

Le cours d'algorithmique

Formation d'ingénieur de l'université des Antilles

Auteur : Vincent Pagé : vincent.page@univ-antilles.fr

Introduction

Dans ce cours, nous verrons les rudiments de l'algorithmique pour vos formations (Matériaux et Systèmes énergétiques). Mon objectif est simple : ne pas focaliser sur les détails, mais vous permettre de faire des choses rapidement. Ce document est extrêmement synthétique. Il ne vous dispense pas d'aller en cours ni d'essayer de programmer vous même. Pour plus de détails, vous pouvez me poser des questions et/ou chercher un peu sur le net ou dans des cours plus détaillés.

Les concepts présentés s'appliquent à presque tous les langages que je connais. Lorsque je devrais faire un vrai programme pour vous présenter quelque chose, je le ferais en python. Installez donc python sur votre machine. Les exemples seront fait pour fonctionner en python 3.

Le dépôt dans lequel vous avez trouvé ce document contient les exemples que nous avons vu en cours. ## Cours 1

Concepts d'algorithmique d'aujourd'hui.

Ceci est ma vision de la programmation actuelle. Tout le monde ne s'y retrouvera peut être pas, mais il me semble que la plupart des développeurs en entreprise seront d'accord avec le constat suivant : Programmer (ou coder) fait appel à deux grands capacités : - la stratégie - la tactique

Explications : TODO

un cours focalise souvent sur la tactique, alors que les langages actuels permettent d'en faire abstraction assez fréquemment. Par exemple, si je dois trier un tableau par ordre croissant, nous avons, dans tous les langages évolués, des fonctions pour le faire.

La stratégie, elle est utile pour programmer mais aussi pour la coordination de n'importe quel projet que vous aurez à réaliser.

Voyons donc le minimum de tactique à savoir pour commencer

Les variables

Comprenons une chose tout d'abord : un programme informatique ne fait qu'une chose : il manipule des variables. Toutes les informations que doit gérer votre programme doivent donc se retrouver dans des variables. Vous en connaissez vraisemblablement quelques types simples de variables : - les entiers (int) - les nombres à virgules (float) - les vrais ou faux (booleen) - les chaînes de caractères (string)

Dans les variables, je stocke des valeurs.

```
a=5
b=7
print(a+b)
```

j'ai créé une variable *a*, lui ai donné la valeur 5, puis crée une variable *b*, lui ai donné la valeur 7, puis affiché le résultat de la somme des valeurs des deux variables.

Les tests (if)

TO DO

Les boucles Tant que (while)

Les fonctions.

Le code que j'ai présenté juste avant est le programme principal. C'est ce que fait mon programme.

Un vrai bon programme découpe le code en petites actions que le programme principal organise. Si je veux programmer ## Cours 2

les tableaux

intérêt

Avec les types de variables que nous avons vus dans le cours précédent, nous pouvons faire beaucoup de choses, mais cela devient vite pénible.

Imaginons que je doive faire un programme qui gère vos notes, je dois stocker une note (un float) par étudiant. Je pourrais par exemple choisir un ordre pour rentrer les notes et stocker la note de chaque étudiant dans une variable. Disons que ma classe ne contient que 3 étudiants.

```
e1 = 12
e2 = 9.5
e3 = 14
```

Pour calculer la moyenne de ma promotion, je pourrais ajouter ceci au code qui précède :

```
moyenne = (e1+e2+e3) / 3
print (moyenne)
```

Si finalement, je dois ajouter un 4eme étudiant arrivé en retard, mon code est transformé à de multiples endroits pour devenir :

```
e1 = 12
e2 = 9.5
e3 = 14
e4 = 2 # oui, il est mauvais, en plus
```

```
moyenne = (e1+e2+e3+e4) / 4
print (moyenne)
```

A chaque ajout d'étudiant, il faudra que je pense à faire toutes les modifications. A terme, je suis sûr de faire une erreur...

De plus, dans le cas de votre promotion, il me faudrait une vingtaine de variables, ce qui devient lourd et pénible. Il est temps d'introduire les tableaux. En python, on les appelle des *listes*, mais c'est sans importance pour nous.

Voici ce que je voudrais : une seule variable contenant ces informations, rangées dans des cases séparées.

notes
12
9.5
14
2

Ici, *notes* est le noms de mon tableau.

Manipulations de base sur les tableaux

Creation d'un tableau

parcours d'un tableau avec des boucles.