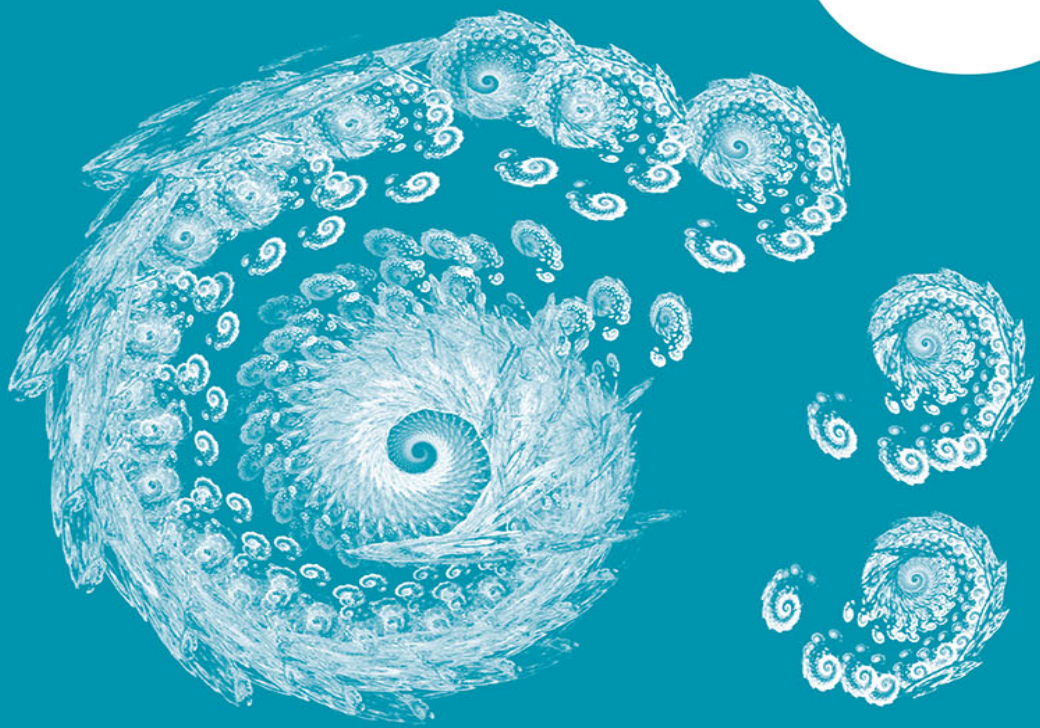
Introduction à **l'informatique**

Serge **Bays**

1re année **prépa scientifique**



ellipses

Introduction à **l'informatique**

Introduction à **l'informatique**

Serge **Bays**

1re année
prépa scientifique

ellipses

ISBN 9782340-049918

32, rue Bargue 75740 Paris cedex 15

www.editions-ellipses.fr

Avant-propos

L'informatique est enseignée en classes préparatoires depuis la rentrée 2013. Elle est présente depuis quelques années déjà dans la vie professionnelle des ingénieurs, des enseignants, des chercheurs, mais aussi dans la vie publique de la plupart des gens. Son rôle est semblable à celui des mathématiques avec un développement propre et des applications à toutes les disciplines scientifiques.

Les notions centrales sont : Algorithme, Information, Langage et Machine. Ces concepts existaient déjà avant la naissance de l'informatique mais ils sont toujours au cœur du fonctionnement des ordinateurs même si ceux-ci contiennent toujours plus de données et travaillent toujours plus vite.

Cet ouvrage est destiné principalement aux étudiants en première année de classes préparatoires aux Grandes Ecoles. Son ambition est de les accompagner tout au long de l'année et de leur permettre un travail en autonomie. Il se veut le plus complet possible tout en restant dans les limites du programme d'informatique commun aux différentes voies de la filière scientifique.

Le langage de programmation utilisé, Python version 3, est très riche mais il a été choisi de n'utiliser que le strict minimum permettant de répondre aux attentes du programme. En phase d'apprentissage, il est important de faire, de construire, de créer, à partir d'éléments simples, ses propres outils. Les bibliothèques NumPy et SciPy viennent en complément à la bibliothèque standard et seulement lorsque leur apport est nécessaire ou présente un intérêt évident ; la bibliothèque Matplotlib est utilisée pour les graphiques.

Le contenu de l'ouvrage correspond à l'enseignement donné en classe depuis la rentrée 2013. Il ressort de cette expérience avec les étudiants qu'il est absolument nécessaire d'insister sur les bases de la programmation qui sont : les notions de variable, de fonction, de test et de boucle.

L'ordre des chapitres, qu'il est conseillé de suivre pour un travail fructueux, est à l'image de la progression suivie en classe avec les étudiants.

Les chapitres 1 à 8 occupent un peu plus d'un semestre et donnent tous les

outils indispensables, en particulier des algorithmes de base, dont certains sont déjà étudiés en terminale S, et qui doivent absolument être maîtrisés pour la suite, ainsi que leur programmation en Python.

Les chapitres 9, 10 et 11 traitent de la partie ingénierie numérique et simulation ; c'est le moment d'investir toutes les connaissances acquises.

Les chapitres 12 et 13 proposent une initiation aux bases de données relationnelles et au langage SQL.

Chacun de ces chapitres est composé d'une partie cours suivie par de nombreux exercices variés et progressifs qui sont tous corrigés. Très souvent un corrigé est donné sous la forme d'un programme et il est bien entendu qu'il s'agit d'une réponse parmi plusieurs envisageables et aussi valables les unes que les autres.

Deux chapitres concluent ce livre avec les principaux outils des bibliothèques NumPy, Matplotlib, SciPy et un résumé condensé du langage Python.

Les logiciels utilisés sont gratuits et disponibles en téléchargement. Ceci permet aux étudiants de travailler en dehors de la classe. Une pratique régulière est absolument nécessaire pour acquérir de bons réflexes.

- Programmation : l'environnement de développement choisi est "Idle" qui rassemble les tâches courantes d'un éditeur de texte, l'interprétation d'un fichier source et l'exécution d'un programme.
- Ingénierie numérique : langage Python avec les bibliothèques complémentaires NumPy, Matplotlib et SciPy. Ces trois bibliothèques doivent être installées.
  Le plus pratique peut être de choisir une version "portable" comme WinPython qui contient tout le nécessaire et dont l'installation est très simple.
- Base de données : PHPMyAdmin pour SQL ou Python avec sqlite.

Je remercie les collègues avec qui j'ai eu la chance de partager cet enseignement pour les nombreux et fructueux échanges ainsi que tous les élèves qui l'ont expérimenté. Sans eux ce livre ne serait pas le même.

Merci à mon épouse et mes deux filles pour leur soutien et leur patience.

Table des matières

Chapitre 1

Architecture

1.1 Introduction

Un système informatique se compose de deux parties :

- une partie matérielle (hardware en anglais) qui représente l'ensemble des composants de la machine,
- une partie logicielle (software en anglais) constituée des logiciels s'exécutant sur le matériel.

Une bonne connaissance du fonctionnement interne de l'ordinateur et des caractéristiques du matériel permet de comprendre pourquoi certains algorithmes se révèlent efficaces alors que d'autres le sont moins, par rapport à une architecture donnée, et comment en améliorer le fonctionnement.

1.1.1 Définition : ordinateur

Un ordinateur est une machine électronique qui effectue des calculs et traite des informations de manière automatique.

Le terme ordinateur a été proposé pour la première fois par Jacques Perret, philologue français, à la demande d'IBM France en 1955, afin de traduire le terme anglais computer.

Un ordinateur est composé de plusieurs parties appelées :

- composants (carte mère, microprocesseur, barrette de mémoire, carte graphique, ...) ;
- périphériques (disque dur, lecteur de DVD, clavier, souris, moniteur, ...).

Un périphérique est éloigné de la carte mère alors qu'un composant est en contact direct avec elle.

Pour certains, le terme périphérique fait plutôt référence à tout ce qui est externe au boîtier : clavier, souris, moniteur, imprimante, ..., bref ce qui se trouve à la périphérie.

1.1.2 Définition : informatique

C'est la science de la recherche et du traitement de l'information effectués de manière automatique par une machine (un ordinateur) à l'aide de logiciels ou programmes.

Le terme informatique a été traduit de l'allemand "informatik" en 1962 par Philippe Dreyfus (Directeur du centre national de calcul électronique de la société Bull dans les années 1950) à partir des mots information et automatique.

Les anglophones emploient les termes Computer Science ou Computer Engineering.

1.2 Histoire

1.2.1 Première génération

Les premiers ordinateurs datent des années 1940 et étaient utilisés par l'armée, par exemple L'ENIAC (Electronic Numerical Integrator Analyser and Computer) aux Etats-Unis, qui pesait 30 tonnes, ou bien la série Z1, Z2 et Z3, en Allemagne. Leur technologie était basée sur des tubes électroniques qui prenaient une place importante et dégageaient beaucoup de chaleur. De plus ils coûtaient très cher. Leur puissance de calcul (unité : le Flop, floating point operation per seconde) était comparable à celle d'une petite calculette actuelle.

En 1951, Grace Hopper (1906-1992), informaticienne américaine, conçoit le premier compilateur. Elle est à l'origine de l'expression "bug informatique".

1.2.2 Deuxième génération

Vers la fin des années 1950, les tubes sont remplacés par des transistors avec un gain en puissance de calcul. La consommation électrique, la taille et le prix sont réduits. Les ordinateurs entrent dans les universités.

1.2.3 Troisième génération

Dans les années 1960, le circuit intégré est inventé, (puce en français, chip en anglais). Un circuit remplace de très nombreux tubes ou transistors. La NASA (National Aeronautics and Space Administration) va pouvoir embarquer un ordinateur pour aller sur la Lune.

1.2.4 Quatrième génération

A partir de 1971, le cœur de l'ordinateur est un ensemble de circuits intégrés appelé processeur. Actuellement, la puissance de calcul d'un ordinateur personnel est d'environ 100 Gigaflops, (Giga = milliard). La démocratisation de l'informatique a suivi l'apparition des micro-ordinateurs.

1.3 Architecture matérielle

1.3.1 Architecture de Von Neumann

L'architecture des ordinateurs actuels repose sur le modèle de Von Neumann.

John Von Neumann (1903-1957) était un mathématicien américain d'origine hongroise. Il travailla comme consultant dans le projet ENIAC. Selon lui la mémoire de l'ordinateur, qui servait à stocker des données, devait également stocker les programmes : c'est le concept de programme enregistré.

L'organisation est la suivante :

- une mémoire
- une unité de calculs CA (Central Arithmetical part) ou ALU (Arithmetic and Logic Unit)
- une unité de contrôle CC (Central Control device)
- des entrées/sorties
- une horloge

La mémoire stocke des nombres et des instructions sur 32 bits ou 64 bits pour les ordinateurs les plus récents.

Le processeur (CA + CC) communique avec la mémoire et les entrées/sorties par des "bus".

L'horloge est un circuit qui émet un signal périodique afin de synchroniser les circuits qui en ont besoin (en particulier les circuits mémoires).

1.3.2 Une machine

- Le boîtier contient l'ensemble des composants ; c'est un élément important pour les raisons suivantes :
  - les composants électroniques dégagent de la chaleur qui doit être évacuée pour éviter les risques de surchauffe ;
  - certains composants comme les disques durs ou les ventilateurs font du bruit et une bonne isolation phonique n'est pas négligeable.

- Un bloc d'alimentation convertissant le courant alternatif 220 V en courant continu 12 V et 5V. (Une batterie rechargeable sur les ordinateurs portables).

- L'unité centrale :
  - Carte mère
  - Micro-processeur
  - Mémoire
  - Périphériques internes : disque dur, lecteur DVD, carte graphique, carte réseau, ...
  - Ports de communication

- Les périphériques externes d'entrée/sortie :

– Moniteur
– Clavier
– Souris
– Enceintes
– Imprimante
– Graveur externe

Carte mère

La carte mère est un circuit imprimé qui permet de mettre en contact physique les composants et les périphériques. Les différents composants de la machine sont reliés par des canaux de communication, appelés bus, permettant d'échanger l'information. La carte mère détermine la vitesse des différents bus.

Sur une carte mère se trouvent :

– le socket qui est le support sur lequel se connecte le processeur et qui détermine son type ;
– des connecteurs pour la mémoire qui déterminent le type de mémoire à utiliser ainsi que la taille de la mémoire maximale ;
– différents ports :
  - PCI (Peripheral Component Interconnect) pour les cartes d'extension graphique, son, réseau ;
  - AGP (Accelerated Graphics Port) pour les cartes graphiques hautes performances ;
  - IDE (Integrated Device Electronics) pour les périphériques internes, les disques durs, les lecteurs/graveurs de CD/DVD ;
  - USB (Universal Serial Bus) destiné à remplacer et uniformiser les différentes connexions comme les claviers, souris, imprimantes qui utilisent les ports parallèle, série, ... ;
  - IEEE 1394 (FireWire ou i.Link) pour des périphériques à très haut débit comme les caméras numériques.

Notion de bus

Un bus se décompose en 3 parties :

- le bus d'adresses pour spécifier l'adresse mémoire visée ;
- le bus de données pour envoyer ou recevoir une donnée ;
- le bus de commandes pour spécifier si c'est une lecture ou une écriture qui doit s'effectuer.

Sa largeur, (en nombre de bits ou d'octets) indique le nombre de bits qui sont transférés en même temps et sa fréquence, (en hertz), indique la vitesse de transfert de l'information. La bande passante est : fréquence $\times$ largeur.

Mémoires

L'information est stockée en mémoire :

- mémoire vive (RAM = Random Access Memory) accessible en lecture et en écriture ;
- mémoire morte (ROM = Read Only Memory) accessible en lecture seule ;
- mémoire de masse (disque dur, clé USB, CD, DVD, bandes).

La mémoire vive est une mémoire volatile : si l'alimentation est coupée, les données qu'elle contient sont perdues. Elle stocke les programmes exécutés par le processeur.

La mémoire morte est une mémoire non volatile. Elle contient du code et des données qui ne sont modifiés que très rarement. Les ROM contiennent généralement les routines d'accès de base aux périphériques.

Dans les ROM classiques, l'information contenue est enregistrée de manière irréversible lors de la fabrication du circuit.

La mémoire de masse est un support de stockage généralement de grande capacité sur lequel des informations sont stockées et archivées de manière persistante.

1.4 Fonctionnement

Une machine ne comprend que le langage binaire et elle exécute tout ce qui lui est demandé sans intelligence. L'utilisateur communique avec elle en utilisant un langage de programmation qui est traduit en langage machine par un compilateur ou un assembleur. Si l'information peut s'écrire avec des nombres, il s'agit donc de pouvoir représenter un nombre dans une machine.

Dans un circuit avec un interrupteur, il passe du courant si l'interrupteur est fermé, et il n'en passe pas si l'interrupteur est ouvert. Si nous ajoutons une lampe dans ce circuit alors si la lampe est allumée, cela signifie "un", et si elle est éteinte cela signifie "zéro". Si un nombre peut s'écrire en base deux, il est donc possible de le représenter physiquement dans la machine avec plusieurs circuits en parallèle.

1.4.1 Systèmes d'exploitation

Un système d'exploitation, (OS = Operating System en anglais), est un ensemble de programmes qui s'exécutent lorsqu'un ordinateur est allumé. Tous les systèmes d'exploitation sont basés sur des concepts communs : des objets (les répertoires et fichiers, les processus, le matériel interne et périphérique, ...) chacun avec son outil respectif (gestion des fichiers, gestion des processus, gestion des périphériques). La mise en œuvre de ces concepts dépend du système d'exploitation. Pour Windows, nous avons les formats Fat32 et NTFS avec l'outils "Explorateur

de fichiers", les applications, les processus ou services avec l'outil "Gestionnaire de tâches", les cartes graphiques, les moniteurs, le réseau avec l'outil "Panneau de configuration".

Le système d'exploitation permet de :

- communiquer avec le disque dur afin d'y gérer les fichiers (leur attribuer un nom, les organiser en arborescence, ...) ;
- gérer les périphériques à l'aide de "pilotes" ;
- exécuter simultanément plusieurs programmes (en partageant le temps alloué à chacun) ;
- gérer l'authentification de chaque utilisateur et ses droits d'accès sur les fichiers (lecture, écriture, ...)

1.4.2 Organisation du disque dur

Un programme permet d'installer sur le disque dur le système d'exploitation. Un partitionnement peut être effectué ; ceci consiste à partager le disque en plusieurs parties afin de séparer par exemple les programmes des données ou d'installer différents systèmes d'exploitation.

Le disque a subi un "formatage de bas niveau" en usine lors de sa fabrication. Ceci a pour but d'organiser la surface du disque en éléments simples, (pistes, secteurs), qui permettent de localiser l'information. L'installation procède à un "formatage de haut niveau" qui organise les pistes et secteurs en un système de fichier qui sera géré par le système d'exploitation (système de fichier NTFS pour Windows, système Ext3, Ext4 pour Linux, ...).

Le système d'exploitation permet de procéder à un formatage de haut niveau autant de fois que nécessaire.

Durant le formatage les secteurs sont regroupés en blocs. Un bloc ou "cluster" devient alors la plus petite unité d'allocation ; la FAT (File Allocation Table) contient la liste des clusters du disque ou de la partition. Avec le système NTFS et une partition d'au moins 2 Go (Giga-octet), un cluster correspond à 4 Ko (Kilo-octet). Donc pour un fichier de 33,6 Ko, c'est 36 Ko qui sont réservés sur le disque, et pour un fichier de quelques octets, le minimum est réservé, soit 4 Ko.

1.4.3 Environnement de développement

Différentes distributions permettent de travailler avec le langage de programmation Python. La distribution standard en version au moins 3.5 nous suffira.

Aller sur la page `https://www.python.org/downloads/` et télécharger le fichier d'installation correspondant à votre système d'exploitation, "python-3.5.0.exe" par exemple pour un système Windows. Lorsque le téléchargement est terminé, lancer l'installation.

En général Python est déjà livré avec les différentes distributions Linux.

Pendant l'installation, un raccourci "Python IDLE (Python GUI)" est créé et servira à ouvrir l'interpréteur Idle. Si ce n'est pas le cas, il est possible de lancer l'application à partir du fichier de commande "idle.bat" qui se trouve à l'adresse "Python . . . /Lib/idlelib/ ".

Ouvrir Idle puis tester dans la fenêtre "Python shell" les différents menus. Le plus important sera de savoir ouvrir une nouvelle fenêtre, écrire et enregistrer un programme (nom de fichier avec l'extension ".py"), le fermer, l'ouvrir et lancer l'exécution avec la commande "Run Module". Deux raccourcis claviers seront très utilisés : les touches ctrl-s pour sauvegarder un fichier et la touche F5 pour interpréter et exécuter un programme.

Note : un fichier ".py" ne s'**ouvre** pas avec Idle mais il s'**édite** avec Idle.

Les bibliothèques complémentaires Matplotlib, NumPy et SciPy pourront être installées plus tard. Elles se trouvent sans difficulté sur le Web, par exemple à l'adresse `http://sourceforge.net/`.

Attention, il peut y avoir des problèmes de compatibilité entre les différentes versions de Python et des bibliothèques complémentaires.

L'installation d'une version portable sur un ordinateur ou sur une clé USB permet d'éviter toutes sortes de problèmes.

Winpython est un environnement complet pour Windows. Brancher une clé USB sur un ordinateur, aller sur la page `http://winpython.sourceforge.net/` où tout est expliqué.

Quand l'installation est terminée, la clé contient un dossier "WinPython-. . . ". Ce dossier peut être copié à volonté sur d'autres clés ou ordinateurs.

Pour ouvrir l'interpréteur Idle, cliquer dans le dossier "WinPython-. . . " sur "IDLE (Python GUI).exe" (ou sur le fichier de commande "idle.bat" qui se trouve à l'adresse "WinPython . . . /Python . . . /Lib/idlelib/ ").

La clé USB peut être utilisée sur n'importe quel ordinateur même si Python n'y est pas installé.

1.5 Exercices corrigés

Exercice 1.5.1 Architecture et système d'exploitation

Il s'agit ici d'un système Windows. Les questions peuvent facilement se transposer pour un autre système. Les réponses dépendent de la machine et du système installé.

1. Donner les caractéristiques de l'ordinateur utilisé (marque, type, . . .).
   Quels sont les périphériques externes connectés à l'ordinateur ?

2. Le système d'exploitation permet d'explorer l'ordinateur (le matériel et les logiciels). Cependant, certaines actions peuvent être impossibles ; il y a des

comptes administrateurs, des comptes utilisateurs dont certains avec des droits restreints. Noter quelques-unes de ces actions qui sont restreintes ou interdites (accès à une ressource, modification, utilisation d'une application) et justifier la raison de ces interdictions.

3. En utilisant un outil du **panneau de configuration**, donner la version du système d'exploitation installé sur l'ordinateur et les paramètres de l'ordinateur (nom, groupe de travail) ; préciser le type du système, les caractéristiques du processeur, la quantité de mémoire RAM installée.

4. Le système d'exploitation permet de gérer tous les périphériques. Lister les différents périphériques et préciser s'ils sont internes ou externes ; donner le modèle de la **carte graphique**, de la **carte réseau**, le modèle du **processeur**.

5. Quelle est la capacité du disque dur (hard drive) et, éventuellement, quelles sont les différentes partitions avec les capacités et les systèmes de fichiers respectifs ?

   Quelle est la taille de l'espace utilisé et quelle est la taille de l'espace libre ?

   Windows utilise une zone du disque dur (un fichier d'échange) comme de la mémoire RAM (mémoire virtuelle). Quel est la taille de ce fichier et sur quelle partition se trouve-t-il ?

6. La plus grande partie des fichiers du système d'exploitation sont regroupés dans un même dossier ; comment se nomme ce dossier ? Quelle est la taille de ce dossier sur le disque dur ? Combien contient-il de dossiers et fichiers ?

7. Les programmes sont en général regroupés dans un même dossier ; comment se nomme ce dossier et quelle est sa taille ? Combien de programmes sont installés sur l'ordinateur ? (Explorer le disque dur et utiliser le **gestionnaire de programmes**).

   A l'aide du **gestionnaire des tâches**, donner des informations sur les programmes et les processus qui s'exécutent en permanence sur l'ordinateur.

8. Les fichiers personnels d'un utilisateur sont en général regroupés dans un dossier. Comment se nomme ce dossier et où est-il stocké ?

Exercice 1.5.2

Supposons que le système de fichiers créé au formatage du disque dur d'un ordinateur utilise des clusters de 4 Ko. (1Ko = 1024 octets)

Quelle est la place totale en Ko réservée sur le disque pour stocker trois fichiers dont les tailles respectives sont 2000 octets, 5624 octets et 5234 octets ?

Corrigé

1 clusters = 4 Ko = 4096 octets.

Il faut donc 1 cluster pour le fichier de 2000 octets, 2 clusters pour le fichier de 5624 octets et 2 clusters pour le fichier de 5234 octets. Soit au total 5 clusters ce qui fait 20 Ko.

Exercice 1.5.3

1. Créer un dossier nommé **informatique** puis, dans ce dossier, créer un dossier nommé **test** ; quelles sont les tailles des différents dossiers créés et les tailles occupées sur le disque dur ?
2. Ouvrir un éditeur de texte basique, écrire le mot "bonjour" et enregistrer le fichier en format "texte brut" avec le nom "test1.txt" dans le dossier **test**. Quelle est la taille du fichier "test1" ? Quelle est la taille occupée par ce fichier sur le disque ?
3. Ecrire dans un deuxième fichier un mot de trois lettres puis un mot de six lettres séparés par une espace et l'enregistrer sous le nom "test2.txt". La taille de ce fichier et la taille qu'il occupe sur le disque sont-elles prévisibles ? Vérifier.

   Quelle est maintenant la taille du dossier **test** ?
4. Ecrire n fois la lettre "a", où n est la taille en octets d'un cluster, sans espace ou retour à la ligne (conseil : utiliser copier-coller). Enregistrer le fichier au format "texte brut" et vérifier la taille. Ajouter un "a" à la suite et observer la nouvelle taille.
5. Utiliser maintenant un éditeur permettant d'enregistrer le fichier "test1" sous différents formats. Comment expliquer les différences entre les tailles des fichiers ?

Corrigé

1. Les tailles sont nulles ; créer un dossier revient simplement à donner un nom à une zone du disque dur.
2. La taille du fichier est 7 octets (un octet par caractère). La taille occupée sur le disque est la taille d'un cluster qui dépend de la taille des partitions et de leur format (4 Ko pour NTFS et une partition d'au moins 2Go).
3. La taille du fichier test2.txt est de 10 octets (l'espace compte comme un caractère). Le fichier occupe un cluster sur le disque.

   Le dossier test a maintenant la taille de deux clusters.

4. La taille occupée sur le disque est d'un cluster ; avec un "a" en plus, la taille passe à deux clusters.
5. La taille dépend du format et de l'éditeur. Plusieurs Ko peuvent être utilisés pour la définition du format et les informations concernant l'éditeur.

Exercice 1.5.4 Environnement de développement

L'environnement de développement utilisé est **Idle**. Cette application regroupe un éditeur de texte avec différents outils, un debugger et un interpréteur. Ouvrir l'application avec un double-clic sur "Python Idle" ; la fenêtre "Python shell" (l'interpréteur Python) s'ouvre et affiche :

```
Python 3.5.0 (v3.5.0 ...
[MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more ...
>>>
```

Nous pouvons alors écrire, après l'invite de commande " >>> ", des commandes Python, c'est-à-dire évaluer des expressions ou exécuter des instructions de manière interactive. Nous pouvons aussi écrire dans un fichier et l'enregistrer avec l'extension .py, par le menu **File** puis **Save As...** pour choisir le dossier de destination et entrer le nom du fichier.

Ouvrir la fenêtre "Python shell", écrire `print ('bonjour')` et appuyer sur la touche "entrée" (on appuie sur la touche "entrée" à la fin de chaque ligne) :

```
>>> print ('bonjour')
bonjour
>>>
```

Ecrire : `print ('bonjour, tout le monde')`
puis : `print ('bonjour,' + 'tout le monde')`.
Quelle est la différence à l'affichage ?

Tester les instructions suivantes, en les exécutant les unes après les autres :

```
1+2
print(1+2)
x=1+2
print(x)
print('x est égal à',x)
```

```
print('"x est égal à',x,'"')
```

L'éditeur colore de lui-même certains mots, (**mot réservé**), qui sont des mots clés, des noms de fonctions (ici la fonction **print**), des commentaires, ... : c'est la coloration syntaxique. Ainsi, si un mot réservé n'est pas écrit correctement, il est facile de s'en rendre compte. Attention, le langage Python est sensible à la casse, ceci signifie que les lettres capitales et les lettres minuscules sont des caractères différents. Par exemple, la fonction **print** ne peut pas s'écrire "Print".

Les mots clés réservés sont : *and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield.*

Question 1 : écrire `nom=input('Quel est votre nom ? ')` ; entrer alors votre nom.

Demander d'afficher la phrase "Bonjour ....., comment vas-tu ?", avec votre nom à la place des pointillés (sans écrire votre nom dans l'instruction).

Question 2 : écrire, sans aucun chiffre dans la parenthèse suivant **print**, une suite d'instructions qui affiche la phrase : x est égal à 37+74 c'est-à-dire 111.

Question 3 : tester les instructions suivantes, en les exécutant les unes après les autres :

```
y=5
print('y est égal à',y)
y=2*y
print('y est égal à',y)
z=2**y
print('z est égal à',z)
print(y==z)
```

Observer les valeurs prises par les variables y et z à l'issue de ces instructions.

Les opérateurs sur les nombres sont +, −, *, / qui correspondent aux quatre opérations habituelles, ** pour l'exponentiation, // et % pour le quotient et le reste de la division euclidienne et >, >=, <, <=, ==, != pour les comparaisons. Attention, le symbole "=" n'a pas le sens mathématiques habituel. L'instruction `z=z+2` signifie "affecter à z la valeur de z+2".

Question 4 : tester ces différents opérateurs sur les variables a $= 9$ et b $= 2$.

Question 5 : tester les priorités de ces opérateurs, par exemple 6/2*2 ou 6/2/2 ou 6/2**2. Ecrire une instruction permettant de calculer la quantité $ax^2 + bx + c$

Cet ouvrage est destiné principalement aux étudiants en première année de classes préparatoires aux Grandes Écoles. Son ambition est de les accompagner tout au long de l'année et de leur permettre un travail en autonomie. Il se veut le plus complet possible tout en restant dans les limites du programme d'informatique commun aux différentes voies de la filière scientifique.

Il convient aussi aux étudiants en deuxième année qui souhaitent tester ou consolider leurs connaissances dans la perspective des concours ainsi qu'aux enseignants souhaitant se former.

Le langage de programmation utilisé, Python version 3, est très riche mais il a été choisi de n'utiliser que le strict minimum permettant de répondre aux attentes du programme.

Treize chapitres sont composés d'une partie cours suivie par de nombreux exercices variés et progressifs qui sont tous corrigés. Deux chapitres concluent ce livre avec les principaux outils des bibliothèques NumPy, Matplotlib, SciPy, utilisées dans le cadre de l'ingénierie numérique et de la simulation, et un résumé condensé du langage Python.

Une grande partie de ce livre est accessible à toute personne souhaitant s'initier à la programmation en Python et peut convenir également aux élèves et aux enseignants dans le cadre de la spécialité ISN en terminale S.

Serge Bays est agrégé de mathématiques et enseigne l'informatique en classes préparatoires au lycée Les Eucalyptus à Nice.

9 782340 012172

www.editions-ellipses.fr