

PYTHON - Introduction

Table des matières

1) Un ordinateur, c'est quoi ?	2
2) Système d'exploitation	3
3) Un langage de programmation c'est quoi ?	4
Première case : « Votre programme est écrit dans un langage simplifié »	4
Deuxième case : « programme de traduction »	4
Troisième case : « programme binaire »	5
En conclusion	5
4) Programmeur.....	5
Qu'est-ce qu'un programmeur ?.....	5
Quelles qualités doit posséder un programmeur ?	5
La démarche du programmeur	6
Mise au point d'un programme - Recherche des erreurs.....	6
Où peut travailler un programmeur ?.....	7
5) Introduction à python.....	8
Histoire	8
À quoi peut servir Python ?.....	9
Un langage de programmation interprété.....	9
Principales caractéristiques du langage.....	10
6) Installation python sous windows (pour les PC personnels).....	10
7) Moodle et sauvegarde du travail.....	11

1) Un ordinateur, c'est quoi ?

–1 – Carte mère

- mise en relation des composants par des bus sous forme de circuits imprimés.
- comporte des emplacements (ou « slots »), prévus pour accepter différents types de composants

–2 – CPU (processeur)

- cerveau de l'ordinateur où sont effectués les opérations (calculs) demandés.
- Sa vitesse (fréquence) de calcul est exprimée en Htz (de nos jours les processeurs dépassent les 3 gHtz).

–3 – Bus

- système de communication entre les composants d'un ordinateur.
- permet(tent) de connecter les différentes parties fonctionnelles d'un ordinateur entre elles.

–4 – RAM – Mémoire vive

- stocke les informations des programmes et données en cours de fonctionnement.
- La capacité de stockage de la mémoire s'exprime en octet (ou Byte), aujourd'hui la Ram d'un ordinateur peut atteindre plusieurs Gigas Bytes (GB).

–5 – Carte graphique (carte vidéo)

- permet de produire une image pouvant être affichée sur un écran.
- envoie à l'écran des images stockées dans sa mémoire (Ram), à une fréquence et dans un format qui dépendent d'une part de l'écran branché et du port sur lequel il est branché (grâce au Plug and Play).

–6 – Entrée / Sortie

- dispositifs qui permettent de communiquer avec le monde extérieur (Usb, port série, ...).

–7 – Disque Dur

- stocke les informations des programmes et données de base de la machine.
- La capacité de stockage d'un disque dur s'exprime en octet (ou byte), aujourd'hui la capacité de stockage peut atteindre plusieurs Terra Bytes (Tb).

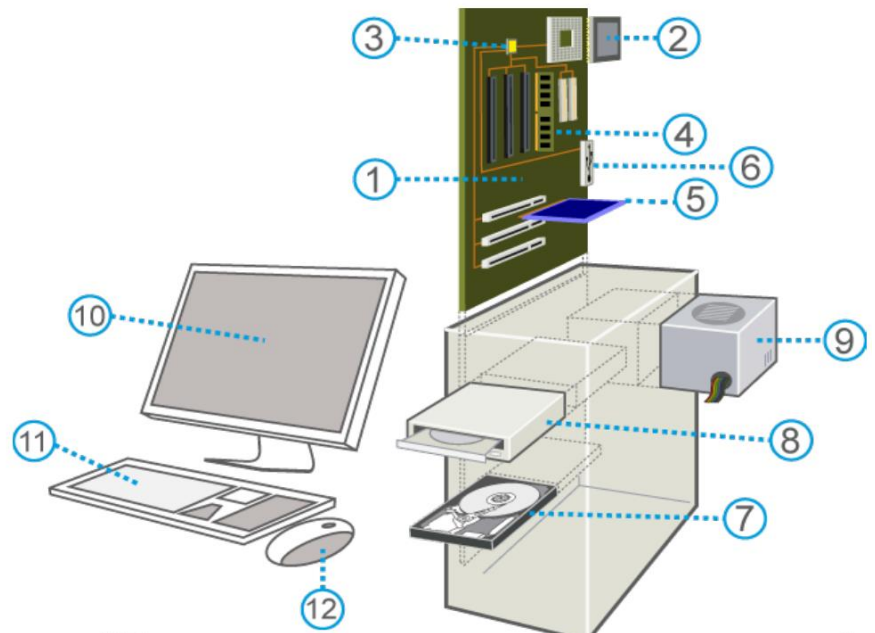
–8 – Lecteur CD

–9 - Alimentation

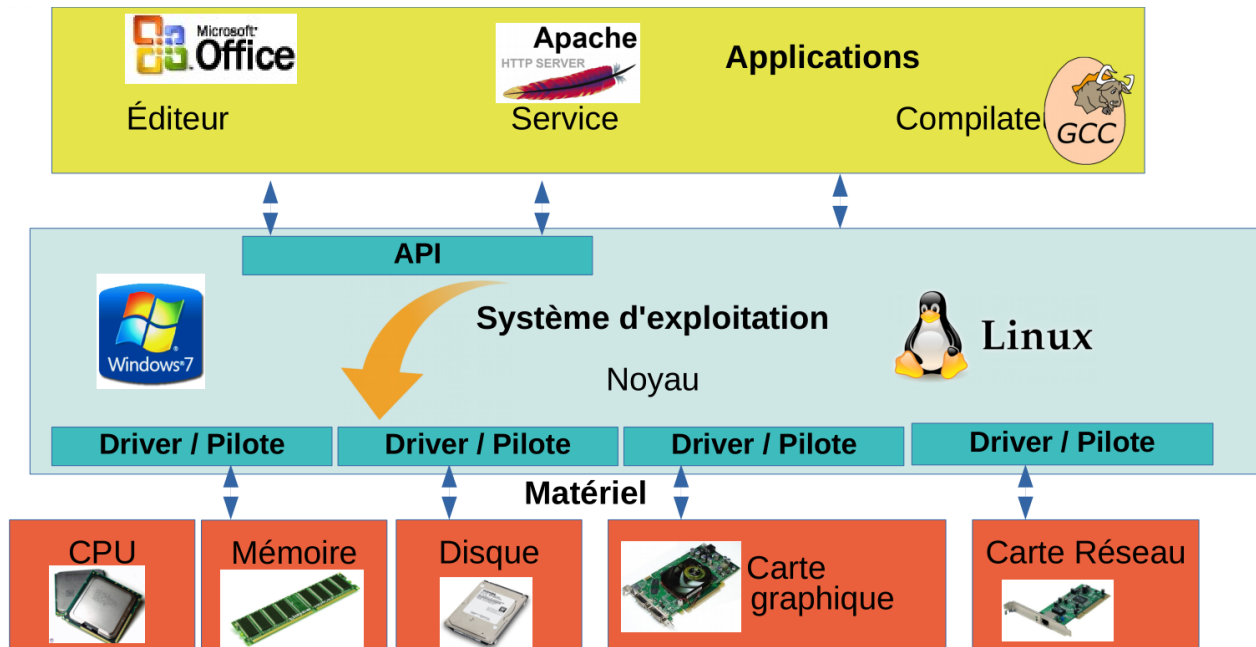
–10 – Ecran

–11 – Clavier

–12 – Souris



2) Système d'exploitation



noté SE ou OS, abréviation du terme anglais Operating System

- Un programme qui agit comme un intermédiaire entre un utilisateur d'un ordinateur et le matériel
- La fonction du système d'exploitation est de masquer la complexité du matériel et de proposer des instructions plus simples à l'utilisateur.
- Le système d'exploitation est un gestionnaire de ressources, c'est-à-dire qu'il contrôle l'accès à toutes les ressources de la machine, l'attribution de ces ressources aux différents utilisateurs et la libération de ces ressources lorsqu'elles ne sont plus utilisées.
- À ce titre, tous les périphériques comme la mémoire, le disque dur ou les imprimantes sont des ressources. Le processeur également est une ressource.

3) Un langage de programmation c'est quoi ?



Première case : « Votre programme est écrit dans un langage simplifié ».

Ce « langage simplifié » est appelé en fait **langage de haut niveau**. Il existe plusieurs niveaux de langages : **plus un langage est haut niveau, plus il est proche de votre vraie langue** (le français)

Il existe de nombreux langages de plus ou moins haut niveau en informatique dans lesquels vous pouvez écrire vos programmes :

- le C ;
- le C++ ;
- Java ;
- Python ...

Le programme tel que nous l'écrivons dans un langage de programmation quelconque **est** à strictement parler **un simple texte**.

Pour rédiger ce texte, on peut faire appel à toutes sortes de logiciels plus ou moins perfectionnés, à la condition qu'ils ne produisent que du texte brut, c'est-à-dire sans mise en page particulière ni aucun attribut de style (pas de spécification de police, donc, pas de gros titres, pas de gras, ni de souligné, ni d'italique, etc.)

Le texte ainsi produit est ce que nous appellerons désormais **un « code source »**.

Le code source est donc le code de votre programme écrit dans un langage de haut niveau

Deuxième case : « programme de traduction »

Ce programme **traduit notre langage de haut niveau en binaire**.

2 types de traduction :

- Langage compilé :

langage qui est traduit en langage machine (binaire) par un **compilateur** avant de pouvoir s'exécuter. exemple : C, C++, JAVA , ...

- Langage interprété

Ce langage est lui **traduit au fur et à mesure** de l'exécution **par l'interpréteur**

exemple : shell, Python, ...

Troisième case : « programme binaire »

C'est le **langage informatique de votre ordinateur**, il ne connaît que ce langage-là.

le programme binaire est appelé **l'exécutable**.

C'est d'ailleurs pour cette raison que les programmes (tout du moins sous Windows) ont l'extension « .exe » comme EXEcutable

En conclusion

La programmation est l'art d'**apprendre à une machine** comment elle pourra **accomplir des tâches nouvelles**, qu'elle n'avait jamais été capable d'effectuer auparavant.

C'est par la programmation que vous pourrez acquérir le plus de contrôle, non seulement sur votre machine, mais aussi peut-être sur celles des autres, par l'intermédiaire des réseaux.

4) Programmeur

Qu'est-ce qu'un programmeur ?

Dans le monde de la programmation, on désigne par le terme de programmeurs les personnes qui

- perfectionnent les systèmes d'exploitation de type Unix
- produisent et améliorent des logiciels/applications à l'aide de langage de programmation

Quelles qualités doit posséder un programmeur ?

- Il veille à ce que ses constructions logiques soient efficaces
- Il veille à que ses constructions aient une structure parfaitement lisible et documentée
Vous découvrirez rapidement qu'il est aisé de produire à la va-vite des programmes d'ordinateur qui fonctionnent, certes, mais qui sont obscurs et confus, indéchiffrables pour toute autre personne que leur auteur ...
- Il s'adapte facilement à de nouveaux langages
- Il fait preuve d'autonomie
 - consulter et/ou participer aux forums informatique (trouver solutions à ses erreurs, veille sur les nouveautés, ...)
 - auto-formation (tutoriels, Moocs, ...)

La démarche du programmeur

L'activité essentielle d'un programmeur consiste à résoudre des problèmes.

Il s'agit-là d'une compétence de haut niveau, qui implique des capacités et des connaissances diverses :

- être capable de (re)formuler un problème de plusieurs manières différentes,
- être capable d'imaginer des solutions innovantes et efficaces,
- être capable d'exprimer ces solutions de manière claire et complète.

La programmation d'un ordinateur consiste en effet à « expliquer » en détail à une machine ce qu'elle doit faire, en sachant d'emblée qu'elle ne peut pas véritablement « comprendre » un langage humain, mais seulement effectuer un traitement automatique sur des séquences de caractères.

Il s'agit la plupart du temps de **convertir un souhait exprimé à l'origine en termes français, en un vrai raisonnement parfaitement structuré et élucidé dans ses moindres détails, que l'on appelle un algorithme.**

Mise au point d'un programme - Recherche des erreurs

les erreurs de programmation s'appellent des « bugs » (ou « bogues », en Français)

l'ensemble des techniques que l'on met en œuvre pour les détecter et les corriger s'appelle « debug » (ou « débogage »).

En fait, il peut exister dans un programme trois types d'erreurs assez différentes :

- Erreurs de syntaxe
- Erreurs sémantiques
- Erreurs à l'exécution

Erreurs de syntaxe

Python ne peut exécuter un programme que si sa syntaxe est parfaitement correcte.

Dans le cas contraire, le processus s'arrête et vous obtenez un message d'erreur.

Le terme **syntaxe** se réfère aux règles que les auteurs du langage ont établies pour la structure du programme.

Tout langage comporte sa syntaxe.

Dans la langue française, par exemple, une phrase doit toujours commencer par une majuscule et se terminer par un point.

Dans un programme d'ordinateur, **la moindre erreur de syntaxe produit invariablement un arrêt de fonctionnement (un « plantage ») ainsi que l'affichage d'un message d'erreur.**

Erreurs sémantiques

Le second type d'erreur est l'erreur sémantique ou erreur de logique.

S'il existe une erreur de ce type dans un de vos programmes, celui-ci s'exécute parfaitement, en ce sens que **vous n'obtenez aucun message d'erreur, mais le résultat n'est pas celui que vous attendiez** : vous obtenez autre chose.

En réalité, le programme fait exactement ce que vous lui avez dit de faire.

Le problème est que ce que vous lui avez dit de faire ne correspond pas à ce que vous vouliez qu'il fasse.

La séquence d'instructions de votre programme ne correspond pas à l'objectif poursuivi.

La sémantique (la logique) est incorrecte.

Il vous faudra analyser patiemment ce qui sort de la machine et tâcher de vous représenter une par une les opérations qu'elle a effectuées, à la suite de chaque instruction.

Erreurs à l'exécution

Le troisième type d'erreur est l'erreur en cours d'exécution (Run-time error), qui apparaît seulement lorsque votre programme fonctionne déjà, mais que **des circonstances particulières se présentent**.
par exemple, votre programme essaie de lire un fichier qui n'existe plus.

Ces erreurs sont également appelées des exceptions, parce qu'elles indiquent généralement que quelque chose d'exceptionnel (et de malencontreux) s'est produit.

Vous rencontrerez davantage ce type d'erreurs lorsque vous programmerez des projets de plus en plus volumineux.

Où peut travailler un programmeur ?

Un programmeur peut trouver de nombreux endroits et façons de travailler

Il peut être employé par :

- Un éditeur de logiciel
- Une entreprise de services du numérique (ESN), appelé anciennement SSII
- Une entreprise dont le cœur de métier n'est pas l'informatique mais qui possède en interne un service informatique (SI)

Il peut également travailler à son compte, en tant que travailleur indépendant.

En effet, le métier permet facilement le travail à distance (télétravail)

5) Introduction à python

Il existe un très grand nombre de langages de programmation, chacun avec ses avantages et ses inconvénients. Il faut en choisir un.

Il nous faut un langage qui ne soit **pas lié à des environnements de programmation** (c'est-à-dire des logiciels) **propriétaires**.

Sinon cela signifie, non seulement que l'institution scolaire désireuse de l'utiliser devrait acheter une licence de ces logiciels pour chaque poste de travail (ce qui pouvait se révéler coûteux), mais surtout que les élèves souhaitant utiliser leurs compétences de programmation ailleurs qu'à l'école seraient implicitement forcés d'acquiescer eux aussi des licences, ce que nous ne pouvons pas accepter.

Il nous faut un langage qui **ne comporte pas de nombreuses « boîtes noires »** dont on ne peut connaître le contenu.

Il nous faut un langage qui **ne soit pas spécifiquement lié au seul système d'exploitation Windows**. Sinon le langage n'est pas « portable » sur d'autres systèmes (Unix, Mac OS, etc.). Cela ne cadrerait pas avec notre projet pédagogique qui ambitionne d'inculquer une formation générale et donc diversifiée !

Python répond à toutes ces exigences

Histoire

À la fin des années 1980, le programmeur **Guido van Rossum** participe au développement du langage de programmation ABC à Amsterdam, aux **Pays-Bas**.

Il estime alors qu'un langage de script inspiré d'ABC pourrait être intéressant.

En 1989, profitant d'une semaine de vacances durant les fêtes de Noël, il utilise son ordinateur personnel pour écrire la première version du langage.

Fan de la série télévisée *Monty Python*, il décide de baptiser ce projet Python.

Il s'est principalement inspiré de :

- **ABC**, par exemple pour l'indentation comme syntaxe ou les types de haut niveau
- **Modula-3** pour la gestion des exceptions
- **langage C**
- **outils UNIX**

Durant l'année suivante, le langage commence à être adopté par son équipe de projet, Guido poursuivant son développement principalement pendant son temps libre.

Depuis, Python est devenu un langage de programmation généraliste.



Python est un langage open source supporté, développé et utilisé par une large communauté : 300 000 utilisateurs et plus de 500 000 téléchargements par an.

Tout code, documentation et spécification ajouté, est **détenu** par la Python Software Foundation (PSF), une **association** sans but lucratif **fondée en 2001**.

La nouvelle version majeure est **Python 3.0**, publié en décembre **2008**. Cette version a été suivie rapidement par une **version 3.1** qui corrige les erreurs de la version 3.0.

À quoi peut servir Python ?

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités.

Dès l'instant où vous l'installez sur votre ordinateur, vous disposez de nombreuses **fonctionnalités intégrées au langage** que nous allons découvrir.

Et il est très facile **d'étendre les fonctionnalités existantes**.

Ainsi, il existe ce qu'on appelle des **bibliothèques** qui aident le développeur à travailler sur des projets particuliers.

voici ce qu'on peut faire avec Python :

- de petits programmes **simples**, appelés **scripts**, chargés d'une mission très précise sur votre ordinateur ;
- des programmes **complets**, comme des jeux, des suites bureautiques, des logiciels multimédias, des clients de messagerie...
- des projets très **complexes**, comme des progiciels (ensemble de plusieurs logiciels pouvant fonctionner ensemble, principalement utilisés dans le monde professionnel).

Un langage de programmation interprété

Python est un langage de programmation **interprété**, c'est-à-dire que les instructions que vous lui envoyez sont « **transcrites** » en **langage machine au fur et à mesure** de leur lecture.

D'autres langages (comme le C / C++) sont appelés « langages **compilés** » car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine.

On appelle cette étape la « **compilation** ».

À chaque modification du code, il faut rappeler une étape de compilation.

Les **avantages** d'un **langage interprété** sont

- la simplicité : on ne passe pas par une étape de compilation avant d'exécuter son programme
- la portabilité : Python est fonctionne aussi bien sous Windows que sous Linux ou Mac OS, et on effectue aucun changement dans le code pour le passer d'un système à l'autre.

Les **inconvenients** d'un **langage interprété** sont :

- un langage interprété se révélera **moins rapide** qu'un langage compilé car la traduction à la volée de votre programme ralentit l'exécution
- il faudra **installer** Python sur le **système d'exploitation** que vous utilisez pour que l'ordinateur puisse comprendre votre code.

Principales caractéristiques du langage



Python est **orienté-objet**.

Python intègre, comme Java ou les versions récentes de C++, un système **d'exceptions**, qui permettent de simplifier considérablement la gestion des erreurs.

Python est **introspectif** : un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même.

Python est **dynamiquement typé**. Tout objet manipulable par le programmeur possède un type bien défini à l'exécution, qui n'a pas besoin d'être déclaré à l'avance.

Python est **extensible** : on peut facilement l'interfacer avec des bibliothèques C existantes. On peut aussi s'en servir comme d'un langage d'extension pour des systèmes logiciels complexes.

La **bibliothèque standard** de Python, et les paquetages contribuéés, donnent accès à une grande variété de services : chaînes de caractères et expressions régulières, services UNIX standards (fichiers, pipes, signaux, sockets, threads...), protocoles Internet (Web, News, FTP, CGI, HTML...), persistance et bases de données, interfaces graphiques.

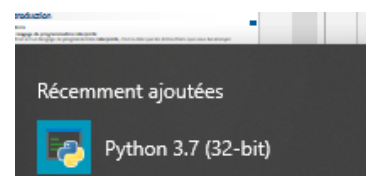
Python est un langage qui **continue à évoluer**, soutenu par une communauté d'utilisateurs enthousiastes et responsables, dont la plupart sont des supporters du logiciel libre.

6) Installation python sous windows (pour les PC personnels)

Pour l'installation de Python, quel que soit votre système d'exploitation, vous devez vous rendre sur le **site officiel de Python** : <https://www.python.org/>

Sous Windows

1. Cliquez sur le lien Download dans le menu principal de la page.
2. Sélectionnez la version de Python que vous souhaitez utiliser (je vous conseille la dernière en date).
3. Enregistrez puis exécutez le fichier d'installation et suivez les étapes.
4. Une fois l'installation terminée, vous pouvez vous rendre dans le menu Démarrer>Tous les programmes.
Python devrait apparaître dans cette liste



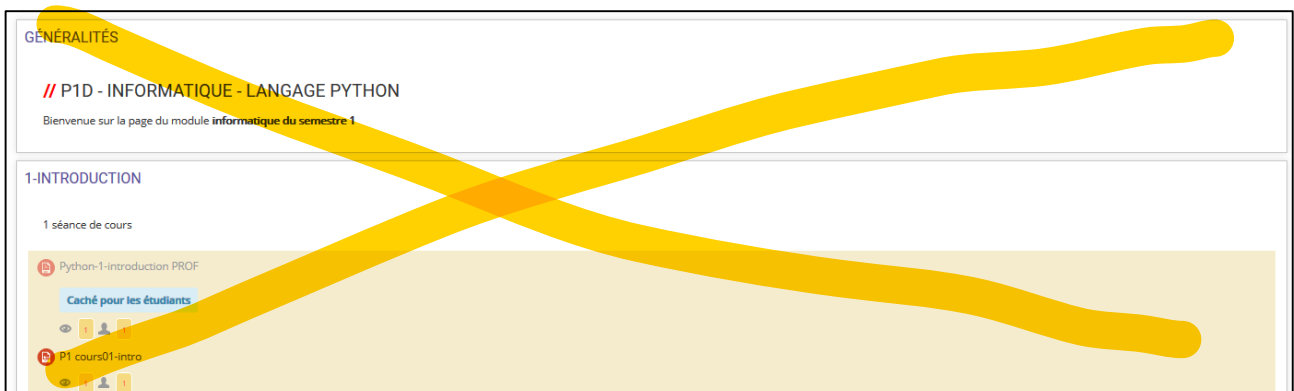
7) Moodle et sauvegarde du travail

Le Moodle se trouve à l'adresse : <https://campus.eseo.fr>



Sur cette page vous trouverez :

- Les cours en version PDF
- Les énoncés des exercices
- Les exercices corrigés (disponible en fin de séance)
- Les énoncés des évaluations
- Le dépôt des évaluations (date limite à respecter)



Veillez à organiser votre travail ! Et veillez à choisir un dossier de sauvegarde que vous pourrez retrouver d'une séance à l'autre ! (Cloud, clé USB, etc...) !

Exemple :

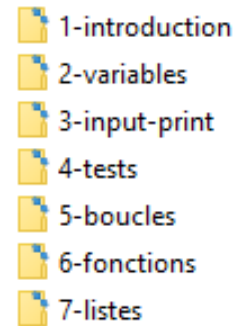
Vous pouvez créer un dossier par chapitre,
puis sauvegarder tous les documents ou exercices

qui concernent ce chapitre dans ce dossier

Z:/Python/1-**introduction**/Python-1-introduction.docx

Z:/Python/2-**variables**/exercice1.py

...



Voici une proposition d'organisation
pour le 1^{er} semestre