

IFT211/IFT776

Programmation scientifique en Python

Présentation du cours

Gabriel Girard

Département d'informatique



4 janvier 2018

Présentation du cours

- 1 Buts du cours
- 2 Résolution de problèmes

Objectifs

- Apprendre à implémenter une solution à un problème donné ;
- Fournir une solution bien codée ;
- Fournir une solution bien documentée ;
- Apprendre à respecter des contraintes.

Pourquoi maîtriser la syntaxe et la sémantique d'un langage est important ?

« When you are programming, you are teaching possibly to stupidest thing in the universe, a computer, how to do something »

Gabe Logan Newell

Valve Software

(Half-life, Portal, ...)

Les phases

- 1 *Ingénierie des exigences* ;
- 2 Analyse ;
- 3 Conception ;
- 4 Implantation (langage de programmation Python) ;

- 5 *Mise au point* ;
- 6 *Maintenance*.



Problèmes de mise au point



Exemples d'erreurs réelles et célèbres

- 1 F16 (mauvaise analyse des besoins)
- 2 Cour de Paris (1989)
Erreur dans la transmission des offenses
- 3 USS Sheffield (Guerre des Falkland)
- 4 Boeing 767 (logiciel trop efficace)

Exemples d'erreurs et des coûts associés

- Ariane 5 : coût \equiv \$ 500 millions
- Therac-25 - machine de radiothérapie : coût \equiv 3 morts, 3 blessés graves
- Échec du missile Patriot : coût \equiv 28 morts, 100 blessés
- Écrasement du *Mars Climate Orbiter* : coût \equiv \$ 125 millions
- Bug de l'an 2000 : coût \equiv \$ 500 milliards
- ... Il y a plein d'autres exemples...

Plan de cours

Loi de Murphy

En fait c'est une série de lois et de lois connexes ou corollaires...

- 1 Rien n'est aussi simple qu'il n'y paraît..
- 2 Tout prend plus de temps que vous ne le pensez.
- 3 Tout ce qui peut aller mal ira mal.
- 4 S'il y a un risque pour que plusieurs choses aillent mal, c'est celle qui causera le plus de dommage qui ira mal.
- 5 Corollaire : s'il y a un pire moment pour que quelque chose aille mal, c'est à ce moment-là que cela ira mal.

Conclusion

- 1 Larry Flon
“There is no programming language—no matter how structured—that will prevent programmers from making bad programs.”
- 2 Brian Kernigan
“Controlling complexity is the essence of computer programming.”