

CRAYONS ET QUESTIONS

Proverbe chinois : «Celui qui pose une question risque cinq minutes d'avoir l'air bête, celui qui ne pose pas de question restera bête toute sa vie».

Bernard Werber, dans les Thanatonautes

Les crayons c'est pas du bois et de la mine, c'est de la pensée par les phalanges.

Henri de Toulouse Lautrec

À la fin de ce chapitre, je sais :

 pourquoi et comment travailler l'informatique.

A L'ingénieur, la construction et le calcul

Avant d'entrer dans le vif du sujet, il peut être utile de s'interroger sur la finalité potentielle de vos études et donc, de se demander ce que recouvre la profession d'ingénieur : qu'est-ce qu'un ingénieur ?

Le technicien est celui à qui on confie une mission qu'il sait faire. Un ingénieur, c'est celui à qui on confie une mission qu'il va savoir faire : la réponse n'existe pas nécessairement sur étagère, il lui faut souvent la construire, la calculer. Or, dans le monde de l'industrie actuelle, les caractéristiques des objets développés font que vous ne pouvez rien construire sans l'informatique.

Si vous voulez être un ingénieur capable, il vous faut donc maîtriser l'informatique, tout comme vos ancêtres avant vous ont appris à maîtriser le crayon, les abaques, la règle, le compas, la trigonométrie, la corde à nœuds, le calcul dans le système de numération de position, les tables de logarithmes, la règle à calcul et les machines à calculer. Ce n'était pas un choix mais une évidence qui s'imposait, cela facilitait la construction et les calculs. **Aujourd'hui, l'évidence informatique s'impose à vous de la même manière et tout en vous confiant une puissance de calcul inégalée : à vous de la maîtriser.**

B Programme d'informatique et philosophie de la construction

Comme cela est détaillé au chapitre ??, **l'informatique est la science de la construction de l'information par le calcul**. Le programme officiel d'informatique est à consulter en ligne. Il est très précis et vaste et indique clairement ce qu'il faut savoir et savoir faire. Il précise également à la fin les éléments du langage Python que l'on attend que vous maîtrisiez.

Un extrait me semble particulièrement important :

La pratique régulière de la résolution de problèmes par une approche algorithmique et des activités de programmation qui en résultent constitue un aspect essentiel de l'apprentissage de l'informatique.

Cette pratique régulière se fait en TP. Les TP qui vous sont proposés présentent des problèmes et montrent une voie pour construire une solution à ce problème. Car *résoudre par une approche algorithmique des problèmes*, cela signifie construire une solution à un problème. Or une solution est une information.

D'une manière générale, les TP sont importants, notamment car il n'est pas prévu dans le programme officiel de cours magistraux au premier semestre. Les TP abordent les sujets du programme dans un ordre logique, en commençant par les concepts les plus faciles. Par ailleurs, mes TP sont progressifs, les questions les plus faciles se trouvent au début et les plus difficiles à la fin. Un TP raconte une histoire : il s'agit souvent de construire¹ une information, de la faire apparaître : un message secret, la solution à un problème concret ou la stratégie gagnante d'un jeu. Cette construction se fait par étapes, progressivement, tout au long du TP. Cette philosophie s'inspire des épreuves de concours qui vous présentent également la construction d'une information autour d'un thème, si bien que chaque TP vous prépare davantage au concours.

Tout ingénieur sait que pour bien construire, il faut commencer par bien bricoler : bricoler en informatique, cela signifie imaginer des algorithmes sur des cas simples, les implémenter, tester les codes et recommencer sur des situations plus complexes. Se construire des outils pour en construire d'autres encore plus puissants, tout comme le bricoleur élabore parfois un outil dédié pour réussir une opération mécanique délicate. L'avantage de l'informatique, c'est qu'on ne risque ni de casser quelque chose, ni de se faire mal : au pire, on a fabriqué un outil et on appréhende mieux une situation problème. Alors, bricolons!

1. Programmer, c'est bricoler de l'information!

C Pourquoi travailler l'informatique?

Il est pertinent de travailler l'informatique parce que² :

1. l'informatique rapporte des points aux concours, points qui font la différence,
2. ces points sont plus faciles à gagner que les points en mathématiques,
3. on manque d'ingénieures et d'ingénieurs capables dans le domaine,
4. l'évolution de votre carrière ne sera pas la même si vous savez utiliser pertinemment l'informatique dans votre contexte professionnel,
5. l'informatique façonne et façonnera toute votre vie. Tout simplement.
6. l'informatique, c'est passionnant.

D Comment travailler l'informatique?

Mon cours en ligne suit exactement le programme officiel. Sur ce site, vous trouverez tous les cours et tous les TP de chaque semestre, avec les solutions. Ils sont à votre disposition dans un but précis : aider simultanément ceux qui ont le plus de difficultés et ceux qui veulent aller plus loin.

Sur le plan matériel, l'informatique étant aujourd'hui ubiquitaire, vous pouvez travailler sur n'importe quel ordinateur³ ainsi que sur vos tablettes ou smartphones. Après avoir installé l'interprète Python, un simple éditeur de texte suffit généralement pour composer les codes. Cependant, un environnement de développement logiciel (IDE en anglais) est souvent un plus en termes de lisibilité et d'intelligibilité du code ainsi qu'en termes de rapidité de développement. Généralement, les classes préparatoires utilisent Pyzo, un IDE orienté calcul scientifique qui a le mérite d'être simple, disponible sur tous les OS et libre.

Sur le plan pédagogique, les sections qui suivent donnent quelques conseils éclairés.

a Avant le TP

- Lire le cours associé au TP : cela veut dire se poser des questions sur le sens du chapitre en prenant des notes sur les concepts présentés (avec un crayon et du papier).
- Lire le TP en entier.
- Préparer les premières questions du TP à l'écrit (avec un crayon et du papier).

b Pendant le TP sur machine

- Ne pas hésiter à prendre le clavier et la souris. Les partager avec son binôme régulièrement. Il faut être acteur de son apprentissage et à l'écoute.
- Quand on n'a pas le clavier, on peut écrire, imaginer et proposer des solutions (avec un crayon et du papier).

2. du plus pragmatique au plus important!

3. car Python est un langage interprété

- Commenter ou prendre des notes sur les points les plus difficiles ou les plus importants du TP (avec un crayon et du papier).
- Interpeler régulièrement l'enseignant et lui poser les questions en lien avec le TP et le cours.

c Après le TP

- Finir le TP si vous n'avez pas eu le temps de finir en classe.
- Poser des questions s'il y a encore des zones d'ombres.
- Écrire une petite synthèse sur le TP que vous pourrez relire avant le devoir (avec un crayon et du papier).
- Suggérer des améliorations au professeur.

d Sur le long terme (trois semestres mois)

- Lire le cours en avance de phase : on comprend toujours mieux la deuxième fois et encore mieux la troisième !
- Poser des questions : à l'oral, par écrit.
- Repérer ses lacunes et les combler : syntaxe du langage, compréhension des structures et des algorithmes, intérêts des concepts. Il est souvent plus chronophage d'avoir des lacunes que de les combler.
- S'entraîner avec un autre élève : se poser des questions, tester l'autre sur la compréhension, expliquer à l'autre les concepts que l'on a soi-même compris. C'est ainsi que vous progresserez vite.
- Créer un lien entre les différents thèmes étudiés. Par exemple, se demander pourquoi je vous présente la récursivité avant les approches diviser pour régner et gloutonne ?
- Associer un exemple concret à chaque concept (généralement abstrait) et être capable de l'expliquer à un autre. Par exemple, associer l'exemple du Morpion à l'algorithme Minimax ou la programmation dynamique au problème du sac à dos.

Pour plus d'éléments sur les techniques d'apprentissage, je vous conseille le site [The Learning Scientist](http://TheLearningScientist.com).