Python

Bastien Gorissen & Thomas Stassin

Année 2016

Table des matières

Chapitre 1

Level 1

1.1 Introduction

Dans le premier cours ("Introduction à la programmation"), on avait toucher aux principes fondamentaux de la programmation dîtes *procédurale*. Dans ce cours, qui peut être vu comme une suitre, nous irons plus en profondeur dans Python afin d'apprendre les bases de la programmation *orientée objets*.

1.1.1 Programmation orientée objets, oui mais en Python

Ce cours, reste un cours de Python, donc, bien que nous verrons la programmation orientée objet, nous n'irons pas en dehors de ce que Python propose. J'entend par là que certains "principes" de l'orienté objets, ne correspondant pas à la philosophie de Python [1], ne sont pas appliqué par ce langage, et donc ne seront pas illustré par ce cours [2].

1. "Keep it simple" (Garde çà simple).
2. Ces principe seront surement abordé dans d'autres cours comme le cours de C#.

1.2 Classes ?, Objets ?

Des *Objets* vous en avez déjà rencontré pas mal sans les avoir nommé comme tel. J'en veux pour preuve que dans Python, tout est *objet*[3].

Exemple :

```
a = 1
```

Dans ce morceau de code, `a` est un *object*. D'ailleurs, tant qu'on y est, la classe de `a` est `int`. Vous allez me dire que `int` c'est son *type*. C'est vrai aussi, en Python le *type* et la *classe*.

Bon cela ne nous dit pas vraiment ce qu'est une classe.

1.2.1 RPG Like

On peut faire le parallèle entre les la *classe* d'un langage de programmation et celle d'un RPG[4] et pareille entre les *objets* et les personnages. Une *classe* regroupe toute les caractéristiques qui définisse un *type*, de la même façon que dans un RPG, la classe "mage" définit ce qu'est un mage, quel types de magie il peut utiliser et quels sont ces caractérisques.

Et de la même façon que dans un groupe de personnages de RPG vous pouvez avoir plusieurs mages, étant différent l'un de l'autre ; dans votre code vous aurez plusieurs *objets* partageant la même *classe*

```
a = 1
b = 3
```

Ici, `a` et `b` ont tout les deux pour classe `int`. On dit que ce sont des *instences* de la *classe* `int`.

1.2.2 Stage 1-1

3. Ou du moins tout ce qui n'est pas une instruction
4. Role playing game (jeu de rôles)