

## Cours No 1 : Introduction

Notes de cours - 2007-2012

### 1 Généralités sur le cours

#### Ouvrages liés

J'ai notamment utilisé pour construire ces notes cours les ouvrages suivants :

- "Passport pour l'algorithmique et Scheme" de R.Cori et P.Casteran.
- "Structure and Interpretation of Computer Programs" de Abelson, Fano, Sussman.
- "Programmation récursive (en Scheme)" de Christian Queinnec

#### Programmation Applicative

- Programmation dans laquelle un programme est un ensemble de définition de fonctions et où l'exécution d'un programme est une succession d'**application** de fonctions à des arguments au sens algébrique du terme (calcul utilisant des opérations, des lettres et des nombres).
- Programmation où la mémoire utilisée par les calculs est automatiquement gérée par l'interpréteur du langage.

**Programmation récursive** : programmation utilisant des fonctions récursives, fonctions capables de s'appeler elles-mêmes et idéales pour implanter un algorithme pour tout problème auquel peut s'appliquer une résolution par récurrence, ou pour traiter des structures de données récursives (listes, arbres).

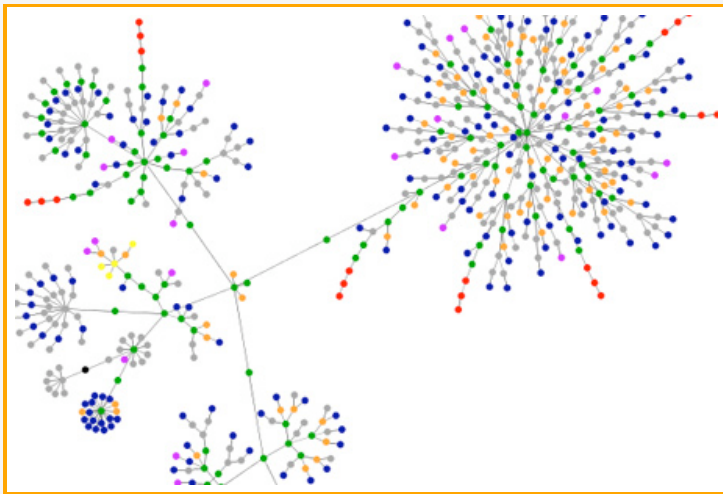


FIG. 1 – Web site graph - une représentation graphique du code HTML des pages WEB - "Libération", 23 janvier 2008

- **Lisp, Scheme** : Langages dédiés à la programmation applicative, langage de programmation universel, historiquement majeur, permettant notamment la programmation applicative. Syntaxe simple. Fort pouvoir d'expression.

- **Lisp**, 1960, John Mc Carthy : pour du calcul numérique classique et du calcul symbolique (calcul dont les données ne sont pas des nombres mais des chaînes de caractères, des mots, des collections de choses).
- **Scheme**, 1970, Gerald J. Sussman et Guy L. Steele : Héritier de LISP à liaison strictement lexicale : enseignement de la programmation.  
“Structure and Interpretation of Computer Programs - Abelson, Fano, Sussman - MIT Press - 1984”.
- Excellent préalable à l’étude des langages à objets, plus complexes.

**Caractéristiques** de Lisp et des langages applicatifs :

- Toute phrase syntaxiquement correcte du langage (y-compris une instruction) est une expression algébrique ayant une valeur. Ceci n’est pas vrai dans les langages dits ”impératifs”.
- La mémoire est gérée automatiquement (allocation et récupération) si on le souhaite (programmation sans **Effets de bords**, lien avec la programmation dite fonctionnelle - Voir langage Haskell).
- Un très grand nombre d’abstractions simplifient la vie du programmeur. Exemple : les grands nombres, les itérateurs, etc.

---

```

1 > (fact 30)
2 265252859812191058636308480000000
3 > (map fact '(2 3 4 30))
4 (2 6 24 265252859812191058636308480000000)

```

---

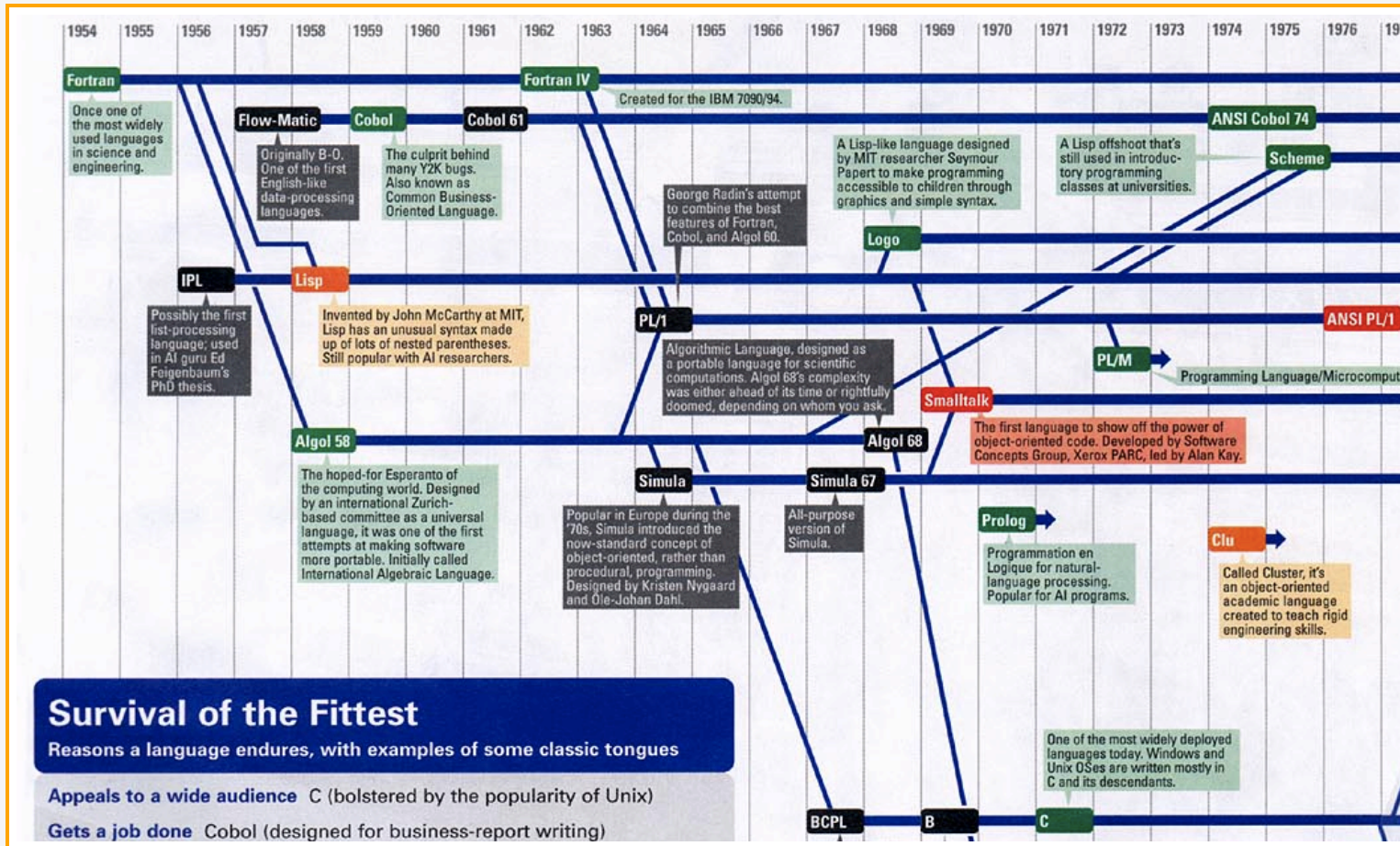


FIG. 2 – Historique des langues de programmation - extrait 1 - <http://www.digibarn.com/collections/posters/tongues/tongues.jpg>

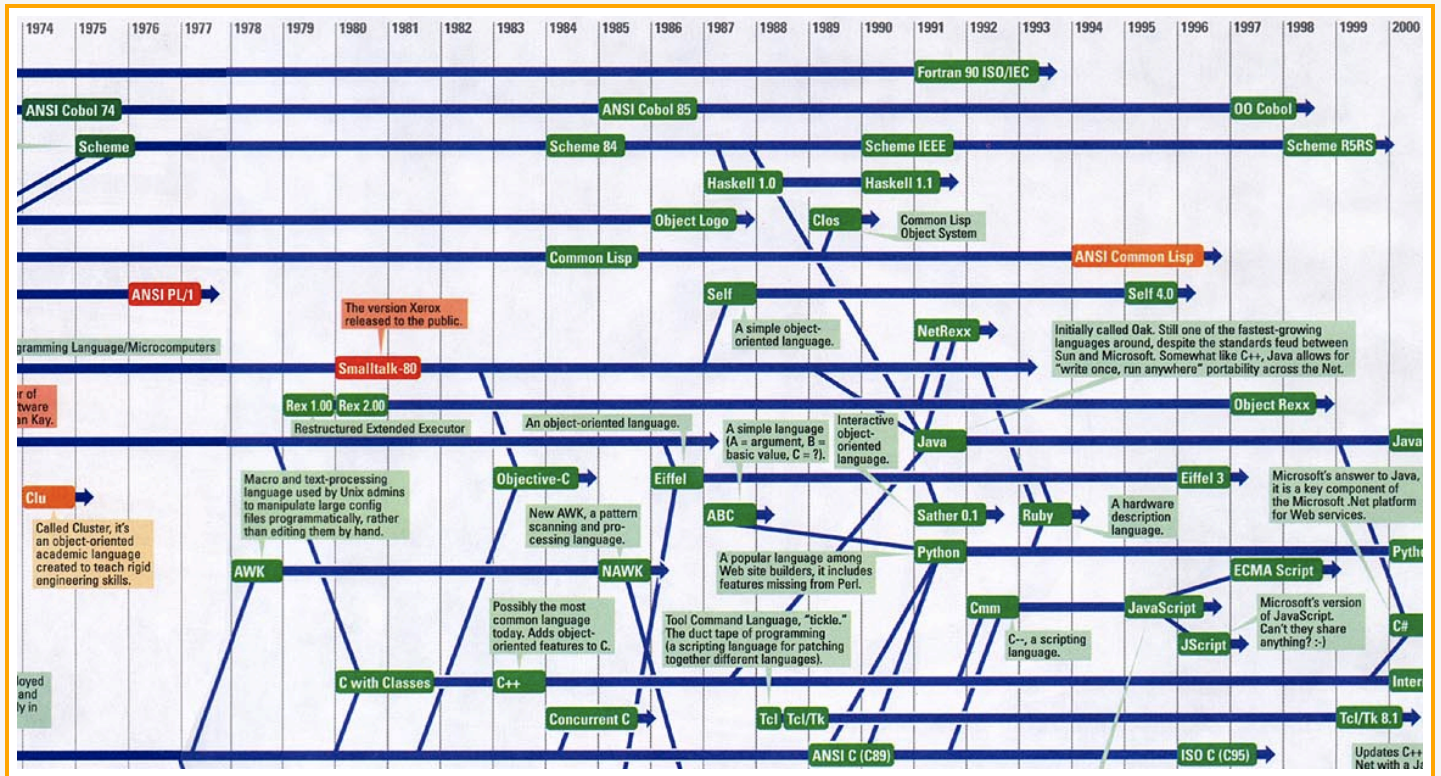


FIG. 3 – Historique des langages de programmation - extrait2- <http://www.digibarn.com/collections/posters/tongues/to>

Langage Universel. Tous les secteurs d'application vous sont ouverts (de la programmation de types récurifs à la programmation graphique ou à la réalisation de sites WEB).

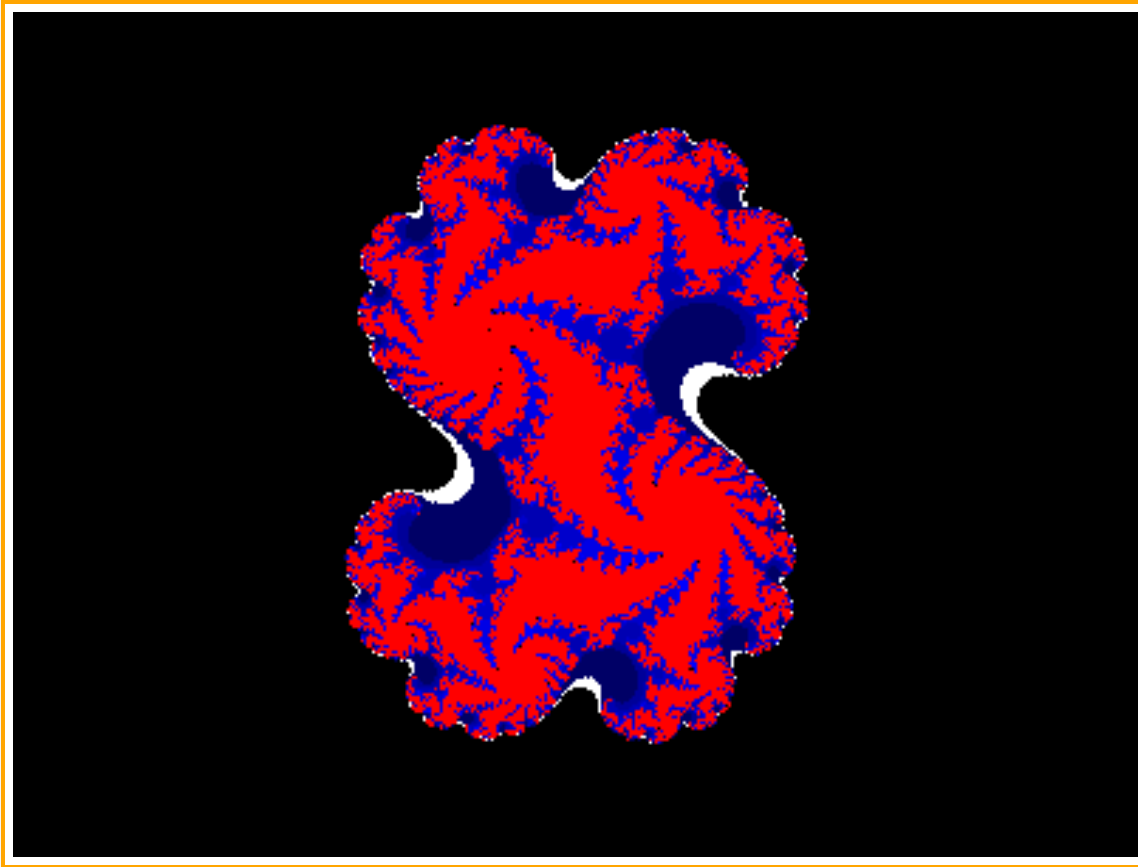


FIG. 4 – Image Fractale programmée en Scheme - Girault Geoffroy - TER L2 2008

### Contenu du cours :

- Syntaxe de *Scheme*
- Types prédéfinis. Base de l'interprétation des expressions.
- Fonctions, Identificateurs, Premières Structures de contrôle.
- Fonctions Récursives Simples.
- Listes - Symboles - Calcul Symbolique
- Fonctions récursives sur les listes. Récursions arborescentes
- Optimisation des fonctions récursives
- Abstraction de données - Arbres binaires
- Introduction à l'interprétation des langages

## 2 Rappels généraux sur la programmation

**Calcul automatisé, Calculateur** : Calculateur : L'ancêtre des ordinateurs modernes. Le terme a été abandonné, car, bien que les ordinateurs travaillent toujours en calculant, leur tâche essentielle est bien plus vaste, le calcul proprement dit étant le plus souvent caché aux yeux de l'utilisateur.

**Informatique** : De « INFORmation AutoMATIQUE ». Mot inventé en 1962 par P. Dreyfus.

L'informatique est la science du traitement automatisé de l'information. Le calcul au sens arithmétique du terme est un cas particulier de traitement de l'information.

« science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux » (académie française).

Traitement automatisé de l'information implique :

- codage de l'information pour sa représentation dans une machine.
- traitement de l'information selon une suite de calculs ou de transformations définis par un texte appelé programme

**Données** : entité manipulées dans les programmes.

**programme** : texte décrivant sous une forme interprétable par un ordinateur, une suite d'actions à réaliser sur des données.

**Ordinateur** : machine où un programme (un texte) est transformé en exécuté (transformé en actions).

**Algorithme**<sup>1</sup> : Séquence d'opérations visant à la résolution d'un problème en un temps fini (mentionner la condition d'arrêt). Fondé sur la thèse de Church.

exemple : algorithme de l'école primaire pour multiplier deux nombres.

**Processus, ou processus de calcul, ou Calcul** : Exécution d'un programme par une machine.

**Texte d'un programme** : texte, écrit dans un langage de programmation, destiné à faire exécuter des actions à un ordinateur.

**Syntaxe** : ensemble de règles de combinaison correcte des mots d'un langage pour former des phrases. (Syntaxe du français ou syntaxe de Lisp).

**Grammaire** : une représentation par intention, de la syntaxe d'un langage.

**Analyse syntaxique**, peut utiliser une grammaire, et dit si une phrase d'un langage est ou non syntaxiquement bien formée.

**Sémantique** : ensemble de règles définissant le sens d'une phrase.

Il est possible d'écrire un texte syntaxiquement correct qui n'a pas de sens selon une sémantique donnée. Exemple en français : "Quelle la différence entre un pigeon ?"

**Interpréteur** : programme transformant un texte de programme syntaxiquement correct en actions. L'interpréteur fait faire à l'ordinateur ce que dit le texte.

**Compilateur** : Traducteur de texte d'un langage L1 vers un langage L2. Par exemple, compilateur de Java en instructions de la JVM, compilateur de C en assembleur.

**Modèle de calcul** : modèle permettant de représenter, afin de comprendre, mesurer prouver, la façon dont un programme est exécuté.

Exemples de modèles de calcul : machine de Turing, lambda-calcul<sup>2</sup>

## Repères

Calcul Automatisé. Traitement automatisé de l'information. Chronologie :

- 1640 (machine à calculer de Blaise Pascal)
- Machines automatisées (Babbage).
- Machine programmable : machine qui exécute des suites d'actions, conformément à des textes appelés programmes

- Premier ordinateur : Howard H. Aiken - IBM (1943)

- Von Neumann : machine à branchements conditionnels (architecture des ordinateurs modernes).

- ...

**Quelques liens** :

\* Histoire de l'informatique par Jean-Yvon Birrien Que Sais-je n2510, Ed. P.U.F., Paris - 1990

\* Histoire de l'informatique de Serge Rossi : <http://histoire.info.online.fr/>

---

<sup>1</sup> Terme venu du XIII<sup>ème</sup> siècle, de la traduction en latin d'un mémoire de Mohammed Ibn Musa Abu Djefar Al-Khwarizmi commençant par : « Algoritmi dixit... ».

<sup>2</sup> Church-Alonso, (14/06/1903 - 11/08/1995).

- \* Dictionnaire de l'informatique et de l'Internet Ed. Micro-Application, Paris - 1998
- \* Jargon de l'informatique de Roland Trique <http://www.linux-france.org/prj/jargonf/index.html>
- \* Structure and Interpretation of Computer Programs Programs. <http://mitpress.mit.edu/sicp/>