

# Brève histoire des langages de programmation

## Table des matières

1. Avant 1950.....	2
2. Les années 1950 : Fortran, Lisp, Cobol, [Algol].....	2
3. Les années 1960 : Apl, Basic, PL/1.....	3
4. Les années 1970 : Pascal, Prolog, SmallTalk, C, Ada.....	3
5. Les années 1980 : Dbase, C++, Eiffel, Perl, Tcl/Tk.....	4
6. Les années 1990 : Java[script], Php, Mysql, VisualBasic.....	5
7. Les années 2000 : Java, Delphi, Php, Perl, C#.....	6
8. Les années 2010 : Matlab, Scilab, R, Ruby, Python et librairies Javascript.....	6
9. Les 500 prochaines années.....	7

Voici un bref rappel de l'apparition des langages les plus marquants car il y a énormément de langages ; de plus la plupart des langages ont différentes versions, variantes et dialectes étalés sur plusieurs années ce qui complique la chronologie. Il se passe en général quelques années entre les premières idées et l'implémentation, d'où parfois des dates différentes suivant les auteurs...



## 1. Avant 1950

Tout a commencé dans les années 1950 avec Fortran, Cobol et Lisp, même si il y a eu de nombreux précurseurs. Ainsi les grecs, les chinois savaient calculer, automatiser (à la main) des calculs. Nous ne retiendrons donc que deux dates importantes à nos yeux. En 820, le mathématicien El Khawarizmi a publié à Bagdad un traité intitulé "La science de l'élimination et de la réduction" qui, importé en Europe Occidentale lors des invasions arabes a eu une grande influence sur le développement des mathématiques. En 1840, Ada Lovelace (1814-1852), a défini le principe des itérations successives dans l'exécution d'une opération. En l'honneur d'El Khawarizmi elle a (probablement) nommé "algorithme" le processus logique d'exécution d'un programme.

De son "vrai" nom Augusta Ada Byron, elle était la fille du poète romantique Lord George Byron et d'une mathématicienne. Féministe, elle a épousé de William King (futur comte de Lovelace) ce qui lui vaudra d'être connue dans l'histoire comme **Lady Lovelace** plutôt que comme Ada King. Elle avait été l'élève de Charles Babbage avant de devenir sa collaboratrice. Elle est aussi à l'origine du "Principe des machines à calculer". Pour elle, une machine à calculer devait comporter :



- un dispositif permettant d'introduire les données numériques (cartes perforées, roues dentées...),
- une mémoire pour conserver les valeurs numériques entrées,
- une unité de commande grâce à laquelle l'utilisateur va indiquer à la machine les tâches à effectuer,
- un "moulin" chargé d'effectuer les calculs,
- un dispositif permettant de prendre connaissance des résultats (imprimante...).

Ces principes seront, un siècle plus tard, à la base des premiers ordinateurs.

## 2. Les années 1950 : Fortran, Lisp, Cobol, [Algol]

En 1950, l'invention de l'assembleur par Maurice V. Wilkes de l'université de Cambridge ouvre la voie aux langages dits "de haut niveau". Avant, la programmation s'effectuait directement en binaire. Grace Murray Hopper (1906-1992), une américaine, mobilisée comme auxiliaire dans la marine américaine, développe pour Remington Rand le tout premier compilateur, nommé A0. Il permet de générer un programme binaire à partir d'un "code source".

Alors que le langage Fortran commence à apparaître vers 1955 notamment chez IBM, **Grace Hopper** s'intéresse aux langages qui utilisent des mots voire des expressions du "langage naturel". Après B0 et Flowmatic, elle participe, dès 1959, à l'élaboration de ce qui deviendra le langage Cobol. Dans le même temps (soit vers 1958) John Mc Carthy, mathématicien au MIT qui y a fondé en 1957 le département d'Intelligence Artificielle, crée le langage Lisp.



Cette même année 1958, selon Allen Tucker "une commission de représentants du GAMM (organisme européen regroupant des chercheurs en informatique) et de l'ACM (son équivalent américain) se réunit à Zurich et rédigea un rapport préliminaire sur un <<langage algébrique international>> nommé LAI ou encore Algol58.

Avec la fin des années 50 s'achève ce qu'on nomme aujourd'hui l'ère des ordinateurs de première génération qui utilisent principalement les cartes perforées. La seconde génération, celle qui utilise les transistors va prendre pleinement son essor.

### 3. Les années 1960 : Apl, Basic, PL/1

Fortran et Cobol s'installent comme langages principaux : pendant 20 ans Cobol sera le langage le plus utilisé au monde. Aujourd'hui, il détient toujours le record du grand nombre de lignes écrites dans un langage de programmation. Par contre, Lisp reste cantonné à la communauté de l'intelligence artificielle. Si Algol devient un langage pour la publication d'algorithmes, il sera peu implémenté. Modifié en Algol W puis en Algol68, il ne parvient pas à devenir le langage standard universel qu'il aurait dû être sans doute parce que trop complet pour l'époque et trop difficile à implémenter pour ces années 60.

En 1962, le mathématicien canadien Kenneth Iverson crée un système de notation pour les mathématiques et qui deviendra grâce à IBM le langage de programmation APL qui est très original car il ne comporte pas de mots mais des symboles spéciaux. Il nécessite un clavier spécial, ce qui freinera beaucoup sa diffusion. Bien que répandu et encore utilisé, il restera toujours à part dans le monde de la programmation. Révolution, il existe un ordinateur transportable (25 kg !) nommé APL5100 qui démarre sous APL là où les autres langages utilisent des cartes perforées utilisables via des machines perforatrices sans écran.

**Thomas Kurtz et John Kemeny** créent en 1964 un langage au Dartmouth College pour leurs étudiants. ce langage pour "débutants" se nomme avec un jeu de mots BASIC. Pendant ce temps, constatant les faiblesses de Cobol et Fortran, et pour assurer sa suprématie dans le monde de la programmation IBM crée le langage de programmation PL/1.



### 4. Les années 1970 : Pascal, Prolog, SmallTalk, C, Ada

Depuis 1968 **Niklaus WIRTH** développe Algol. A force de le modifier, il finit par mettre au point un successeur d'Algol nommé le langage PASCAL. Ce langage, moins puissant qu'Algol, est bien structuré, très lisible, très "coercitif" et se trouve donc bien adapté à l'enseignement de la programmation. Il va être pendant une quinzaine d'années le langage préféré de nombreux enseignants en informatique.

Dès 1970, Ken Thompson, pensant que le système d'exploitation UNIX ne serait pas complet sans un langage de programmation de haut niveau commence à porter le Fortran sur le PDP 7 mais change rapidement



d'avis et crée en fait un nouveau langage, le B (en référence au BCPL dont il s'inspire). 2 ans plus tard, Dennis Ritchie du Bell Lab d'ATT reprend ce langage B pour mieux l'adapter au PDP/11 sur lequel UNIX vient juste d'être porté. Il fait évoluer le langage et le dote d'un vrai compilateur générant du code machine PDP/11 : c'est la naissance du langage C. En 1978, Brian Kernighan écrit un livre au sujet de la programmation en langage C devenu LA référence au point que ce livre est surtout connu sous le nom : "Le Kernighan & Ritchie".



En 1970 aussi, la DOD (Département de la défense américaine) trouve qu'il y a beaucoup trop d'assembleur dans les systèmes embarqués et aimerait utiliser un "bon" langage de programmation. En 1975, une groupe d'études de penche sur 23 langages déjà existant. La conclusion de l'analyse sera qu'il faudra un nouveau langage, sans doute basé à la fois sur Pascal, Algol et Pl/1. Après un appel d'offre et plusieurs sélections, c'est finalement en 1979 le projet de Jean Ichbiach, avec l'équipe d'Honeywell Bull de Marseille qui sera retenu. Ce langage sera nommé Dod-1 puis Ada. Ce choix par le Pentagone Américain comme l'unique langage de développement imposé à ses services à la place de la jungle des centaines de langages et dialectes aurait du en faire le premier langage de développement au monde mais la difficulté à maîtriser Ada en fera un "outsider".

Toujours au début des années 1970, Philippe Roussel et Alain Colmerauer dans leur Groupe d'Intelligence Artificielle de Marseille développent un langage qui fonctionne de façon totalement différente des autres langages : on y programme logiquement c'est à dire en spécifiant seulement des relations logiques : il se nomme Prolog et devient le langage privilégié pour l'Intelligence Artificielle.

L'année 1972 passera inaperçue ; pourtant c'est l'année de la création du premier langage totalement objet, SmallTalk par Alan Kay au Xerox PARC. Révolutionnaire, il utilise une souris et comporte un environnement de développement :

## 5. Les années 1980 : Dbase, C++, Eiffel, Perl, Tcl/Tk

Le début des années 80 consacre le développement de la petite informatique et de la micro-informatique : on y voit naître les premiers PC et les premiers Apple (mais Windows n'existe pas encore). Dbase va s'imposer dans ces années 80 comme "le" gestionnaire de "bases de données relationnelles" lorsque l'approche tableur (Multiplan, Visicalc...) n'est pas adaptée.

Dbase est facile à utiliser, il propose des menus, des masques de saisie ce que ne fournit pas à l'époque Pascal (Turbo ou pas..) ni C, ni Fortran. De plus, Dbase permet aux entreprises de se libérer de la gestion à distance via les infocentres...

Pourtant, la révolution objet est en marche ; elle permet d'écrire de plus gros programmes mieux structurés, plus facilement modifiables et plus surs. En 1983 Bjarne Stroustrup développe une extension orientée objet au langage C qui deviendra le langage C++ alors que

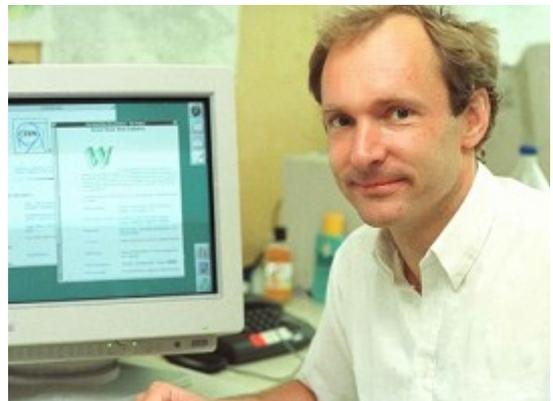


Bertrand Meyer conçoit un langage complètement objet nommé Eiffel pour lequel un compilateur sort dès 1986. Il faudra pourtant quelques années avant que les "objets" deviennent prépondérants en programmation...

Cette même année 1986, Larry Wall qui est un programmeur système mais aussi un linguiste décide de parfaire les outils Unix de traitement d'informations texte dont sed et le langage Awk : le langage Perl vient de naître. Grâce aux Web et à des scripts (programmes) parfois très courts, Perl devient un outil indispensable pour gérer les fichiers-textes, les affichages des commandes des systèmes d'exploitation.

Vers la fin des années 80, les langages de commandes et de script se développent pour tous les types d'ordinateurs. Parallèlement, la notion d'interface graphique pour utilisateur (GUI) commence à entrer dans les mœurs pour les "grands systèmes" ; John Osterout invente à l'Université de Californie (Berkeley) en 1988 les langages Tcl et Tk pour des "développements rapides" : Tcl est la partie "scripts de commandes" dont Tk produit l'interface. Le langage est connu depuis sous le nom de Tcl/Tk.

Enfin, même s'il n'est pas vraiment considéré comme un langage de programmation, le langage HTML (Hypertext Markup Language) est développé en 1989 par **Tim Berners-Lee**. Ce sera "LE" langage du Web (diapositive aimablement fournie par D. Genest du Laboratoire LERIA, Angers).



## 6. Les années 1990 : Java[script], Php, Mysql, VisualBasic

Les années 90 voient s'installer un peu partout dans le monde un produit logiciel qui révolutionne Dos : c'est Windows. C'est en 1990 que Microsoft sort son produit Windows 3.0 qui est une version complètement revue des premiers "Microsoft Windows". Un an plus tard, mais sans publicité, Linux 0.01 est annoncé par un étudiant, Linus Torvald à l'Université d'Helsinki. Linux va se développer très rapidement grâce à Internet et grâce à deux concepts-phare : la disponibilité du code-source des programmes et la gratuité, suivant en cela le projet GNU de by Richard Stallman, le fameux créateur de l'éditeur emacs. Dans le même temps, les laboratoires de Sun étoffent Tcl pour en faire un "langage de script universel adapté à Internet" et le portent pour Windows et Macintosh. Tk va devenir une "sur-couche" à de nombreux langages.

Le développement très fort d'Internet influence fortement les concepteurs de langage. En 1995, suite à de nombreuses réunions de comité du WWW, le langage LiveScript est renommé en Javascript et est considéré comme une "bonne" solution pour gérer dynamiquement les pages Web. Il est aussitôt incorporé dans Netscape 2. Mais il manque toujours un langage complet, objet, capable de dialoguer avec les serveurs Web et les bases de données. Dans la même année 95, Java est introduit comme langage de développement objet "multi-OS" pour combler ce manque.



La gestion des formulaires et des bases de données accessibles par le Web voit apparaître pratiquement en même temps le langage Php, souvent couplé au langage de base de données Sql

notamment dans ses implémentations Mysql et PosgresSql.

Parallèlement, la domination de Windows comme système d'exploitation pour le grand public avec ses deux logiciels phares Word et Excel induit progressivement l'utilisation de "macros" pour tirer pleinement profit des possibilités de ces logiciels : le langage Basic, remanié, rendu objet avec ses fonctions liées aux documents devient VBA c'est à dire "Visual Basic for Applications".

## 7. Les années 2000 : Java, Delphi, Php, Perl, C#

Les années 2000 ne voient pas apparaître de nouveau langage marquant. La tendance est plutôt à l'amélioration et à l'enrichissement des langages présents. L'omniprésence d'Internet force les langages développés pour le Web à progresser encore : un programme (ou "application" en anglais) pour le Web devient une "applet", une "weblet" ou quelque chose en "let" ainsi un script général imbriqué dans une page Web est un "scriptlet". Programmé en Tcl, c'est une "tcl-let". Conçue comme sous le mode client-serveur, c'est une "servlet". Pour répondre à ces attentes, les langages accumulent version sur version jusqu'à fournir un produit "stable" et "complet".

Il faut noter que Java 1.5 est une version très mûre, très stable de Java, enrichie de nouvelles bibliothèques de sous-programmes (plus précisément de "classes"), disponible pour tout système d'exploitation : Unix, MacOS, Dos et Windows et qui devient une référence pour la programmation et le développement de tous types d'applications.

De même Delphi, le successeur de Pascal, TurboPascal et Pascal Windows propose une interface complète de développement dit "intégré" (IDE) sous Windows et un frère jumeau sous GNU/Linux : Kylix. L'écriture d'interfaces utilisateur avec des boutons, des menus, des fenêtres multi documents n' a jamais été aussi facile.

Il faut toutefois signaler une mini-révolution dans la façon de penser de penser les langages, due à Microsoft. Devant la diversité des systèmes d'exploitation et de façon à réutiliser le même programme sur tous les ordinateurs et sur tous les systèmes dans tous les modes (local, distant) Bill Gates propose de fonctionner en mode .net ou encore # : tout programme doit être traduit en un code intermédiaire exécutable par un programme spécial, sur l'ordinateur de l'utilisateur ou sur celui du serveur Web. Les logiciels classiques, comme Word et Excel pourraient même être utilisés à partir de serveurs de logiciels, plutôt que d'être installés sur l'ordinateur...



## 8. Les années 2010 : Matlab, Scilab, R, Ruby, Python et bibliothèques Javascript

Les langages de calcul scientifique (par opposition au langages de calcul formel) que sont Matlab (payant) et Scilab (libre) acquièrent dans cette décennie leur maturité : les bibliothèques de programmes additionnels nommées tools-box leur permettent de s'insérer dans les cursus d'ingénieurs et de programmer "tous azimuts" en science, notamment en physique, chimie et biologie.

Parallèlement, le logiciel R s'impose comme LE grand logiciel de statistiques : complet, avec de jolis graphiques, il est le seul à s'enrichir rapidement et quotidiennement (!) de nombreuses bibliothèques de programmes additionnels (nommées packages) en particulier pour la bio informatique via le projet Bioconductor.

Les pages Web prenant de plus en plus d'ampleur, les langages de script Ruby et Python deviennent, via leurs "frameworks de développement" phares respectifs, nommés Rails et Django, des langages très adaptés aux méthodes AGILES et RAD même si le langage PHP reste un grand classique pour le développement de sites Web. L'importance de "grands" sites Web comme Facebook, Flickr... qui ne peuvent pas fonctionner sans Javascript font de Javascript et ses bibliothèques comme JQuery, Prototype... le langage incontournable du développement Web.

L'apparition de bibliothèques et d'environnements scientifiques comme NumPy, SciPy, Pandas, Anaconda pour Python, de packages et de données bio informatiques pour R et de serveurs avec des I/O non bloquantes via Node.js font de Python, R et Javascript les grands "gagnants" de ces années 2010.

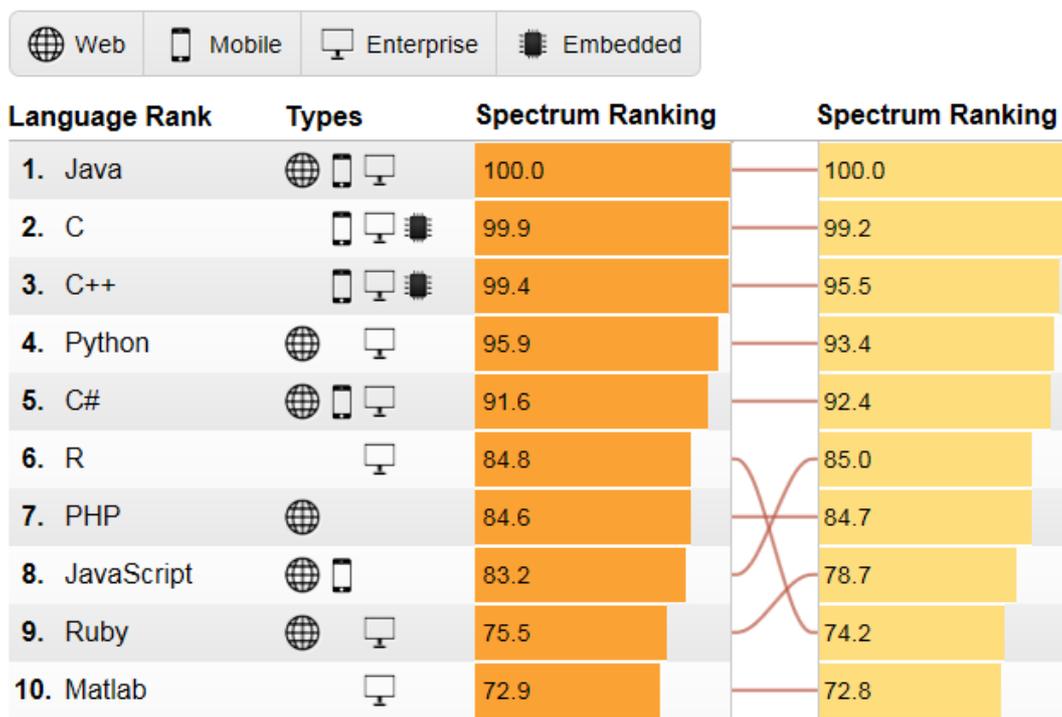
## 9. Les 500 prochaines années

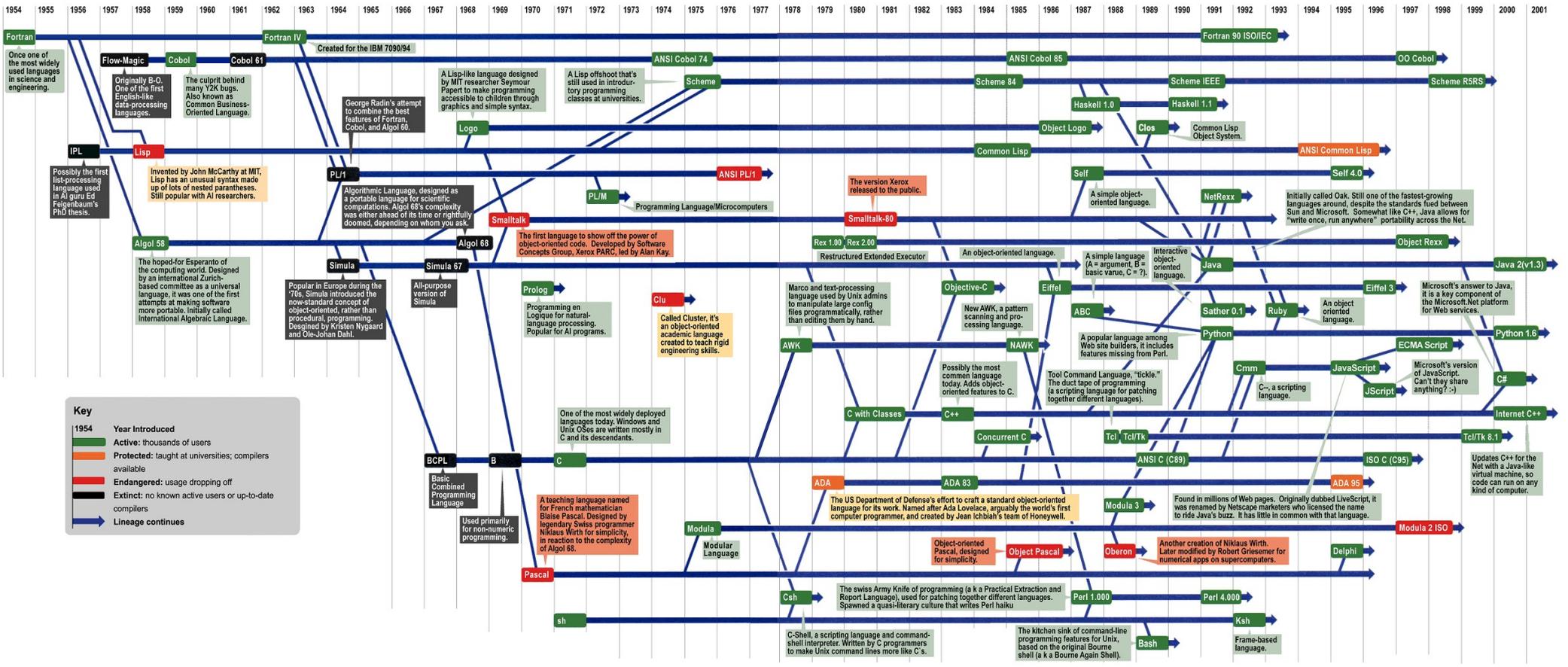
Il est impossible de prévoir ce que seront les prochaines années... La technologie en général et donc l'informatique et les langages de programmation en particulier évoluent suffisamment vite pour qu'il y ait encore des surprises.

Toutefois, on peut remarquer que :

- C++ et Java sont "installés" comme langages de grosses applications,
- Perl, Php, Mysql sont très utilisés pour la programmation Web,
- tous les langages ne fonctionnent pas forcément sur tous les ordinateurs,
- le diction "à objectif précis, langage imposé" reste encore valide,

Comme l'histoire des langages de programmation est liée à celle de l'informatique, de la technologie et même à celle de la téléphonie, on verra peut-être des changements au niveau de l'interface, il sera possible de commander des ordinateurs "à la voix" en utilisant le "langage naturel" c'est à dire en parlant presque comme vous et moi.





**Fortran** (1954) - Once one of the most widely used languages in science and engineering. Fortran 90 ISO/IEC.

**Flow-Magic** (1956)

**Cobol** (1957) - Originally B-O. One of the first English-like data-processing languages. The culprit behind many Y2K bugs. Also known as Common Business-Oriented Language. Cobol 61, Cobol 68.

**IPL** (1958) - Possibly the first list-processing language used in AI guru Ed Feigenbaum's PhD thesis.

**Lisp** (1958) - Invented by John McCarthy at MIT. Lisp has an unusual syntax made up of lots of nested parentheses. Still popular with AI researchers.

**PL/1** (1962) - Algorithmic Language, designed as a portable language for scientific computations. Algol 68's complexity was either ahead of its time or rightfully doomed, depending on whom you ask.

**Logo** (1967) - A Lisp-like language designed by MIT researcher Seymour Papert to make programming accessible to children through graphics and simple syntax.

**ANSI Cobol 74** (1974) - A Lisp offshoot that's still used in introductory programming classes at universities.

**Scheme** (1975) - A Lisp offshoot that's still used in introductory programming classes at universities.

**ANSI Cobol 85** (1985)

**ANSI Cobol 88** (1988)

**OO Cobol** (1990)

**Scheme 84** (1984)

**Scheme IEEE** (1990)

**Scheme R5RS** (1995)

**Haskell 1.0** (1990)

**Haskell 1.1** (1992)

**Object Logo** (1990)

**Clos** (1990)

**Common Lisp Object System** (1990)

**ANSI Common Lisp** (1990)

**Self** (1990)

**Self 4.0** (1990)

**NetRexx** (1990)

**Initially called Oak. Still one of the fastest-growing languages around, despite the standards feud between Sun and Microsoft. Somewhat like C++, Java allows for "write once, run anywhere" portability across the Net.**

**Smalltalk** (1973) - The version Xerox released to the public. Smalltalk-80.

**PL/M** (1973) - Programming Language/Microcomputers.

**ANSI PL/1** (1973)

**Rex 1.00** (1973)

**Rex 2.00** (1973)

**Restructured Extended Executor** (1973)

**Objective-C** (1988) - An object-oriented language.

**Eiffel** (1988) - A simple language (A = argument, B = basic var, C = ?).

**Interactive object-oriented language** (1988)

**Java** (1995) - A simple language (A = argument, B = basic var, C = ?). Initially called Oak. Still one of the fastest-growing languages around, despite the standards feud between Sun and Microsoft. Somewhat like C++, Java allows for "write once, run anywhere" portability across the Net.

**Java 2(v1.3)** (1998)

**Object Rexx** (1990)

**Simula** (1962) - Popular in Europe during the '70s, Simula introduced the now-standard concept of object-oriented, rather than procedural, programming. Designed by Kristen Nygaard and Ole-Johan Dahl.

**Simula 67** (1967)

**All-purpose version of Simula** (1967)

**Prolog** (1972) - Programming on Logique for natural-language processing. Popular for AI programs.

**CLU** (1973) - Called Cluster, it's an object-oriented academic language created to teach rigid engineering skills.

**AWK** (1977) - Marco and text-processing language used by Unix admins to manipulate large config files programmatically, rather than editing them by hand.

**Objective-C** (1988) - An object-oriented language.

**Eiffel** (1988) - A simple language (A = argument, B = basic var, C = ?).

**ABC** (1988) - A popular language among Web site builders, it includes features missing from Perl.

**Sather 0.1** (1988) - An object oriented language.

**Python** (1990) - A popular language among Web site builders, it includes features missing from Perl.

**Ruby** (1995) - An object oriented language.

**ECMA Script** (1996) - Microsoft's answer to Java. It is a key component of the Microsoft .Net platform for Web services.

**Python 1.6** (1996)

**JavaScript** (1996) - Microsoft's version of JavaScript. Can't they share anything? :-)

**C#** (2002) - Microsoft's answer to Java. It is a key component of the Microsoft .Net platform for Web services.

**Internet C++** (1996) - Updates C++ for the Net with a Java-like virtual machine, so code can run on any kind of computer.

**BCPL** (1967) - Basic Combined Programming Language.

**B** (1970) - Used primarily for non-numeric programming.

**C** (1972) - One of the most widely deployed languages today. Windows and Unix OSes are written mostly in C and its descendants.

**Modula** (1978) - A teaching language named for French mathematician Blaise Pascal. Designed by legendary Swiss programmer Niklaus Wirth for simplicity, in reaction to the complexity of Algol 68.

**Modular Language** (1978)

**ADA** (1979) - The US Department of Defense's effort to craft a standard object-oriented language for its work. Named after Ada Lovelace, arguably the world's first computer programmer, and created by Jean Ichbiah's team of Honeywell.

**ADA 83** (1983)

**ADA 95** (1995)

**Modula 3** (1988) - Found in millions of Web pages. Originally dubbed LiveScript, it was renamed by Netscape marketers who licensed the name to ride Java's buzz. It has little in common with that language.

**Object Pascal** (1988) - Object-oriented Pascal, designed for simplicity.

**Object Pascal** (1988) - Object-oriented Pascal, designed for simplicity.

**Oberon** (1988) - Another creation of Niklaus Wirth. Later modified by Robert Griesemer for numerical apps on supercomputers.

**Delphi** (1995) - Another creation of Niklaus Wirth. Later modified by Robert Griesemer for numerical apps on supercomputers.

**Pascal** (1970) - A teaching language named for French mathematician Blaise Pascal. Designed by legendary Swiss programmer Niklaus Wirth for simplicity, in reaction to the complexity of Algol 68.

**Csh** (1979) - The swiss Army Knife of programming (a.k.a. Practical Extraction and Report Language), used for patching together different languages. Spawned a quasi-literary culture that writes Perl haiku.

**Perl 1.000** (1987)

**Perl 4.000** (1997)

**C-Shell, a scripting language and command-shell interpreter. Written by C programmers to make Unix command lines more like C's.**

**sh** (1979)

**Csh** (1979) - The swiss Army Knife of programming (a.k.a. Practical Extraction and Report Language), used for patching together different languages. Spawned a quasi-literary culture that writes Perl haiku.

**Perl 1.000** (1987)

**Perl 4.000** (1997)

**Ksh** (1991) - The kitchen sink of command-line programming features for Unix, based on the original Bourne shell (a.k.a. Bourne Again Shell).

**Bash** (1989) - The kitchen sink of command-line programming features for Unix, based on the original Bourne shell (a.k.a. Bourne Again Shell).

**Frame-based language** (1991)