Introduction à l'informatique en BCPST

Alexandre Benoit

BCPST

L'informatique en BCPST

« L'enseignement de l'informatique en classes préparatoires de la filière BCPST a pour objectif d'introduire puis de consolider les concepts de base de l'informatique, à savoir l'analyse et la conception de processus de raisonnement automatisé, c'est-à-dire des algorithmes, et la question de la représentation des données. »
Programme officiel 2013.

L'informatique en BCPST

« L'enseignement de l'informatique en classes préparatoires de la filière BCPST a pour objectif d'introduire puis de consolider les concepts de base de l'informatique, à savoir l'analyse et la conception de processus de raisonnement automatisé, c'est-à-dire des algorithmes, et la question de la représentation des données. »
Programme officiel 2013.
En pratique :

- On étudiera et maitrisera certains algorithmes fondamentaux

L'informatique en BCPST

« L'enseignement de l'informatique en classes préparatoires de la filière BCPST a pour objectif d'introduire puis de consolider les concepts de base de l'informatique, à savoir l'analyse et la conception de processus de raisonnement automatisé, c'est-à-dire des algorithmes, et la question de la représentation des données. »

Programme officiel 2013.

En pratique :

- On étudiera et maitrisera certains algorithmes fondamentaux
- On apprendra à programmer dans le langage **Python**.

L'informatique en BCPST

« L'enseignement de l'informatique en classes préparatoires de la filière BCPST a pour objectif d'introduire puis de consolider les concepts de base de l'informatique, à savoir l'analyse et la conception de processus de raisonnement automatisé, c'est-à-dire des algorithmes, et la question de la représentation des données. »
Programme officiel 2013.
En pratique :

- On étudiera et maitrisera certains algorithmes fondamentaux
- On apprendra à programmer dans le langage **Python**.
- On appliquera le tout à des problèmes concrets.

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,
- En première année : cours d'introduction,

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,
- En première année : cours d'introduction,
- En deuxième année : Réalisation d'un projet,

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,
- En première année : cours d'introduction,
- En deuxième année : Réalisation d'un projet,
- Au concours : exercice et exposée de la solution du projet avec examen du code imprimé sur papier.

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,
- En première année : cours d'introduction,
- En deuxième année : Réalisation d'un projet,
- Au concours : exercice et exposée de la solution du projet avec examen du code imprimé sur papier.

L'informatique aux concours

- Épreuve **obligatoire** à l'oral du concours, en même temps que l'épreuve de mathématiques,
- En première année : cours d'introduction,
- En deuxième année : Réalisation d'un projet,
- Au concours : exercice et exposée de la solution du projet avec examen du code imprimé sur papier.

Comparable au TIPE

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,
4. qu'il puisse exécuter les quatre opérations élémentaires (addition, soustraction, multiplication, division) et -souvent- qu'il puisse extraire une racine carrée ou adresser une table qui en contient,

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,
4. qu'il puisse exécuter les quatre opérations élémentaires (addition, soustraction, multiplication, division) et -souvent- qu'il puisse extraire une racine carrée ou adresser une table qui en contient,
5. qu'il puisse exécuter des programmes enregistrés en mémoire. »

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,
4. qu'il puisse exécuter les quatre opérations élémentaires (addition, soustraction, multiplication, division) et -souvent- qu'il puisse extraire une racine carrée ou adresser une table qui en contient,
5. qu'il puisse exécuter des programmes enregistrés en mémoire. »

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,
4. qu'il puisse exécuter les quatre opérations élémentaires (addition, soustraction, multiplication, division) et -souvent- qu'il puisse extraire une racine carrée ou adresser une table qui en contient,
5. qu'il puisse exécuter des programmes enregistrés en mémoire. »

Bernard Cohen (2000)

Qu'est ce qu'un ordinateur ?

« Souvent, quelques-unes des caractéristiques fondamentales nécessaires pour être considérées comme un ordinateur sont :

1. qu'il soit électronique,
2. numérique (au lieu d'analogique),
3. qu'il soit programmable,
4. qu'il puisse exécuter les quatre opérations élémentaires (addition, soustraction, multiplication, division) et -souvent- qu'il puisse extraire une racine carrée ou adresser une table qui en contient,
5. qu'il puisse exécuter des programmes enregistrés en mémoire. »

Bernard Cohen (2000)

Exemple

Les ordinateurs de bureau, les ordinateurs portables mais aussi les tablettes ou les smartphones sont des ordinateurs.

Architecture des ordinateurs

Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
- Le processeur ;
- La mémoire de masse ;
- Carte graphique ;
- Périphériques.

Architecture des ordinateurs

Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
  - Pas de sens à priori ;
  - Inertie ;
  - Accès direct.
- Le processeur ;
- La mémoire de masse ;
- Carte graphique ;
- Périphériques.

Architecture des ordinateurs

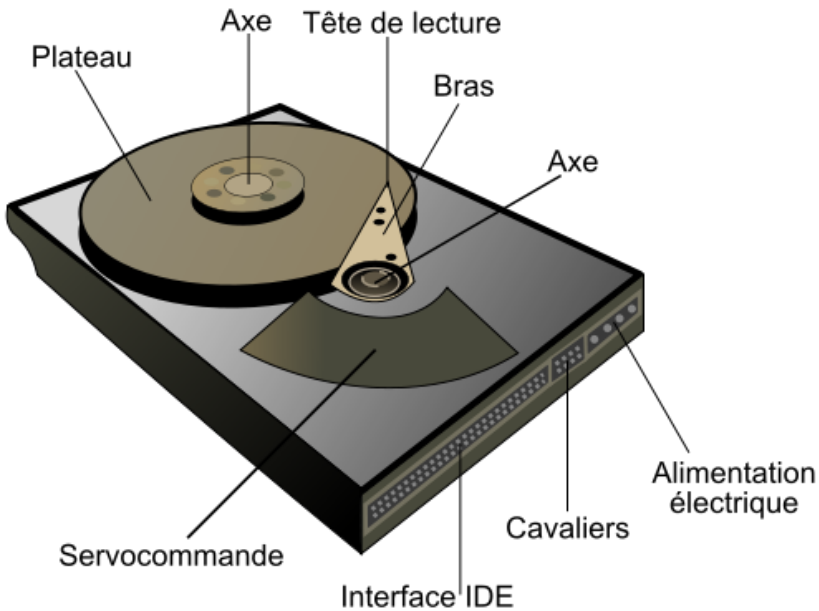
Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
- Le processeur ;
  - Arithmétique de base ($+$, $-$, $\times$, $\div$) et opérations logiques (disjonction, conjonction, négation) sur des valeurs de vérité ;
  - Toute petite mémoire mais avec un accès très rapide (plus rapide que la mémoire vive) ;
  - Accès direct à la mémoire vive.
- La mémoire de masse ;
- Carte graphique ;
- Périphériques.

Architecture des ordinateurs

Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
- Le processeur ;
- La mémoire de masse ;
  - Mémoire non volatile (elle reste en place après une coupure de courant) ;
  - Grosse capacité ;
  - Accès très lent (1000 fois plus lente que la mémoire vive) ;
  - Différents aspects : Disque dur (ordinateur), mémoire flash (smartphone), CD Rom, Clé USB, etc ...
- Carte graphique ;
- Périphériques.



Architecture des ordinateurs

Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
- Le processeur ;
- La mémoire de masse ;
- Carte graphique ;
  Tous les calculs liés à l'affichage sont délégués à la carte graphique.
- Périphériques.

Architecture des ordinateurs

Les composants essentiels de l'ordinateur sont :

- La mémoire vive ;
- Le processeur ;
- La mémoire de masse ;
- Carte graphique ;
- Périphériques.
  Clavier, Lecteur CD, Souris,
  Moniteur, Enceinte, Scaner,
  Webcam, Micro etc ..

Système d'exploitation

Architecture : Partie matérielle.

Système d'exploitation

Architecture : Partie matérielle.

Système d'exploitation : Partie logicielle.

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;
- Gérer l'organisation du disque dur et de ses fichiers ;

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;
- Gérer l'organisation du disque dur et de ses fichiers ;
- Contrôler l'accès aux données du disque ;

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;
- Gérer l'organisation du disque dur et de ses fichiers ;
- Contrôler l'accès aux données du disque ;
- Gérer le lancement des différentes applications utilisées ;

Système d'exploitation

Architecture : Partie matérielle.
Système d'exploitation : Partie logicielle. Il existe deux grandes familles de système d'exploitation

Exemple

- Les systèmes issues d'UNIX (Mac OS, IOS, GNU/Linux, Android)
- Les systèmes issues de la famille Microsoft Windows

Le système d'exploitation a les responsabilités suivantes :

- Donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;
- Gérer l'organisation du disque dur et de ses fichiers ;
- Contrôler l'accès aux données du disque ;
- Gérer le lancement des différentes applications utilisées ;
- Servir de garde-fou en cas de tentative de mauvaise utilisation des ressources de l'ordinateur

Système de fichiers

- Sous Linux, tous les fichiers sont regroupés dans une unique arborescence. Le sommet de l'arbre est un répertoire appelé / . Cette racine possède plusieurs sous-répertoire, dont généralement un appelé home.

Système de fichiers

- Sous Linux, tous les fichiers sont regroupés dans une unique arborescence. Le sommet de l'arbre est un répertoire appelé / . Cette racine possède plusieurs sous-répertoire, dont généralement un appelé home.
- Pour se repérer, on utilise le chemin absolu du fichier (ex : /home/login/travail/td1/data/img.jpeg) ou le chemin relatif (ex :data/img.jpeg)

Droit d'accès et de modification

Dans un système GNU/Linux, tout fichier se voit attribuer des droits pour 3 identités :

- le propriétaire - c'est l'utilisateur qui a créé le fichier ou l'utilisateur que root a désigné comme propriétaire
- le groupe (qui n'est pas forcément le groupe du propriétaire)
- les autres (ceux qui ne font pas partie du groupe)

Droit d'accès et de modification

Dans un système GNU/Linux, tout fichier se voit attribuer des droits pour 3 identités :

- le propriétaire - c'est l'utilisateur qui a créé le fichier ou l'utilisateur que root a désigné comme propriétaire
- le groupe (qui n'est pas forcément le groupe du propriétaire)
- les autres (ceux qui ne font pas partie du groupe)

La commande ls -l nous permet d'afficher les droits d'un fichier sous GNU/Linux.
Pour chaque identité (voir plus haut), il existe 3 droits d'accès :

- r - read (le droit de lecture)
- w - write (le droit d'écriture)
- x - execute (le droit d'exécution)

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.
  - Libre : Toute utilisation (y compris commerciale) possible, accès au code source, maintenue par une fondation.

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.
  - Libre : Toute utilisation (y compris commerciale) possible, accès au code source, maintenue par une fondation.
  - interprété instructions exécutées une par une, accès direct par une ligne de commande

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.
  - Libre : Toute utilisation (y compris commerciale) possible, accès au code source, maintenue par une fondation.
  - interprété instructions exécutées une par une, accès direct par une ligne de commande
  - haut niveau Proche du langage humain

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.
  - Libre : Toute utilisation (y compris commerciale) possible, accès au code source, maintenue par une fondation.
  - interprété instructions exécutées une par une, accès direct par une ligne de commande
  - haut niveau Proche du langage humain
  - Modulaire Bibliothèque pour le calcul scientifique, le graphique, l'analyse de signaux et d'images, etc.

Le langage Python

- Python est un langage de programmation moderne, utilisé de plus en plus dans l'industrie et dans l'éducation,
- Ce langage est :
  - Multi-plateforme : Linux, Windows, Mac OS, Android, etc.
  - Libre : Toute utilisation (y compris commerciale) possible, accès au code source, maintenue par une fondation.
  - interprété instructions exécutées une par une, accès direct par une ligne de commande
  - haut niveau Proche du langage humain
  - Modulaire Bibliothèque pour le calcul scientifique, le graphique, l'analyse de signaux et d'images, etc.
  - Très simple à prendre en main (tout se fait sur l'indentation du programme).

Spyder : Un **IDE** pour **Python**

La fenêtre de Spyder est divisée en trois parties

- L'éditeur à gauche, dans lequel on écrira les programmes.
- L'explorateur en haut à droite, que nous utiliserons surtout comme débogueur, mais qui peut également servir de documentations.
- La console interactive en bas à droite, dans laquelle s'exécuteront les programmes.

La console interactive

Taper dans la console

2+2

La console interactive

Taper dans la console

2+2

a=2

puis

a+a

La console interactive

Taper dans la console

```
2+2
```

```
a=2
```

puis

```
a+a
```

Taper ensuite

```
b+1
```

On obtient alors une erreur, la dernière ligne du message indique d'où vient l'erreur.

La console interactive

Taper dans la console

```
2+2
```

```
a=2
```

puis

```
a+a
```

Taper ensuite

```
b+1
```

On obtient alors une erreur, la dernière ligne du message indique d'où vient l'erreur.

Quels sont les défauts de cette console ?

Éditeur

Avec l'éditeur, on peut sauvegarder son programme et le modifier facilement

Éditeur

Avec l'éditeur, on peut sauvegarder son programme et le modifier facilement
Essayer :

```
print("Bonjour")
x=42
print(x)
```

Éditeur

Avec l'éditeur, on peut sauvegarder son programme et le modifier facilement
Essayer :

```
print("Bonjour")
x=42
print(x)
```

La première chose à faire quand on commence un programme est de le sauvegarder.

Éditeur

Avec l'éditeur, on peut sauvegarder son programme et le modifier facilement
Essayer :

```
print("Bonjour")
x=42
print(x)
```

La première chose à faire quand on commence un programme est de le sauvegarder.
Pour exécuter ce travail, on utilise la touche F5 ou l'icône avec le *petit bonhomme qui court*.

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.

- Appeler la commande débugeur du menu Execution (Ctrl+F5 ou la coccinelle).

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.

- Appeler la commande débugeur du menu Execution (Ctrl+F5 ou la coccinelle).
- Appuyer sur le bouton *Pas en avant* ou taper *n* dans l'interpréteur. Que remarquez-vous ?

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.

- Appeler la commande débugeur du menu Execution (Ctrl+F5 ou la coccinelle).
- Appuyer sur le bouton *Pas en avant* ou taper *n* dans l'interpréteur. Que remarquez-vous ?
- Continuer jusqu'à trouver le problème.

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.
Il n'est pas toujours aisé de tester chaque ligne de votre programme (il peut en exister beaucoup), on peut aussi choisir ou s'arrêter.

- Appeler la commande débugeur du menu *Run* (Ctrl+F5 ou la coccinelle).

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.
Il n'est pas toujours aisé de tester chaque ligne de votre programme (il peut en exister beaucoup), on peut aussi choisir ou s'arrêter.

- Appeler la commande débugeur du menu *Run* (Ctrl+F5 ou la coccinelle).
- *Ajouter un point d'arrêt* dans le menu *Run* (F12), on peut également double cliquer dans le marge gauche du programme

Le débogueur

On prend l'exemple suivant :

```
x=10
y=7
x=x+y
y=x
x=5/(x-y)
```

On remarque que cet exemple provoque une erreur.
Il n'est pas toujours aisé de tester chaque ligne de votre programme (il peut en exister beaucoup), on peut aussi choisir ou s'arrêter.

- Appeler la commande débugeur du menu *Run* (Ctrl+F5 ou la coccinelle).
- *Ajouter un point d'arrêt* dans le menu *Run* (F12), on peut également double cliquer dans le marge gauche du programme
- Choisir *Continuer* (Le bouton *play* ou *c* dans l'interpréteur interactif).

Un premier exercice

1. Taper le programme suivant dans l'éditeur et l'exécuter. Que se passe-t'il ?

```
i=10
while i!=0:
    i = 1-i
```

Un premier exercice

1. Taper le programme suivant dans l'éditeur et l'exécuter. Que se passe-t'il ?

```
i=10
while i!=0:
    i = 1-i
```

2. Éxecuter ce programme pas à pas et observer les valeurs successives prises par la variable i. Expliquer le comportement observé à la première question.

Un premier exercice

1. Taper le programme suivant dans l'éditeur et l'exécuter. Que se passe-t'il ?

```
i=10
while i!=0:
    i = 1-i
```

2. Éxecuter ce programme pas à pas et observer les valeurs successives prises par la variable i. Expliquer le comportement observé à la première question.
3. Placer un point d'arrêt à un endroit approprié du programme pour montrer son comportements sans avoir besoin de détailler les étapes inutiles.

Un premier exercice

1. Taper le programme suivant dans l'éditeur et l'exécuter. Que se passe-t'il ?

```
i=10
while i!=0:
    i = 1-i
```

2. Éxecuter ce programme pas à pas et observer les valeurs successives prises par la variable i. Expliquer le comportement observé à la première question.
3. Placer un point d'arrêt à un endroit approprié du programme pour montrer son comportements sans avoir besoin de détailler les étapes inutiles.
4. Proposer une correction à ce programme pour qu'il termine.

Quelques références :

- Le site de Sylvain Pelletier pour installer Spyder chez vous comme au lycée : `http://math1b.bcpsthoche.fr/python.php`
- L'informatique pour tous de Wack et al aux editions Eyrolles (disponible gratuitement sur le web) `https://wiki.inria.fr/sciencinfolycee/Fichier:Informatique_pour_tous_en_classes_préparatoires_aux_grandes_écoles.pdf`. (Ce cours est largement inspiré de ce livre). Attention : une grosse partie de ce livre n'est pas au programme BCPST
- Le site openclassrooms propose un bon tutoriel sur Python (et d'autres) et démarre d'un niveau basique.