

# I11 - Programmation I : Python

## TP 7

---

### Objectifs

- Utiliser des fichiers
- Implanter des modules

---

### Notions

- Modèle de données fichier
- Modularité

---

### Consignes

- Les étudiants de la licence SI utiliseront un terminal de commande Unix pour toutes les manipulations du système de fichiers.
- Les étudiants des licences Mass/Maths/PC utiliseront le navigateur de fichiers.
- Les scripts doivent commencer par

```
# fichier: modele_script.py  
# auteur: Nom prenom
```

---

## 1 Fichiers

### EXERCICE 1. Bataille navale

(source Wikipédia) La bataille navale, appelée aussi touché-coulé, est un jeu de société dans lequel deux joueurs doivent placer des « navires » sur une grille tenue secrète et tenter de « toucher » les navires adverses. Le gagnant est celui qui parvient à torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient. C'est un jeu de hasard.

|    | A | B | C | D | E | F | G | H | I | L |
|----|---|---|---|---|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |   |   |   |   |
| 3  |   |   |   |   |   |   |   |   |   |   |
| 4  |   |   | X |   |   |   |   |   |   |   |
| 5  |   |   |   |   |   | X | X |   |   |   |
| 6  |   | X |   |   |   |   |   | X |   | X |
| 7  |   |   |   | X |   |   |   |   |   | X |
| 8  | X | X |   |   |   |   |   | X |   |   |
| 9  |   |   |   |   |   |   |   |   |   |   |
| 10 |   |   |   |   |   |   |   |   |   |   |

1. Ecrire un module qui demande pour chaque type de bâtiments (sous-marin, torpilleur, destroyer, croiseur, porte-avion), dont la taille varie de 1 case à 5 cases, sa position initiale et son orientation (horizontale ou verticale) puis qui enregistre ces informations dans un fichier de nom `joueur_h`. Les éventuelles erreurs de saisie doivent être signalées (y compris l'impossibilité de placer le navire aux coordonnées choisies à cause de sa taille ou de la présence d'autres bâtiments). Dans cette version, les bâtiments peuvent se toucher.

Le codage du fichier est le suivant : chaque ligne comporte un caractère indiquant le type de bâtiment ('s', 't', 'd', 'c' ou 'p'), un espace, un nombre indiquant l'abscisse de sa position initiale, un espace, un nombre indiquant l'ordonnée de sa position initiale, un espace, une lettre indiquant son orientation ('h' ou 'v'). Exemple : d 2 5 v.

Remarques : le système de coordonnées est indirect (cf. figure ci-dessus) ; la position initiale est celle de la 1ère case du bâtiment, les autres cases sont déterminées en respectant le repère choisi (indirect) et l'orientation du bâtiment.

2. Adapter le module précédent afin de générer aléatoirement (cf. module `random`) des positions et des orientations pour chaque type de bâtiment (en vérifiant si celles-ci respectent les règles du jeu) puis d'enregistrer le résultat dans un fichier de nom `joueur_o`, suivant le format de la question 1.
3. Ecrire un script qui utilise le module `cng` pour afficher la grille, comportant tous les bâtiments, de l'un des deux joueurs (humain ou ordinateur), au choix de l'utilisateur. Pour plus d'informations sur le module `cng`, une fois chargé celui-ci par la commande `import`, taper `help(cng)`.

## 2 Modules

### EXERCICE 2. Séquence biologique

(source Wikipédia) Une séquence biologique est généralement un fichier informatique au format texte contenant la séquence d'une macromolécule biologique. Ce peut être la séquence des bases de l'ADN (acide désoxyribonucléique) à la sortie d'un séquenceur de gène ou la traduction de cette séquence en acide aminés. Il s'agit d'un fichier contenant l'information génétique brute.

1. Ecrire une fonction qui génère aléatoirement puis retourne une chaîne de 100 caractères correspondant à une séquence d'ADN i.e une suite formée uniquement de 4 lettres correspondant aux quatre nucléotides formant l'enchaînement de l'ADN : A pour adénine, G pour guanine, T pour thymine, C pour cytosine.
2. Ecrire une fonction qui, à partir d'une séquence de nucléotides sous la forme d'une chaîne de caractères, détermine le pourcentage de chaque type de nucléotide et retourne le résultat sous la forme d'un tuple de 4 pourcentages.
3. Ecrire une fonction qui recherche une sous-séquence dans une séquence ADN et retourne l'index du 1er nucléotide si la sous-séquence est présente et -1 sinon (on n'utilisera pas la méthode `find` pour cet exercice sauf pour comparaison).
4. Ecrire la fonction qui recherche la plus longue sous-séquence commune de deux séquences ADN fournies en paramètre et retourne celle-ci. L'algorithme « force brute » est le suivant :

---

**Algorithme** Plus longue sous-séquence commune(*seq1*, *seq2*)

**données**

Séquences *seq1*, *seq2*

**variables**

Entiers *i1*, *j1*, *i2*, *j2*

Séquence *seqcom*

---

---

```

seqcom ← ""
i1 ← index du 1er caractère de seq1
tantque (on n'a pas parcouru la séquence seq1) faire
  i2 ← index du 1er caractère de seq2
  tantque (on n'a pas parcouru la séquence seq2) faire
    on recherche le caractère de seq1 dans seq2
    si (on a trouvé le caractère) alors
      j1 ← i1
      j2 ← i2
    finsi
    on compare la suite des caractères dans seq1 (j1) et seq2 (j2)
    si (la longueur de la séquence trouvée est plus longue) alors
      on l'enregistre dans seqcom
    finsi
  i2 ← i2 + 1
fintq
i1 ← i1 + 1
fintq
retourner(seqcom)

```

---

Par exemple, pour les séquences **AAGTACTGACGGTTCAG** et **GTGTCATTACTACG**, le résultat est **TACT**.

5. Ecrire le test unitaire c'est à dire un bloc d'instructions qui teste toutes les fonctions du module et qui les valide (il doit donc pouvoir se lancer comme un script indépendant ou être importé comme un module, au choix).

### 3 Pour aller plus loin ...

#### EXERCICE 3. Bataille navale

Reprendre l'exercice 1 en affichant la grille du joueur humain puis en proposant la saisie de 3 couples de coordonnées pour le joueur humain, la visualisation du résultat (par exemple par l'affichage d'un petit cercle blanc si raté ou d'un petit cercle rouge si touché et d'un message informant d'un bâtiment éliminé) ainsi que la simulation d'un adversaire qui fera 3 propositions à son tour et dont les conséquences seront visibles dans la fenêtre graphique (par exemple en colorant en rouge les carrés des bâtiments touchés). Et ainsi de suite jusqu'à ce que l'application détecte l'élimination de l'un des deux joueurs.

(source Wikipédia) On rappelle qu'il existe une stratégie efficace pour mieux repérer la flotte ennemie : jouer ses tirs en croix. En admettant que le navire le plus petit du jeu fasse 2 cases, il suffit de jouer une case sur deux pour le repérer. Ce qui évite de jouer les cases qui sont entourées par vos tirs parce que vous savez qu'aucun bateau adverse ne peut s'y trouver.