

Inititation à la programmation

Bastien Gorissen & Thomas Stassin

Année 2016

Table des matières

1	Level 1	2
1.1	Qu'est-ce qu'un langage de programmation?	2
1.1.1	Stage 1	2
1.2	Commander à l'ordinateur	2
1.2.1	Stage 2	3
1.2.2	À retenir	3
1.3	Les instructions de sorties	4
1.3.1	Stage 3	4
1.4	Les variables	4
1.4.1	Stage 4	5
1.5	Opération sur une chaîne de caractère	5
1.5.1	Conversion	5

Chapitre 1

Level 1

1.1 Qu'est-ce qu'un langage de programmation ?

Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Ca, ce sont les caractéristiques de la magie.

Dave Small

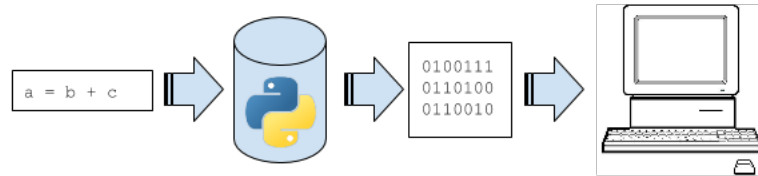
Un langage de programmation est un moyen d'interagir avec l'ordinateur afin de lui donner des instructions, le langage qui est *intelligible*, sera *interprété* afin d'être compris par la machine.

1.1.1 Stage 1

Executer le script se trouvant dans le dossier `lv1/run_game.py`

1.2 Commander à l'ordinateur

Pour “commander” l'ordinateur, on lui donne des *instructions*, le plus souvent, c'est instructions seront regroupées dans un *script*.



1.2.1 Stage 2

Dans l'exercice précédent, nous avons exécuté le script `run_game.py`, ce script est remplie d'instructions, dont celle-ci :

```
dungeon_size = (5, 5)
game = world.Game(dungeon_size)
game.run()
```

Que ce passerait t'il si on modifiait l'instruction qui définit la grandeur du donjon dans le script ? Changez les données de dimension du donjon dans le script et observez ce qu'il se passe lors de l'exécution du script.

1.2.2 À retenir

Python est un *langage sensible à la casse*¹, Ce qui veut dire qu'il fait la différence entre les majuscules et les minuscule. Autrement dit `A` sera différent de `a` et `world` différent de `World`

1. *Case sensitive* en anglais.

Donc si je remplace `world` par `World` dans le script précédent, il génère une erreur.

```
dungeon_size = (5, 5)
game = World.Game(dungeon_size)
game.run()
```

Vous devriez obtenir un message ressemblant à “`NameError: name 'World' is not defined`”

1.3 Les instructions de sorties

Il existe plusieurs sorte d’instruction, l’une d’elles sont les instructions de sortie. Une instruction de sortie envoie vers une “sortie” se qu’on lui donne. Une des sortie les plus couramment utilisé est la console et python, l’instruction de sortie vers la console est `print`

Et donc voici le classique, mais indémodable “Hello World” en python

```
print("Hello World!")
```

1.3.1 Stage 3

Lorsque le donjon est créé, signalez-le par un message dans la *console*.

1.4 Les variables

Souvent, il sera utile de stocké certaines valeur tout au long de l’exécution de votre code. Par exemple pour stocker la valeur d’un calcul, où même afin de réutiliser ses valeurs plusieurs fois. Pour stocker des valeurs, on utilise des *variables*.

La *variables* est un moyen de stocker une valeur quelconque (un nombre, du texte, voir même des objets plus complexes) dans la mémoire du programme.

Dans le cas de notre jeu, le donjon, ainsi que le héros sont contenu chacun dans une *variable*.

```
dungeon_size = (5, 5)
game = world.Game(dungeon_size)
hero = Game.hero
```

En python, l'*affectation* d'une valeur se fait avec l'opérateur =

Dans `a=3`, On affecte 3 à la variable `a`²

1.4.1 Stage 4

Faites en sorte de stocker les dimensions du donjon dans des variables (par exemple `longueur` et `largeur`).

1.5 Opération sur une chaîne de caractère

Il vous sera parfois utile de savoir manipuler des chaînes de caractères, l'exemple classique est dans une ligne de dialogue, On voudrait dire au héros le nombre d'ennemis qui lui reste à tuer avant de finir la quête, or lorsque que vous codez, vous ne connaissez pas le nombre d'ennemis. Vous avez sûrement stocké cette information dans une variable et donc vous allez devoir intégrer cette variable à votre ligne de dialogue.

1.5.1 Conversion

Il y a un moyen facile de convertir une variable en chaîne de caractères³, il suffit d'utiliser la fonction `str`.

Si je voulais convertir, la variable `nbr_enemies` qui contient le chiffre 3 je procéderaï comme suit :

```
nbr_enemies = 3
nbr_enemies = str(nbr_enemies)
```

A la fin de l'exécution de ce petit script, `nbr_enemies` ne vaut plus 3 mais "3".

2. En Python il suffit d'affecter une valeur à une variable pour qu'elle commence à exister, ce n'est pas vrai pour la plus part des langages (comme le C# par exemple).

3. En python les chaînes de caractère ont le *type* `str` pour le mot anglais *string*, qui veut dire chaîne

1.5.2 Concaténation