui renvoie l'âge d'une personne.

---

## V Compléments

**Exercice 10** *Famille de polynômes*

---

On pose  $P_0 = 2$ ,  $P_1 = X$  et pour tout  $n \in \mathbb{N}$ ,

$$P_{n+2}(X) = XP_{n+1}(X) - P_n(X).$$

1. Écrire une fonction Python `liste_coeff(n)` qui renvoie la liste des coefficients de  $P_n$ .
2. Représenter, sur un même graphique  $P_0, P_1, \dots, P_{10}$ .
3. Montrer que pour tout  $n \in \mathbb{N}$  et  $\theta \in \mathbb{R}$ ,

$$P_n(2 \cos \theta) = 2 \cos(n\theta).$$

4. Déterminer les racines de  $P_n$  (on confirmera avec le graphique précédent).
-