

# Livre Dont Vous Êtes Le Héros ou L'Héroïne

## Objectif du TP

Nous souhaitons développer un moteur de Livre Dont Vous Êtes le Héros ou l'Héroïne, en langage python. Il s'agit d'une gamme inspirée des jeux de rôles, popularisée dès les années 1980 en France. Leur principe est de proposer un découpage de l'histoire en paragraphes numérotés, ainsi qu'une navigation non-linéaire. Cette dernière permet de vivre une aventure en circulant au sein du livre en suivant les indications de fin de paragraphe, telles que *"si vous désirez tourner à gauche en **direction des montagnes**, lisez le 34. Si le chemin de droite, au travers **des marécages**, vous semble en valoir les risques, **allez au 238**. Si vous préférez faire **halte au village** dont vous apercevez les fumées au loin, **rendez-vous au 125**."*

Ce TP est pour nous une occasion (et un prétexte) pour s'exercer à la Programmation Orientée Objet en python. Cela nous permet, en outre, de voir nos premiers diagrammes de classe, modélisation indispensable à sa bonne compréhension.

## Exercices

### Chapitre

Chapitre
-int orGagne
-int pvPerdus
-str texte
-int[] listeLiens
+str getTexte()
+int getPvPerdus()
+int getOrGagne()
+int[] getListeLiens()

Un objet de classe Chapitre sert surtout à stocker des informations.

Pour commencer, on considère qu'un chapitre fait parfois perdre des points de vie ou gagner de l'or à notre personnage. Il y aura du texte.

Quant à la liste de liens, il s'agit d'une simple liste d'entiers, lesquels sont les numéros de chapitres accessibles.

Bien entendu, il nous faudra aussi un constructeur de Chapitres.

Définissez et implémentez la classe Chapitre.

## TP Bases de l'orienté objet

## Livre

Un Livre est un ensemble de Chapitres. Dans notre cas présent, c'est aussi l'équivalent du jeu en lui-même. Il ne nous est donc pas forcément utile (ou du moins, pas dans ce présent TP) d'en faire une classe et un objet. Il s'agit, en fait, de notre programme principal.

Nous avons donc une variable livre, qui contient une liste de variables de classe Chapitre.

## Personnage

Personnage
-str nom -int pv -int or
+str getNom() +int getPv() +int getOr() +int perdPv(int) +int gagneOr(int)

Pour commencer modeste, on considère qu'un personnage possède juste un nom, des points de vie, et de l'or.

La classe Personnage doit donc avoir des méthodes *get* pour ces mêmes attributs (on appelle ces méthodes des *accesseurs*).

Nous avons aussi besoin de méthodes pour perdre des points de vie et pour gagner de l'or (on appelle ces méthodes des *mutateurs*).

On ne prévoit a priori pas la possibilité pour le personnage de changer de nom en cours d'aventure.

## Parcourir le Livre

La boucle principale du jeu consiste, tout d'abord, à afficher le texte du chapitre courant (au tout début du jeu, il s'agit du chapitre 0, accessible via la variable livre[0]). Ensuite, on rappelle au joueur ou à la joueuse les données concernant son personnage (nom, points de vie, or). Puis on lui laisse la possibilité de faire le choix d'un nouveau chapitre, en vérifiant qu'il est bien accessible. Quand c'est chose faite, on peut recommencer en début de boucle.

La boucle principale s'arrête si les points de vie du personnage atteignent 0 ou moins, ou si le chapitre courant atteint une valeur déterminée à l'avance (le chapitre de fin de jeu).

## TP Bases de l'orienté objet

## Mais les listes, quelle galère !

Oui. Les listes présentent plusieurs problèmes dans la gestion de notre livre :

- Il faut créer la liste en un bloc, ou ajouter les entrées une par une avec *append()*.
- On peut avoir des difficultés à connaître le numéro de chaque chapitre.
- On ne peut pas avoir de "trous" dans notre numérotation, et on risque des bugs.

### La solution : utilisez un Dictionnaire

Si on considère que la variable `livre` n'est plus une liste mais un dictionnaire, elle s'instancie de la manière suivante :

`variable = {clef1:contenu1, clef2:contenu2, ..., clefN:contenuN}`

La clef peut être un entier, ou une chaîne de caractères par exemple...

Dans notre cas, cela ressemblera à...

```
livre = {1:chapitre1, 2:chapitre2, 7:chapitre7}
```

On peut ajouter une nouvelle valeur de façon bien plus simple qu'avec une liste.

```
livre[4] = chapitre4
```

Un dictionnaire s'utilise de la même façon qu'une liste, en remplaçant l'index par la clef.

Ici, on aura donc rien à modifier, car on remplace l'index de la liste (qu'on utilisait comme numéro de chapitre) par... le numéro du chapitre.

On va donc remplacer notre liste par un dictionnaire, et ce sera d'autant plus simple.

Retenez surtout que même si en python, liste et dictionnaire se ressemblent vraiment beaucoup (on dit qu'ils partagent une interface similaire), ce n'est pas le cas dans tous les langages.

## Améliorations possibles (servez-vous !)

- Importer un Livre depuis un fichier texte.
- Exporter/importer une sauvegarde (numéro courant & caractéristiques du personnage).
- Système de combat avec valeur de caractéristique & affichage des jets de dés.
- Inventaire + possibilité d'acheter ou d'utiliser des items (potions...).
- Ouvrir la possibilité de conditionner des déplacements à la possession d'items.