

Sommaire

1	La programmation avec Python 3	1
1.1	Introduction	1
1.1.1	C'est quoi un programme ??	1
1.1.2	Comment sont créés ces programmes ??	2
1.1.3	Les langages de programmation les plus connus	2
1.1.4	Compilé vs Interprété	3
1.1.5	Le langage Python	3
1.2	Les logiciels nécessaires pour la programmation	4
1.3	Les variables	5
1.4	les opérateurs arithmétique	5
1.5	Les conditions	5
1.6	Les boucles	5
1.7	Découper le programme en fonctions	5
1.8	TP	5
1.9	Le help	5
2	La programmation orientée objet	6
2.1	C'est quoi déjà un objet ?	6
2.2	Les différents types d'objets disponibles	6
2.3	Les opérations sur les fichiers	6
2.4	Les chaînes de caractères	6
2.5	Les listes	6
2.6	Les tuples	6
2.7	Les dictionnaires	6
2.8	TP	6
3	Les bibliothèques standards	6
3.1	Exécutions des commandes systèmes	6
3.2	L'aléatoire	6
3.3	Gestion des mots de passes	6
3.4	Le réseau	6
4	Allez plus loin	6

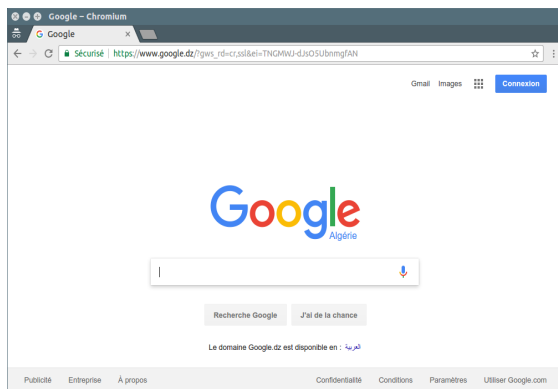
1 La programmation avec Python 3

1.1 Introduction

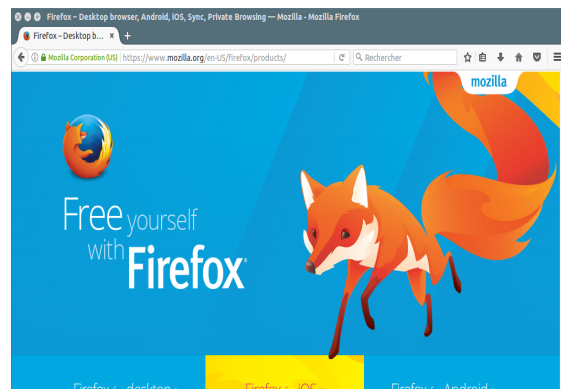
Ceci est une petite introduction où on va parler des programmes et des langages de programmations.

1.1.1 C'est quoi un programme ??

C'est un ensemble d'instructions exécuté par l'ordinateur pour qu'il puisse accomplir une tâche bien précise. On a pour exemple des programmes les navigateurs web (comme chromium, chrome, firefox), qui permettent de visiter les sites et de chatter en ligne ; on a aussi les jeux vidéos, les logiciels de retouches de photos ... etc



(a) Google Chrome



(b) Mozilla Firefox



FIGURE 2 – Le jeu Assassin's Creed

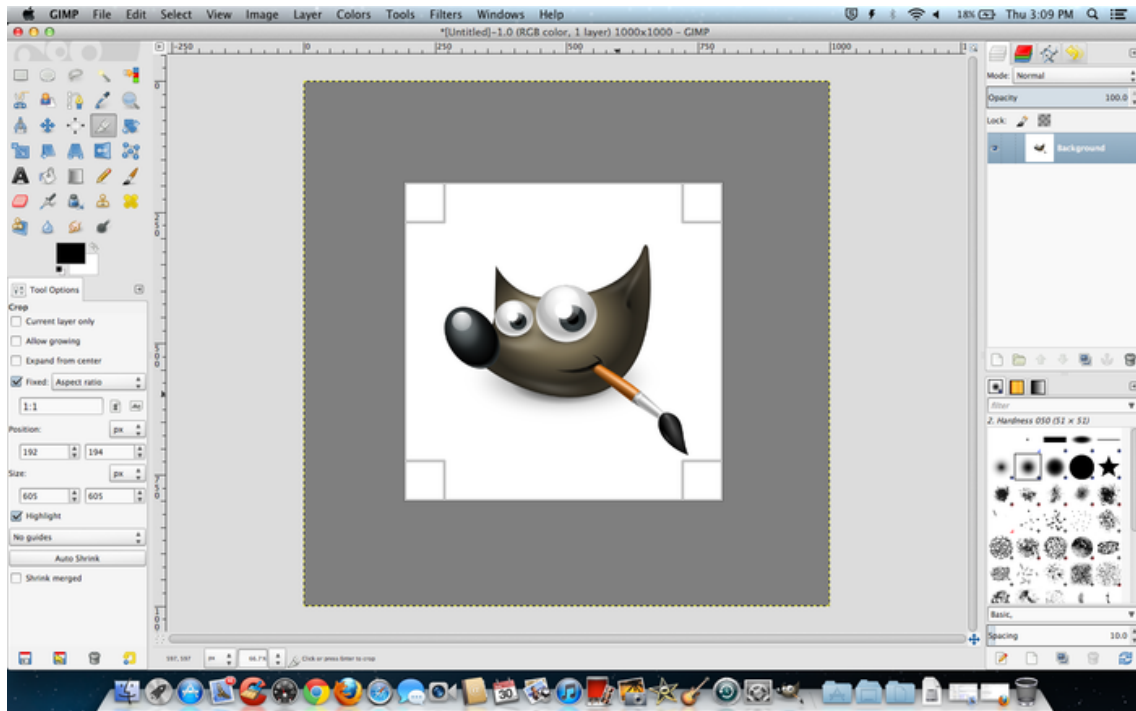


FIGURE 3 – Logiciels de retouches “Gimp”

1.1.2 Comment sont créés ces programmes ??

L'ordinateur est une machine bête et disciplinée qui ne comprend qu'un seul langage, qui est le langage *binaire*. Ce langage est une succession de 1 et de 0 comme suit :

```
011011100001100110011000111101011101
```

Donc, normalement, pour créer des programmes, il fallait apprendre ce langage. Mais heureusement, les informaticiens ont eu la réflexion de créer d'autres langages intermédiaires (qu'on appelle langage de programmation) qui sont beaucoup plus faciles que le binaire. Ces langages ont tous le même but : permettre de créer des programmes plus facilement qu'en binaire.

Voici comment cela fonctionne :

- On donne nos instructions à l'ordinateur en utilisant un langage de programmation.
- Les instructions sont traduites en binaire grâce à un programme de traduction.
- L'ordinateur peut alors lire le binaire et faire ce qu'on lui a demandé.

1.1.3 Les langages de programmation les plus connus

C++ : La formation est faite par YASSER.

Java : La formation est faite par WISSAM et SOFIANE.

Python : C'est la formation à laquelle vous assistez maintenant.



(a) Logo C++



(b) Logo Java



(c) Logo Python

1.1.4 Compilé vs Interprété

Nous avons dit tout à l'heure que les instructions sont traduites en binaire pour que l'ordinateur puisse les comprendre et les exécuter. Mais ce qu'on a pas dit c'est qu'il y a deux façons de traduire les instructions : *Compilation* et *Interprétation*.

Compilation : Avec ce processus, toutes nos instructions sont traduites en binaires à la fois. Ce qui nous donne en résultat un gros fichier binaire qui peut être exécuté directement par l'ordinateur.

Interprétation : Avec cette technique, on ne traduit pas tous les instructions en binaires, mais on va plutôt traduire une instruction par instruction à chaque fois qu'on veut exécuter notre programme. Par exemple, si mon code contient 3 instructions, quand je veux l'exécuter je traduit la première instruction en binaire, et je la passe à l'ordinateur pour qu'il l'exécute ; ensuite je traduit la deuxième instructions, et je la passe à l'ordinateur pour qu'il l'exécute ... et ainsi de suite jusqu'à la traduction de toute mes instructions.

Le programme qui fait l'interprétation est appelé *l'interpréteur*, et dans notre cas on va utiliser l'interpréteur *Python*.

1.1.5 Le langage Python

Python est un langage de programmation interprété, facile à apprendre, et très pratique. On cite quelques avantages de ce langage :

- Python est facile à apprendre et son code est facile à lire.
- Il multiplate-forme, il marche sous Linux, Windows, et Mac OS.
- C'est un langage orienté objet (on en parlera de ça dans le prochaines séances).
- Il a une bibliothèque tierce qui permet de faire de la programmation web d'une manière très flexible. Cette bibliothèque c'est *Django*.

- Il permet de développer très rapidement en utilisant peu de code, et de créer des applications très complexes avec facilité. Tout ça grâce à son style et à sa bibliothèque standard très complète.

Pour finir, il faut savoir que les géants de l'informatique comme Google, Yahoo, NASA préfèrent le langage python ; et ce langage est intégré à la quasi totalité des distributions Linux (C'est-à-dire que vous pouvez l'utiliser dans ces distributions sans devoir l'installer).

Juste pour information, il en existe deux versions de python qui ne sont pas compatibles entre eux. Ces deux versions sont les 2.x et les 3.x . Nous on va utiliser la version 3.x qui est le futur de python.

1.2 Les logiciels nécessaires pour la programmation

Pour pouvoir créer des programmes en python, il existe deux manières :

1. Soit on écrit nos instructions dans un fichier texte, et puis on appelle l'interpréteur python pour qu'il exécute nos instructions et nous affiche le résultat. Cette méthode comporte de nombreux avantages, et elle est très flexible, mais elle est compliquée (surtout pour les débutants). Donc nous allons opter pour la deuxième méthode dans cette formation.
2. Cette méthode consiste à utiliser un programme appelé *IDE* (Integrated Development Environment), ou *EDI* en français (Environnement de Développement Intégré). Ce programme contient tout ce dont on a besoin pour créer nos programmes : une partie où on peut écrire nos instructions, des boutons pour demander à python d'exécuter nos instructions, et finalement une partie où le résultat de l'exécution est affiché.

L'IDE qu'on va utiliser est appelé *PyCharm* (Figure 5). C'est un IDE *Open Source* et très puissant.

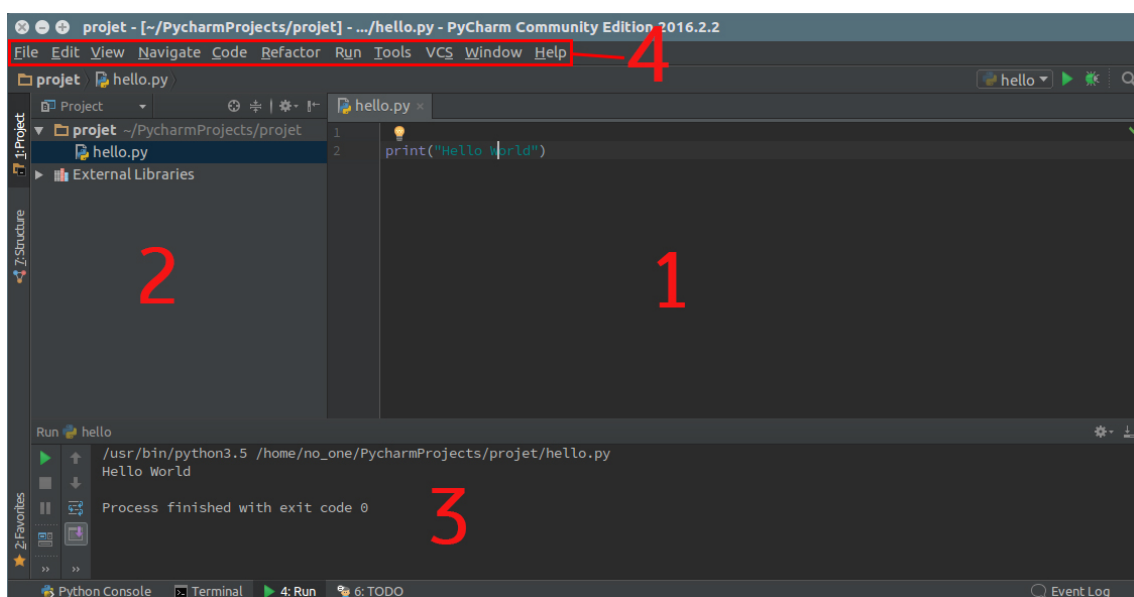


FIGURE 5 – Pycharm

PyCharm est composé essentiellement de 4 partie :

La zone principale (1) : c'est dans cette zone où on écrira nos instructions (code).

La liste des fichiers du projet (2) : dans cette zone on peu voir tous les fichiers de notre projet. On voit bien que le projet présenté dans la [Figure 5](#) contient un seul fichier seulement.

La zone d'affichage (3) : c'est cette zone qui nous affichera le résultat d'exécution de notre programme, où les éventuelles erreurs s'il y en a.

La barre d'outils (4) : elle contient de nombreux boutons qui font différentes choses : exécuter le projet, configurer l'IDE ... etc, mais nous on utilisera que quelques boutons.

1.3 Les variables

Affichage, lecture des variables.

1.4 les opérateurs arithmétique

1.5 Les conditions

Explication des structures avec un exemple après chaque petite explication.

1.6 Les boucles

Explication des structures avec un exemple après chaque petite explication.

1.7 Découper le programme en fonctions

C'est quoi une fonction. Quelles sont les fonctions que nous avons déjà utilisés. Comment créer nos propres fonctions.

1.8 TP

1.9 Le help

Obtenir le help depuis la console.

2 La programmation orientée objet

2.1 C'est quoi déjà un objet ?

2.2 Les différents types d'objets disponible

On parlera des string, listes, tuples et des dictionnaires.

2.3 Les opérations sur les fichiers

Ouverture, lecture, écriture des fichiers.

2.4 Les chaînes de caractères

2.5 Les listes

2.6 Les tuples

2.7 Les dictionnaires

2.8 TP

3 Les bibliothèques standards

3.1 Exécutions des commandes systèmes

3.2 L'aléatoire

3.3 Gestion des mots de passes

3.4 Le réseau

4 Allez plus loin

Des tutos pour la programmation graphique avec PyQt. Le tuto d'openclass-rooms.