

GRANDES FAMILLES DE LANGAGE DE PROGRAMMATION

● ● ●
Présenté par :
Yamna MOSTAPHI
Kawthar ASSERRAR

Encadré par :
Mme Khaoula BENMOUSSA



TABLE DE MATIERES



01

INTRODUCTION

02

HISTOIRE DES LANGAGES
DE PROGRAMMATION

03

TYPES DE LANGAGES

04

TYPES DE PROGRAMMATION

Introduction



- ◆ Un **langage de programmation** est un langage informatique, permettant à un être humain d'écrire un code source qui sera analysé par une machine, généralement un ordinateur. Le code source subit ensuite une transformation ou une évaluation dans une forme exploitable par la machine, ce qui permet d'obtenir un programme. Les langages permettent souvent de faire abstraction des mécanismes de bas niveau de la machine, de sorte que le code source représentant une solution puisse être écrit et compris par un être humain. L'activité de rédaction du code source d'un programme est nommée programmation. Elle consiste en la mise en œuvre de techniques d'écriture et de résolution d'algorithmes informatiques, lesquelles sont fondées sur les mathématiques.

IER LANGAGE DE PROGRAMMATION

Saviez-vous que le premier langage de programmation a été inventé en 1843 ? **Ada Lovelace** a mis au point le tout premier algorithme pour une machine informatique qu'elle a noté sur un morceau de papier parce qu'aucun ordinateur n'existait à l'époque !

Les langages de programmation ont évidemment fait beaucoup de chemin depuis lors, mais pour comprendre l'histoire des langages de programmation, il faut d'abord en reconnaître l'origine.



HISTOIRE DES LANGAGES DE PROGRAMMATION : CHRONOLOGIE



Les premiers langages de programmation connus étaient des codes machine compliqués qui étaient entrés manuellement dans les premières machines informatiques. Comme vous le découvrirez, la programmation informatique s'est rapidement développée, passant des codes machine à un code entièrement automatisé lisible par l'homme.



Le premier langage binaire (Joseph Marie Jacquard 1801)



La première programmation binaire (carton/trou) à Joseph-Marie Jacquard en 1801. Il s'agissait d'un procédé industriel visant à accroître la productivité des métiers utilisait des trous dans des cartes perforées pour représenter les mouvements du bras du métier à tisser, et ainsi générer automatiquement des motifs décoratifs.





1843 : L'ALGORITHME DE LA MACHINE D'ADA LOVELACE

Ada Lovelace a inventé le tout premier algorithme machine pour la Machine des différences de Charles Babbage qui pose les bases de tous les langages de programmation.



Le premier programme sans ordinateur (Ada Lovelace 1842)





- ◆ En 1840, Ada Lovelace (1814-1852) parvient à définir le principe d'itérations successives dans l'exécution d'une seule opération. Elle a probablement créé le mot algorithme en l'honneur d'Al Khawarizmi (780-850), pour désigner le processus logique d'exécution d'un programme.
- ◆ Al Khawarizmi était un mathématicien du califat abasside (Ouzbèke), célèbre pour avoir introduit l'algèbre en Europe et notamment pour avoir classifié les algorithmes connus de son époque en fonction de leur terminaison. Ada Lovelace (son vrai nom étant Augusta Ada Byron) traduit en 1842-1843 le mémoire d'un mathématicien Italien sur la machine analytique de l'Anglais Charles Babbage (1791-1871).
- ◆ Les calculs de la machine reposent alors sur des cartes perforées, sur lesquelles de l'information est encodée.
- ◆ Elle y détaille une méthode pour calculer les nombres de Bernoulli sur ces cartes perforées, **ce qui sera plus tard considéré comme le tout premier programme informatique du monde.**

Programme » de calcul des nombres de Bernoulli dans la note G d'Ada Lovelace (1843)

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.										Working Variables.												Result Variables.			
						$1V_1$	$1V_2$	$1V_3$	$1V_4$	$1V_5$	$1V_6$	$1V_7$	$1V_8$	$1V_9$	$1V_{10}$	$1V_{11}$	$1V_{12}$	$1V_{13}$	$1V_{14}$	$1V_{15}$	$1V_{16}$	$1V_{17}$	$1V_{18}$	$1V_{19}$	$1V_{20}$	$1V_{21}$	$1V_{22}$	$1V_{23}$	$1V_{24}$		
						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						1	2	n																							
1	×	$1V_2 \times 1V_3$	$1V_4, 1V_5, 1V_6$	$1V_2 = 1V_2$ $1V_3 = 1V_3$ $1V_4 = 1V_4$	$-2n$...	2	n	2n	2n	2n																				
2	-	$1V_4 - 1V_5$	$1V_6$	$1V_4 = 1V_4$ $1V_5 = 1V_5$ $1V_6 = 1V_6$	$-2n-1$	1	...		2n-1																						
3	+	$1V_5 + 1V_6$	$1V_7$	$1V_5 = 1V_5$ $1V_6 = 1V_6$ $1V_7 = 1V_7$	$-2n+1$	1	...		2n+1																						
4	+	$1V_6 + 1V_7$	$1V_{11}$	$1V_6 = 1V_6$ $1V_7 = 1V_7$ $1V_{11} = 1V_{11}$	$-2n-1$ $-2n+1$	1	...		0	0																					
5	+	$1V_{11} + 1V_7$	$1V_{12}$	$1V_{11} = 1V_{11}$ $1V_7 = 1V_7$ $1V_{12} = 1V_{12}$	-1 $-2n+1$...	2																								
6	-	$1V_{12} - 1V_{11}$	$1V_{13}$	$1V_{12} = 1V_{12}$ $1V_{11} = 1V_{11}$ $1V_{13} = 1V_{13}$	-1 $-2n+1 = A_0$...																									
7	-	$1V_8 - 1V_9$	$1V_{10}$	$1V_8 = 1V_8$ $1V_9 = 1V_9$ $1V_{10} = 1V_{10}$	$n-1 (=3)$	1	...	n																							
8	+	$1V_9 + 1V_{10}$	$1V_7$	$1V_9 = 1V_9$ $1V_{10} = 1V_{10}$ $1V_7 = 1V_7$	$-2+0=2$...	2																								
9	+	$1V_{10} + 1V_7$	$1V_{11}$	$1V_{10} = 1V_{10}$ $1V_7 = 1V_7$ $1V_{11} = 1V_{11}$	$-2n = A_1$...																									
10	×	$1V_{11} \times 1V_{11}$	$1V_{12}$	$1V_{11} = 1V_{11}$ $1V_{12} = 1V_{12}$	$-B_1 \cdot \frac{2n-1}{2} = B_1 A_1$...																									
11	+	$1V_{12} + 1V_{12}$	$1V_{13}$	$1V_{12} = 1V_{12}$ $1V_{13} = 1V_{13}$	-1 $-2n+1 + B_1 \cdot \frac{2n}{2}$...																									
12	+	$1V_{13} + 1V_8$	$1V_{10}$	$1V_{13} = 1V_{13}$ $1V_8 = 1V_8$ $1V_{10} = 1V_{10}$	$n-2 (=2)$	1	...																								
13	-	$1V_6 - 1V_7$	$1V_6$	$1V_6 = 1V_6$ $1V_7 = 1V_7$ $1V_6 = 1V_6$	$-2n-1$	1	...																								
14	+	$1V_7 + 1V_7$	$1V_7$	$1V_7 = 1V_7$ $1V_7 = 1V_7$ $1V_7 = 1V_7$	$-2+1=3$	1	...																								
15	+	$1V_7 + 1V_7$	$1V_6$	$1V_7 = 1V_7$ $1V_7 = 1V_7$ $1V_6 = 1V_6$	$2n-1$...																									
16	×	$1V_8 \times 1V_{11}$	$1V_{12}$	$1V_8 = 1V_8$ $1V_{11} = 1V_{11}$ $1V_{12} = 1V_{12}$	$\frac{2n-1}{2} \cdot \frac{2n-1}{3}$	1	...																								
17	-	$1V_6 - 1V_7$	$1V_6$	$1V_6 = 1V_6$ $1V_7 = 1V_7$ $1V_6 = 1V_6$	$-2n-2$	1	...																								
18	+	$1V_7 + 1V_7$	$1V_7$	$1V_7 = 1V_7$ $1V_7 = 1V_7$ $1V_7 = 1V_7$	$-3+1=4$	1	...																								
19	+	$1V_7 + 1V_7$	$1V_9$	$1V_7 = 1V_7$ $1V_7 = 1V_7$ $1V_9 = 1V_9$	$2n-2$...																									
20	×	$1V_9 \times 1V_{11}$	$1V_{12}$	$1V_9 = 1V_9$ $1V_{11} = 1V_{11}$ $1V_{12} = 1V_{12}$	$\frac{2n-1}{3} \cdot \frac{2n-2}{3} = A_2$...																									
21	+	$1V_{12} \times 1V_{11}$	$1V_{12}$	$1V_{12} = 1V_{12}$ $1V_{11} = 1V_{11}$ $1V_{12} = 1V_{12}$	$B_2 \cdot \frac{2n-1}{2} \cdot \frac{2n-2}{3} = B_2 A_2$...																									
22	+	$1V_{12} + 1V_{12}$	$1V_{12}$	$1V_{12} = 1V_{12}$ $1V_{12} = 1V_{12}$ $1V_{12} = 1V_{12}$	$-A_0 + B_1 A_1 + B_2 A_2$...																									
23	-	$1V_{10} - 1V_7$	$1V_{10}$	$1V_{10} = 1V_{10}$ $1V_7 = 1V_7$ $1V_{10} = 1V_{10}$	$n-3 (=1)$	1	...																								
Here follows a repetition of Operations thirteen to twenty-three.																															
24	+	$1V_{12} + 1V_{12}$	$1V_{24}$	$1V_{12} = 1V_{12}$ $1V_{12} = 1V_{12}$ $1V_{24} = 1V_{24}$	$= B_7$...																									
25	+	$1V_1 + 1V_7$	$1V_8$	$1V_1 = 1V_1$ $1V_7 = 1V_7$ $1V_8 = 1V_8$	$n+1 = 4+1=5$ by a Variable-card.	1	...	n+1																							



Ada Lovelace

**La visionnaire de
l'informatique**



- ◆ On pense aujourd'hui qu'Ada Lovelace avait compris toutes les potentialités de la machine analytique mieux que Babbage et qu'elle avait prévu que la machine pouvait manipuler non seulement des nombres, mais aussi des symboles. Si le mathématicien a voulu créer une calculatrice améliorée, la jeune femme est beaucoup plus visionnaire et imagine déjà une sorte de calculateur universel. Dans ces notes, elle écrit que ces calculateurs peuvent être des *"partenaires de l'imagination"* en programmant musique, poésie ou peinture : « *la machine pourrait composer de manière scientifique et élaborée des morceaux de musique de n'importe quelle longueur ou degré de complexité* ». En cela, très en avance sur son temps, elle imagine déjà une sorte d'intelligence artificielle.

A black and white photograph showing two men in suits and glasses. The man on the left, Konrad Zuse, is looking down at a document held by the man on the right. The background is dark and out of focus.

1944-45 : PLANKALKUL

Entre 1944 et 1945, **Konrad Zuse** a développé le premier « **vrai** » langage de programmation appelé **Plankalkül** (Plan de calcul). Le **langage de Zeus** permettait (entre autres choses) de créer des **procédures**, qui stockaient des **morceaux de code** pouvant être invoqués à plusieurs reprises pour effectuer des opérations de routine.

Langage assembleur (1949)



- ◆ Un **langage d'assemblage** ou **langage assembleur** est, en programmation informatique, le langage de plus bas niveau qui représente le langage machine sous une forme lisible par un humain. Les combinaisons de bits du langage machine sont représentées par des symboles dits « mnémoniques », c'est-à-dire faciles à retenir. Le programme assembleur convertit ces mnémoniques en langage machine, ainsi que les valeurs (écrites en décimal) en binaire et les libellés d'emplacements en adresses, en vue de créer par exemple un fichier objet ou un fichier exécutable.
- ◆ Les programmes de l'EDSAC (1949), premier ordinateur à programmes enregistrés, étaient rédigés en utilisant des mnémoniques alphabétiques d'une lettre pour chaque instruction. La traduction était alors faite à la main par les programmeurs, une opération longue, fastidieuse et entachée d'erreurs.

- ◆ Le premier programme assembleur a été écrit par **Nathaniel Rochester** pour l'**IBM 701** (le premier ordinateur commercialisé par IBM) en 1954.



IBM 701

```
org 100h
mov dx, message
mov ah, 09h
int 21h

; Les quatre lignes suivantes servent à attendre qu'une touche soit frappée
; avant de quitter le programme normalement.
; Elles peuvent donc être exclues du programme,
; à ceci près que la visualisation du résultat sera impossible,
; la fenêtre de console se refermant trop vite.
xor ah, ah
int 16h
mov ax, 4c00h
int 21h

message db "Hello, world!$"
```

1949 : **LANGUE DE L'ASSEMBLEE**

La langue de l'assemblée a été utilisée dans le **Calculateur automatique du délai de conservation électronique** (EDSAC). Le langage assembleur est un type de **langage de programmation de bas niveau** qui simplifie le langage du code machine. En d'autres termes, les instructions spécifiques nécessaires au fonctionnement d'un ordinateur.

1949 : **CODE COURT**

Shortcode (ou code de commande court), a été le premier **Langue de haut niveau** (HLL) proposé par **John McCauley** en 1949. Cependant, c'est **William Schmitt** qui l'a mis en œuvre pour l'ordinateur **BINAC** la même année et pour l'**UNIVAC** en 1950.

1952 : **AUTOCODE**

Autocode était un terme général utilisé pour une famille de langages de programmation. Développé à l'origine par **Alick Glennie** pour l'ordinateur **Mark 1** de l'université de Manchester, Autocode était la toute première langue compilée à être mise en œuvre ce qui signifie qu'il peut être traduit directement en code machine à l'aide d'un programme appelé **compilateur**. Autocode a été utilisé sur les premières machines de calcul de **Ferranti Pegasus** et **Sirius** en plus du **Mark 1**.

1954 : Fortran



◆ **Fortran** (mathematical FORMula TRANslating system) est un langage de programmation généraliste dont le domaine de prédilection est le calcul scientifique et le calcul numérique. Il est utilisé aussi bien sur ordinateur personnel que sur les superordinateurs, a été créé par **John Backus** et est considéré comme le plus ancien langage de programmation **en usage aujourd'hui**. Le langage de programmation a été créé pour les calculs scientifiques, mathématiques et statistiques de **haut niveau**. FORTRAN est toujours utilisé aujourd'hui dans certains des superordinateurs les plus avancés du monde.



1954 : Fortran



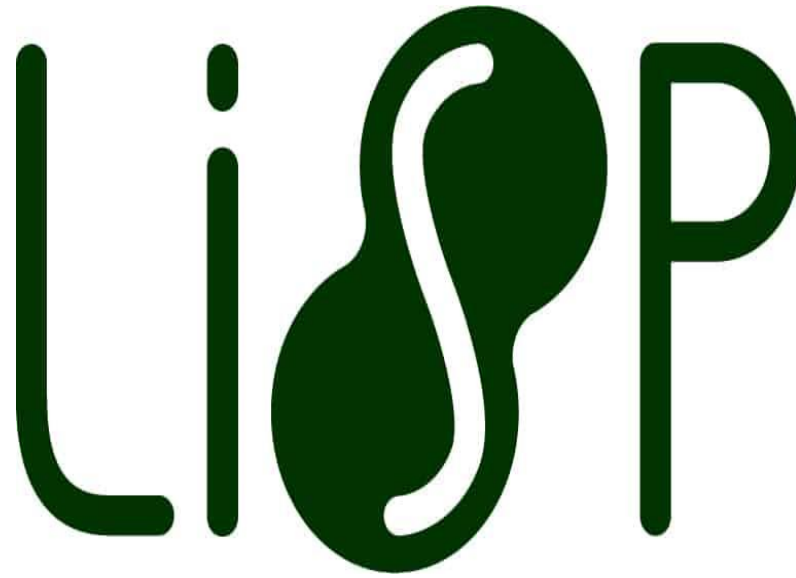
```
1  program helloworld
2  print *, 'Hello World'
3  end program helloworld
4
```

alamy - 2GKAGG6

1958 : LISP

(LIST PROCESSOR)

Le **processeur de liste** ou **LISP** a été inventé par **John McCarthy** au Massachusetts Institute of Technology (MIT). Destiné à l'origine à l'**intelligence artificielle**, le **LISP** est l'un des plus anciens langages de programmation encore utilisés aujourd'hui et peut remplacer Ruby ou Python. Des entreprises telles qu'Acceleration,

A large, stylized logo for LISP in a dark green color. The letters are bold and rounded. The 'i' has a dot, and the 'S' is a continuous, flowing shape that connects to the 'P'.

1958 : ALGOL

(LANGAGE ALGORITHMIQUE)

Le **langage algorithmique** ou **ALGOL** a été créé par un comité conjoint d'informaticiens américains et européens. **ALGOL** a servi de point de départ au développement de certains des plus importants langages de programmation, dont **Pascal**, **C**, **C++** et **Java**.

1958 : LISP



```
Common Lisp.lisp
0 (defun hello-world ()
1   (format t "Hello World~%"))
2
3 (hello-world)
4
```

LISP

1958 : ALGOL



Programming Language: Algol-60

```
001:'BEGIN'
002: 'COMMENT' Hello World in Algol 60;
003:  OUTPUT(4, '('('Hello World!'), '/')')
004:'END'
```

'Hello World!' NFTs

1959 : COBOL (COMMON BUSINESS ORIENTED LANGUAGE)



La langue commune axée sur les affaires (COBOL), est un langage de programmation. Il est à l'origine de nombreux processeurs de cartes de crédit, de distributeurs automatiques de billets, d'appels téléphoniques et cellulaires, de signaux d'hôpitaux et de systèmes de signalisation routière. Le développement du langage a été dirigé par le **Dr Grace Murray Hopper** et a été conçu de manière à pouvoir fonctionner sur toutes les marques et tous les types d'ordinateurs. Le COBOL est encore utilisé aujourd'hui, principalement dans le **secteur bancaire** et dans le **secteur des services**.

```
IDENTIFICATION DIVISION.  
    PROGRAM-ID. HELLO-WORLD  
*  
ENVIRONMENT DIVISION.  
*  
DATA DIVISION.  
*  
PROCEDURE DIVISION.  
    PARA-1.  
        DISPLAY "Hello, world."  
*  
        EXIT PROGRAM.  
    END PROGRAM HELLO-WORLD
```


1964 : BASIC

(CODE D'INSTRUCTION
SYMBOLIQUE
UNIVERSEL POUR
DEBUTANTS)

Basic (de l'acronyme anglais BASIC pour *Beginner's All-purpose Symbolic Instruction Code* est littéralement « code d'instruction symbolique multiusage du débutant »), fait partie d'une famille de langages de programmation de haut niveau ayant pour caractéristique leur facilité d'utilisation. La première version est présentée par John George Kemeny et Thomas Eugene Kurtz au Dartmouth College en 1964. À l'époque, la quasi-totalité des ordinateurs exigeait l'écriture de logiciels personnalisés, que seuls les scientifiques et les mathématiciens avaient tendance à apprendre. La création de ce système a permis à des étudiants de domaines non scientifiques d'utiliser des ordinateurs.



Microsoft

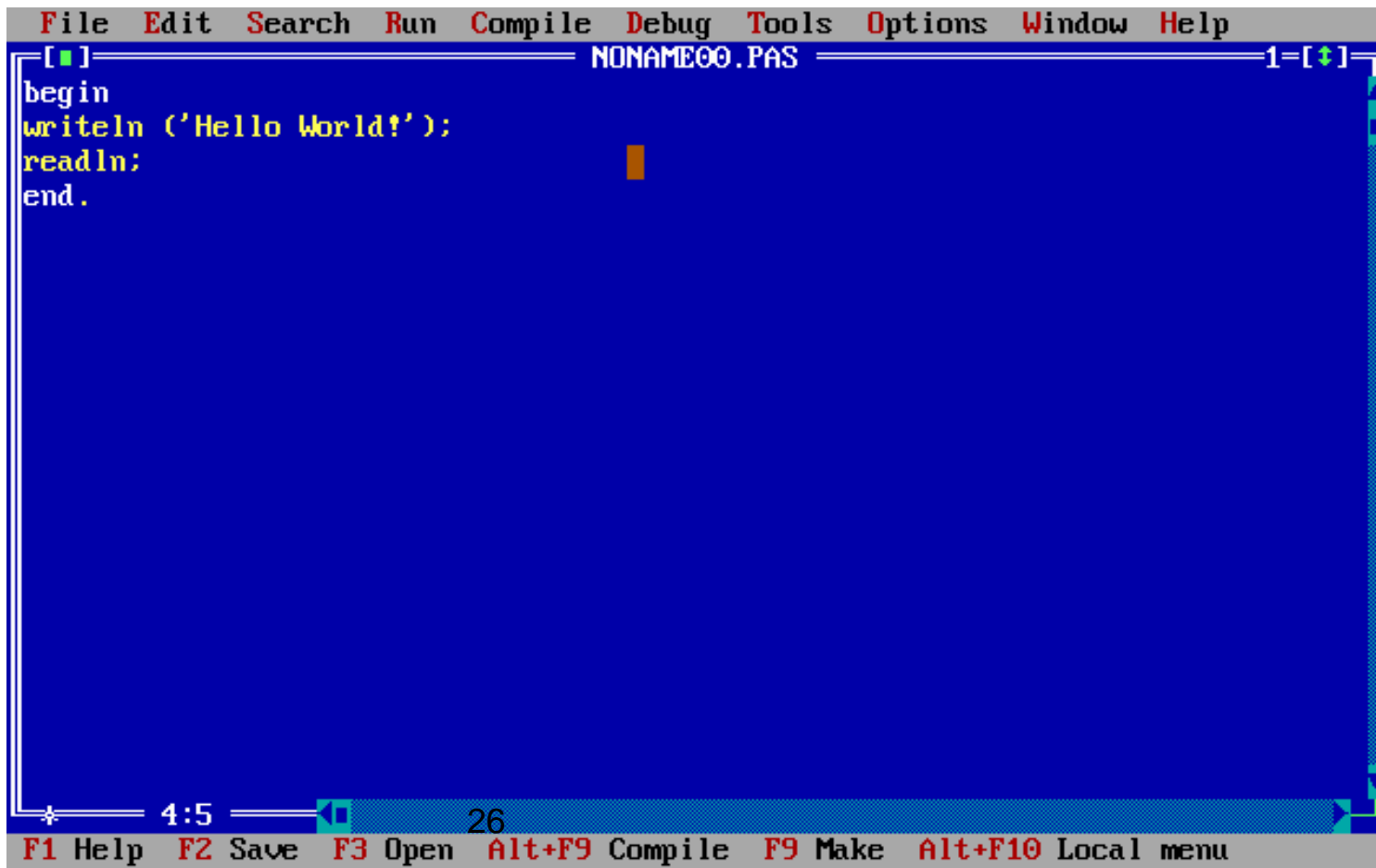
1970 : PASCAL



Pascal

Nommé d'après le mathématicien français Blaise Pascal, Niklaus Wirth a développé le langage de programmation en son honneur. Il a été développé comme un outil d'apprentissage de la programmation informatique, ce qui signifie qu'il est facile à apprendre. Il a été préféré par Apple à ses débuts, en raison de sa facilité d'utilisation et de sa puissance.

PASCAL



The image shows a screenshot of a Turbo Pascal IDE window. The title bar reads "NONAME00.PAS". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The main editing area has a blue background and contains the following Pascal code:

```
[ ] begin  
writeln ('Hello World!');  
readln;  
end.
```

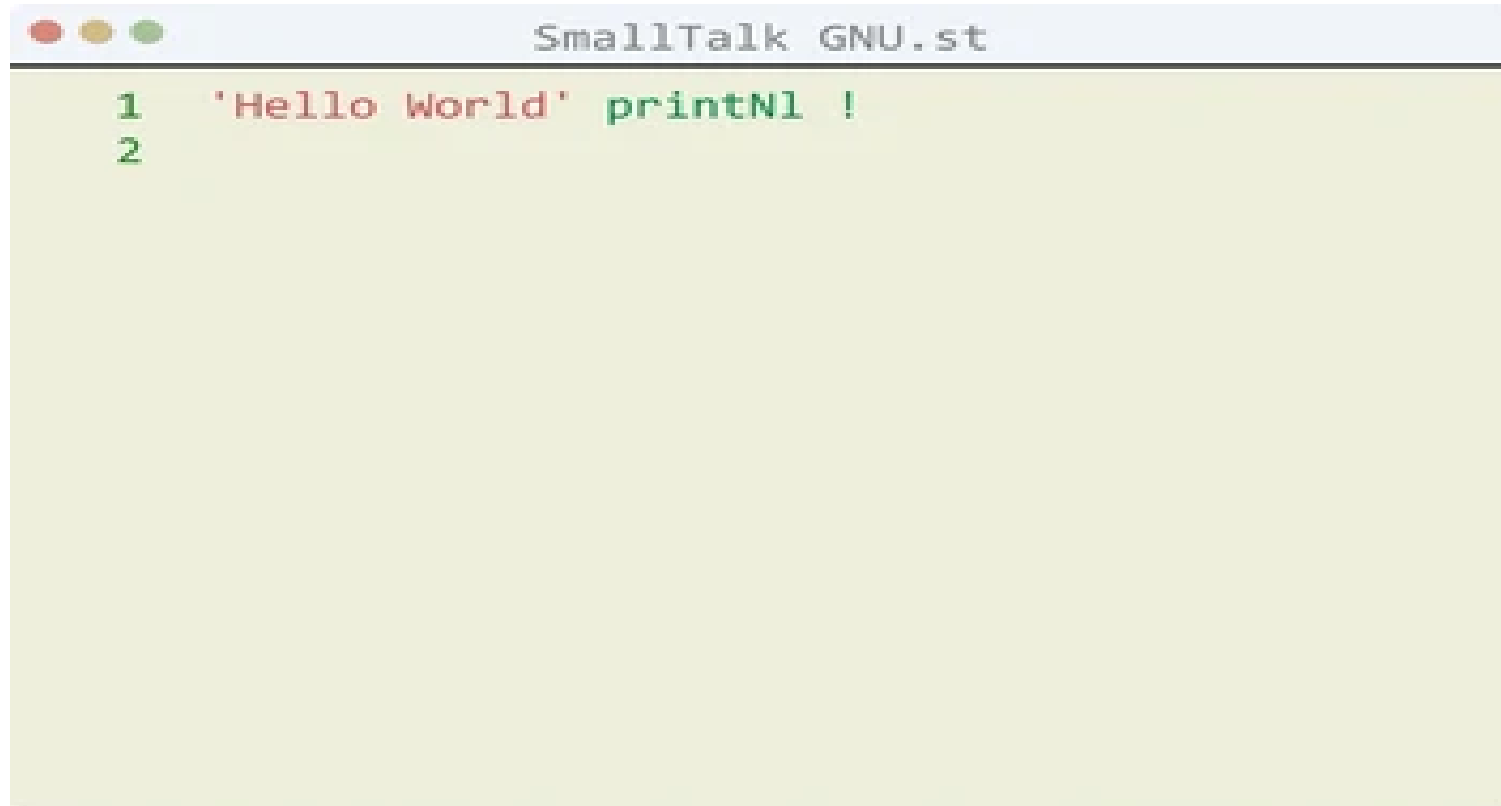
An orange cursor is positioned at the end of the "readln;" line. The status bar at the bottom shows the cursor position as "4:5" and the page number "26". The bottom-most bar contains function key shortcuts: "F1 Help", "F2 Save", "F3 Open", "Alt+F9 Compile", "F9 Make", "Alt+F10 Local menu", and "1=[]".

1972 : **SMALLTALK**

Développé au centre de recherche de **Xerox Palo Alto** par **Alan Kay**, **Adele Goldberg** et **Dan Ingalls**, **Smalltalk** permettait aux programmeurs informatiques de **modifier le code à la volée**. Il a introduit une variété d'aspects des langages de programmation qui sont des langages visibles aujourd'hui, tels que **Python**, **Java** et **Ruby**. Des entreprises telles que **Leafly**, **Logitech** et **CrowdStrike** déclarent utiliser **Smalltalk** dans leurs piles technologiques.



1972 : SMALLTALK



1972 : C

Développé par **Dennis Ritchie** aux **Bell Telephone Laboratories** pour une utilisation avec le système d'exploitation **Unix**. Il a été appelé **C** parce qu'il était basé sur un langage antérieur appelé "**B**". La plupart des principaux langages actuels sont des dérivés du C, notamment le **C#**, **Java**, **JavaScript**, **Perl**, **PHP** et **Python**. Il a également été/est toujours utilisé par de grandes entreprises comme **Google**, **Facebook** et **Apple**.



```
#include <stdio.h>
#include <stdlib.h>

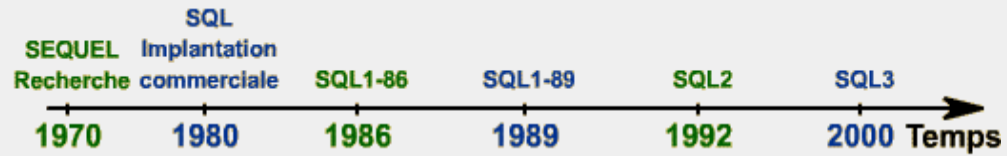
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

1972 : **SQL** (*SEQUEL* à l'époque)



SQL a d'abord été développé par les chercheurs d'IBM **Raymond Boyce** et **Donald Chamberlain**. **SEQUEL** (comme on l'appelait à l'époque), est utilisé pour **visualiser** et **modifier** les informations stockées dans les **bases de données**. Aujourd'hui, le langage est un **acronyme** - **SQL**, qui signifie **Structured Query Language** (langage de requête structuré). Il y a une pléthore d'entreprises qui utilisent SQL et parmi elles, **Microsoft** et **Accenture**.

SQL (1970)



SQL (1970)



- ◈ SEQUEL est implémenté dans les années 1970 à l'université de Berkeley puis sur un prototype du laboratoire d'IBM à San José
- ◈ la première implémentation commerciale démarre en fin 1970-début 1980 avec les bases de données SQL-DS d'IBM. Suivent rapidement DB2, Oracle et Ingres,
- en 1986, c'est la 1ère normalisation de SQL (SQL1-86) par l'ANSI,
- en 1989, la norme SQL1-89 intègre l'intégrité référentielle,
- en 1992, la version SQL2-92, dite SQL2, est normalisée par l'ANSI et l'ISO : elle est supportée par un grand nombre **de bases de données relationnelles**,
- en l'an 2000, la version objet de SQL (SQL3) permet d'interroger **les bases de données objet-relationnelles**

SQL (1970)



```
1. DECLARE
2. text VARCHAR2(25);
3. BEGIN
4. text:= 'Hello World';
5. dbms_output.put_line (text);
6. END;
7. /
```

Output:

Hello World

ADA (1980)



- ◆ **Ada** est un langage de programmation orienté objet dont les premières versions remontent au début des années 1980
- ◆ Ada est un langage de programmation conçu par l'équipe de CII-Honeywell Bull dirigée par Jean Ichbiah en réponse à un cahier des charges établi par le département de la Défense des États-Unis (DoD). Les premiers travaux autour du projet ont commencé à la fin des années 1970² et dès 1979 un *preliminary ADA reference manual* est publié dans la revue du *Special Interest*. La première version validée du projet sort au début des années 1980 sous le nom d'Ada 83.

Un exemple courant pour montrer la syntaxe d'un langage est le programme Hello world



```
with Ada.Text_IO;  
  
procedure Hello is  
begin  
    Ada.Text_IO.Put_Line("Hello, world!");  
end Hello;
```

1983 : C++



Bjarne Stroustrup a modifié le langage C aux Bell Labs, C++ est une extension du C avec des améliorations telles que des **classes**, des **fonctions virtuelles** et des **modèles**. Il a été répertorié dans les *10 principaux langages de programmation* depuis 1986 et a reçu le statut de Hall of Fame en 2003. Le C++ est utilisé dans MS Office, Adobe Photoshop, les moteurs de jeu et d'autres logiciels performants.

1983 : C++

```
1 // c++ hello world program
2 #include <iostream>
3
4 int main() {
5     // Program code here
6     std::cout << "Hello world!";
7
8     return 0;
9 }
```

1987 : PERL

*(Practical Extraction
and Report Language)*

Perl a été créé par Larry Wall et est un langage de programmation généraliste de haut niveau. Il a été conçu à l'origine comme un langage de script conçu pour l'édition de texte, mais il est aujourd'hui largement utilisé à de nombreuses fins telles que CGI, les applications de base de données, l'administration système, la programmation réseau et la programmation graphique.

```
#!/usr/bin/perl

# import $PATH/$PWD variables as array and scalar
use Env qw(@PATH PWD);

print "PWD: $PWD\n";
print "PATH:\n";

foreach (@PATH) {
```

1972 : OBJECTIF-C

Développé par **Brad Cox** et **Tom Love**, **Objective-C** est un langage de programmation orienté objet. C'est une extension du C ANSI, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Objectif-C est le principal langage de programmation utilisé pour écrire des logiciels pour **MacOs** et **iOS**, les systèmes d'exploitation d'**Apple**.



Example: Hello World

```
#import <Foundation/Foundation.h>

int main (int argc, const char * argv[]) {
    NSAutoreleasePool * pool =
    [[NSAutoreleasePool alloc] init];

    // insert code here...
    NSLog(@"Hello, World!");
    [pool drain];
    return 0;
}
```

NSLog() is equivalent to a C printf()

Very similar to C

C programs are valid in Objective-C

25

1990 : HASKELL



Haskell est un langage de programmation **polyvalent** nommé d'après le logicien et mathématicien américain **Haskell Brooks Curry**. C'est un langage de programmation purement **fonctionnel**, ce qui signifie qu'il est essentiellement **mathématique**. Il est utilisé dans de nombreuses industries, en particulier celles qui font des **calculs complexes**, des **enregistrements** et des **calculs de chiffres**. Comme beaucoup d'autres langages de programmation de cette époque, il n'est pas très courant de voir Haskell utilisé pour des applications bien connues. Cela dit, le langage de programmation a été utilisé pour écrire un certain nombre de **jeux**, dont **Nikki et les robots**.



HASKELL

[illegible]

1991 : PYTHON

Nommé d'après la troupe de comédiens britanniques "**Monty Python**", **Python** a été développé par **Guido Van Rossum**. Il s'agit d'un outil d'usage général Le langage de programmation de haut niveau, créé pour supporter une variété de styles de programmation et être amusant à utiliser (un certain nombre de tutoriels, d'exemples et d'instructions contiennent souvent des références à Monty Python). Python est, à ce jour, l'un des langages de programmation les plus populaires au monde, utilisé par des sociétés telles que Google, Yahoo et Spotify.



1991 : PYTHON

```
Python 3.py
1  #!/usr/bin/env python3
2  print("Hello World")
3
```

alamy

Image ID: 2GKAH3
www.alamy.com

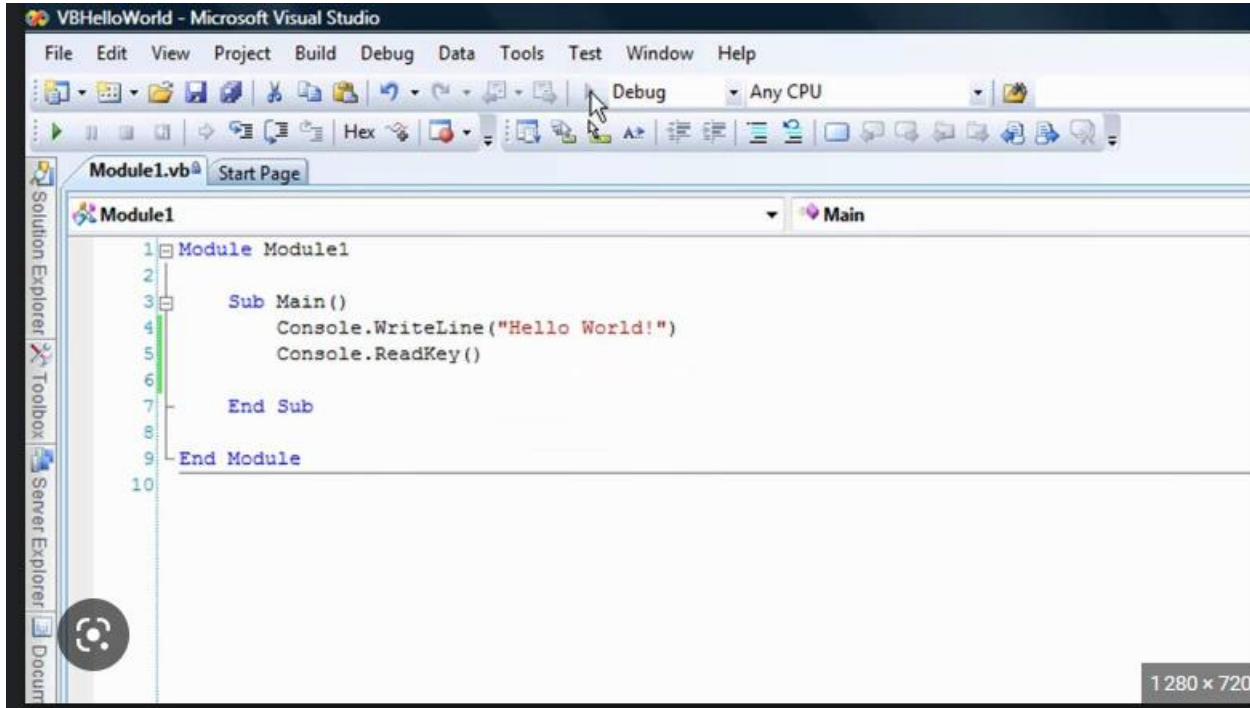


1991 : VISUAL BASIC

Développé par **Microsoft**, **Visual Basic** permet aux programmeurs d'utiliser un style de **glisser-déposer** pour choisir et modifier des morceaux de code présélectionnés grâce à une **interface utilisateur graphique (GUI)**. Le langage n'est pas trop utilisé de nos jours, mais Microsoft a utilisé des parties de Visual Basic pour un certain nombre de ses applications comme **Word**, **Excel** et **Access**.



1991 : VISUAL BASIC

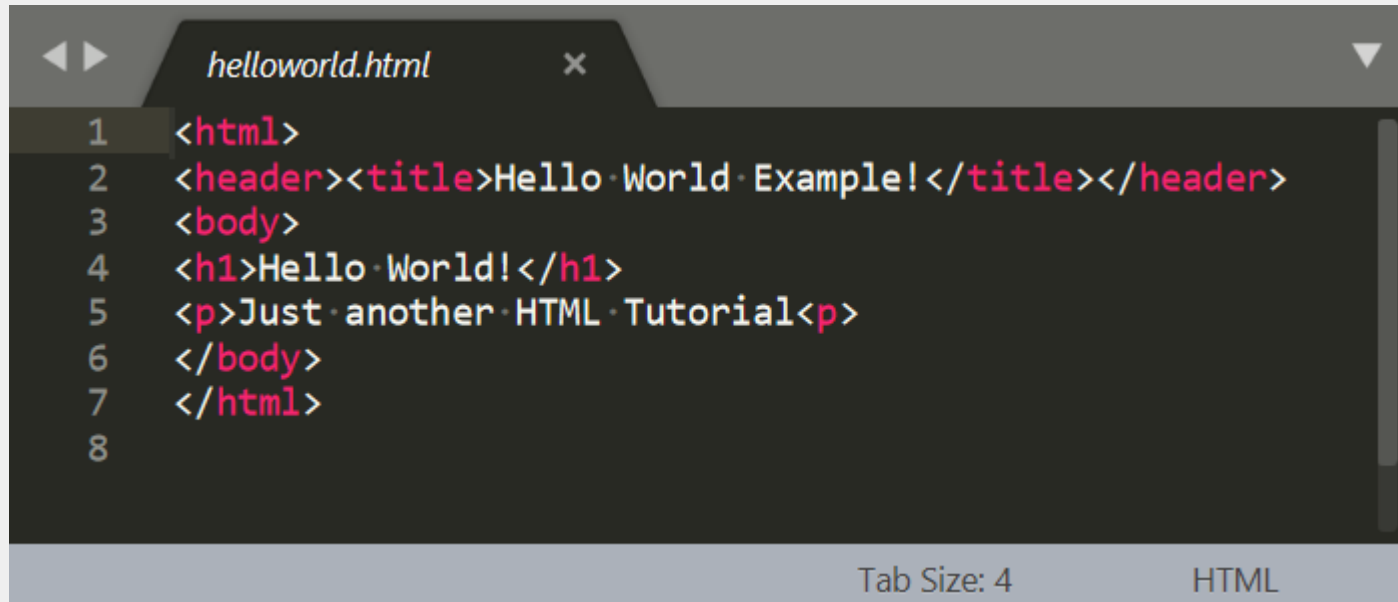


1993 : HTML

Le **HyperText Markup Language**, généralement abrégé **HTML** ou, dans sa dernière version, **HTML5**, est le langage de balisage conçu pour représenter les pages web. HTML est une des trois inventions à la base du *World Wide Web*, avec le ***Hypertext Transfer Protocol*** (HTTP) et les **adresses web** (URL). HTML a été inventé pour permettre d'écrire des documents hypertextuels liant les différentes ressources d'Internet avec des hyperliens. Aujourd'hui, ces documents sont appelés « **page web** ». En août 1991, lorsque **Tim Berners-Lee** annonce publiquement le web sur Usenet, il ne cite que le langage **Standard Generalized Markup Language** (SGML), mais donne l'URL d'un document de suffixe `.html`.



1993 : HTML



A screenshot of a code editor window titled "helloworld.html". The editor displays a simple HTML document structure with the following code:

```
1 <html>
2 <header><title>Hello World Example!</title></header>
3 <body>
4 <h1>Hello World!</h1>
5 <p>Just another HTML Tutorial<p>
6 </body>
7 </html>
8
```

The code is color-coded: tags are in red, and content is in white. The editor has a dark background and a light gray status bar at the bottom showing "Tab Size: 4" and "HTML".

1995 : PHP



Anciennement connu sous le nom de "page d'accueil personnelle", qui signifie maintenant "préprocesseur hypertexte", PHP a été développé par Rasmus Lerdorf. Ses principales utilisations comprennent la création et la maintenance de pages web dynamiques, ainsi que le développement côté serveur. Certaines des plus grandes entreprises du monde entier utilisent PHP, notamment Facebook, Wikipedia, Digg, WordPress et Joomla. Créé en 1994 par Rasmus Lerdorf, la toute première incarnation de PHP était un simple jeu de binaires CGI écrits en langage C.



1995 : PHP



```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

1996 : JavaScript

JavaScript a été créé en 1996 par Brendan Eich et intégré au navigateur web Netscape Navigator 2.0. L'implémentation concurrente de JavaScript par Microsoft dans Internet Explorer jusqu'à sa version 9 se nommait Jscript, est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter.



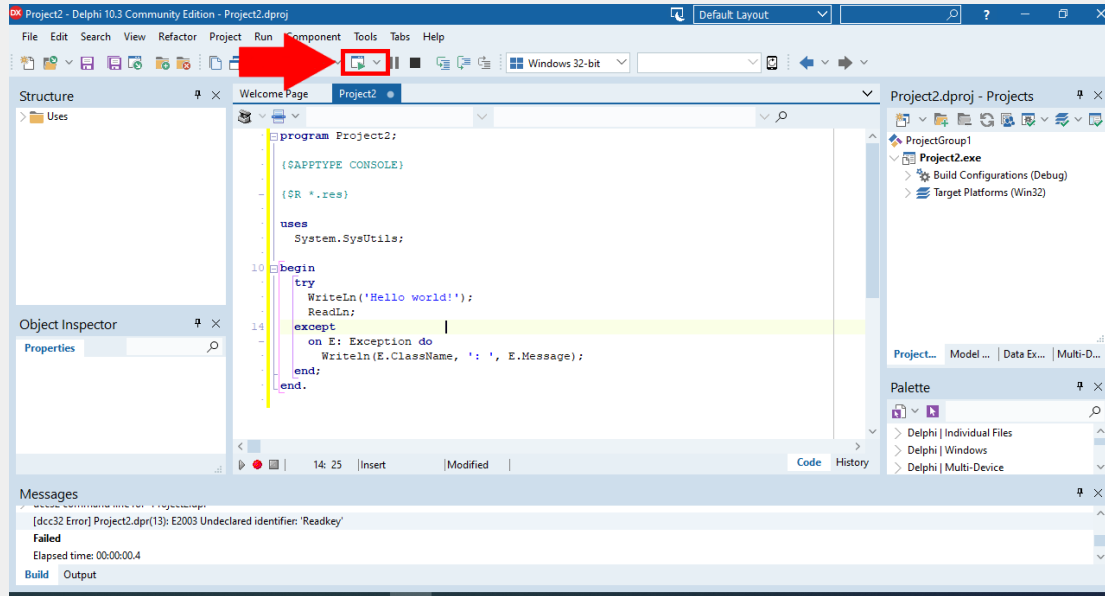
1995 : DELPHI

Delphi est à la fois un langage de **programmation haut niveau orienté objet** et un **environnement de développement intégré (EDI)** pour ce langage.

Delphi est un EDI propriétaire fonctionnant sous Windows créé en **1995** par l'éditeur de logiciels Borland Software Corporation. À l'époque, créer des programmes graphiques sous Windows se faisait en grande majorité en utilisant soit la chaîne de compilation Visual C++, soit le RAD Visual Basic. Le premier outil étant excessivement complexe et le second assez peu structuré, Delphi apparut alors comme une alternative viable pour beaucoup de développeurs qui souhaitaient créer des programmes pour Windows.



1995 : DELPHI



1995 : JAVA

Java est un langage de programmation de haut niveau orienté objet créé par **James Gosling** et **Patrick Naughton**, employés de **Sun Microsystems**, avec le soutien de **Bill Joy** (cofondateur de Sun Microsystems en 1982), présenté officiellement le **23 mai 1995** au *SunWorld*.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers un code intermédiaire formé de *bytecodes* qui peut être exécuté dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

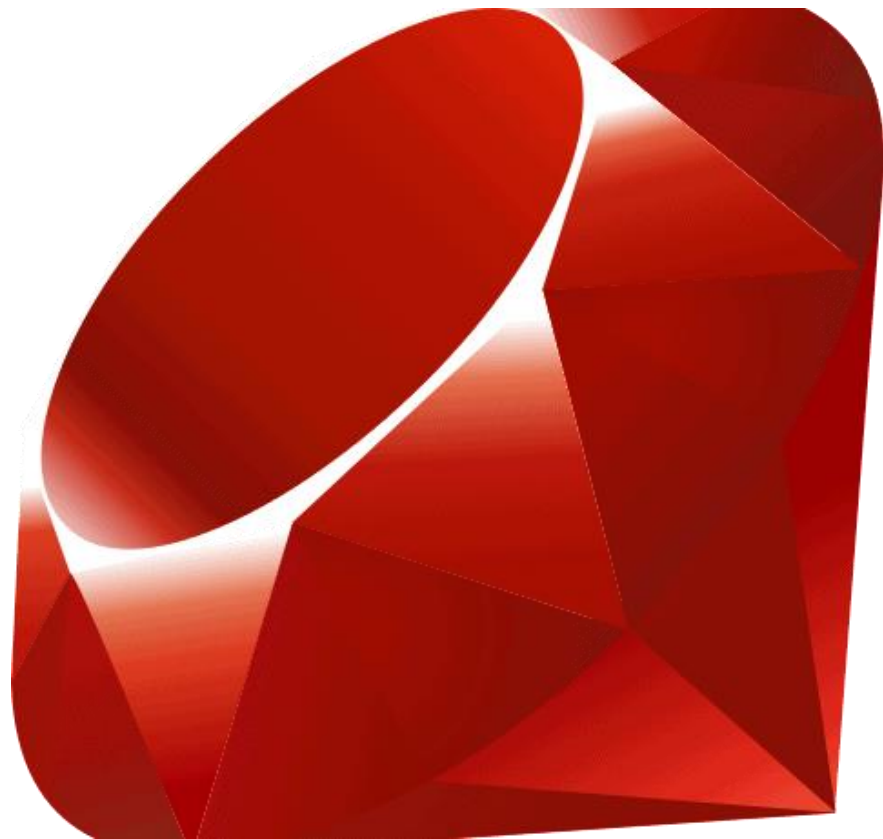


```
HelloWorld.java  SwingIntoHelloWorld.java
1 package com.example.helloworld;
2
3 public class HelloWorld {
4
5     public static void main(String args[]) {
6         System.out.print("Hello World!");
7     }
8
9 }
10
11
12
13
14 |
```


1993 : RUBY



Créé par Yukihiro Matsumoto, **Ruby** est un langage de programmation interprété de **haut niveau**. Un **langage d'enseignement** qui a été influencé par **Perl**, **Ada**, **Lisp** et **Smalltalk**, entre autres. Les principales utilisations de **Ruby** sont le **développement d'applications web** et **Ruby on Rails**. **Twitter**, **Hulu** et **Groupon** sont des exemples bien connus d'entreprises qui **utilisent Ruby**.



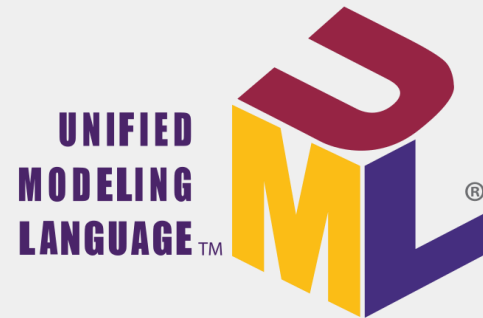
1993 : RUBY

```
Ruby.rb
1  #!/usr/bin/env ruby
2  puts "Hello World"
3
```

UML (1997)



L'UML Le **Langage de Modélisation Unifié**, est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet. L'UML est une synthèse de langages de modélisation objet antérieurs : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997; UML 2.0 a été adopté par l'OMG en juillet 2005¹. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017).



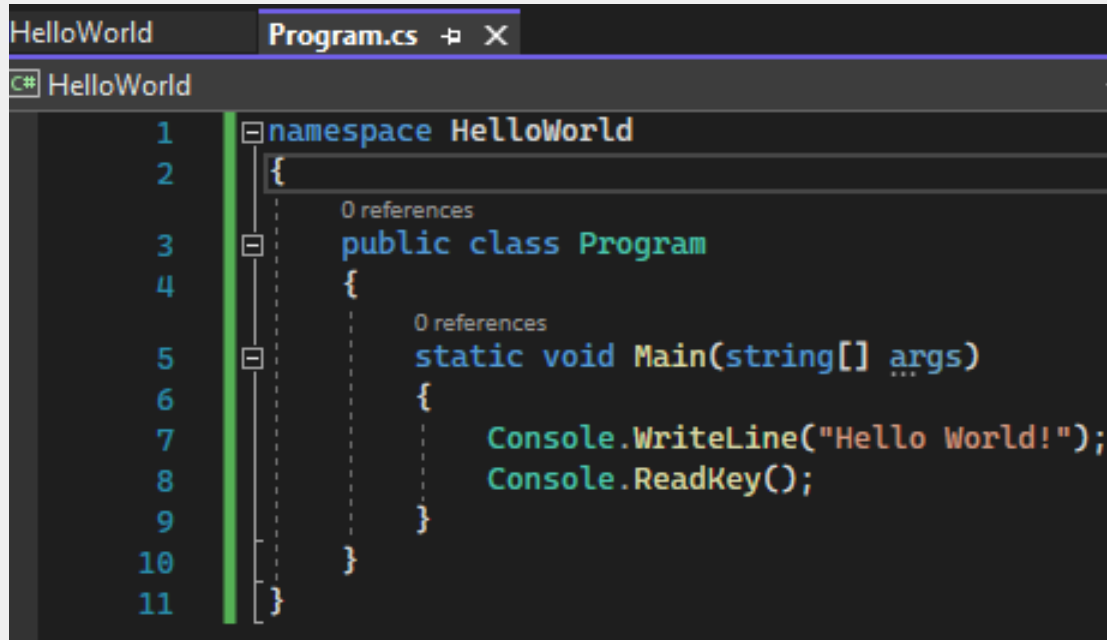
2000 : C#



C# est un langage introduit par Microsoft en 2000. C'est un langage objet avec un typage statique fort, une syntaxe héritée du C/C++ et une philosophie très proche de Java. C# est un langage phare du framework .net qui se popularise pour la conception de sites Web (ASP), d'ERP (Sharepoint), de scripting et d'applications lourdes. A l'origine considéré comme une pâle copie de Java, le C# a ensuite bénéficié d'une politique de développement très dynamique par rapport à Java, resté en désérence plusieurs années par Sun avant son rachat par Oracle. Aujourd'hui, C# est un langage de programmation moderne avec une bibliothèque standard très riche et des outils de développement avancés, La syntaxe du C# est très proche du C et de Java et ne devrait donc pas poser de difficulté si vous avez déjà pratiqué l'un de ces langages. Les éléments de base de la structuration du code sont les accolades {}



2000 : C#



The screenshot shows a code editor window with a tab labeled 'Program.cs'. The code is written in C# and defines a namespace 'HelloWorld' containing a public class 'Program'. The class has a static method 'Main' that takes an array of strings as an argument. Inside the 'Main' method, it prints 'Hello World!' to the console and waits for a key press. The code is as follows:

```
1 namespace HelloWorld
2 {
3     0 references
4     public class Program
5     {
6         0 references
7         static void Main(string[] args)
8         {
9             Console.WriteLine("Hello World!");
10            Console.ReadKey();
11        }
12    }
13 }
```

2003 : Scala



Martin Odersky a commencé à développer **Scala** en **2001** à l'École Polytechnique Fédérale de Lausanne (EPFL). Après une diffusion interne fin 2003, Scala a été rendu public début 2004 sur la plateforme Java. Une seconde version a suivi en mars 2006.

Scala est un langage de programmation multi-paradigme conçu à l'École polytechnique fédérale de Lausanne (EPFL) pour exprimer les modèles de programmation courants dans une forme concise et élégante. Son nom vient de l'anglais **Scalable language** qui signifie à peu près « langage adaptable » ou « langage qui peut être mis à l'échelle ». Il peut en effet être vu comme un métalangage.



2003 : Scala



The screenshot shows a window titled "HelloWorld.Scala (~)" with a standard menu bar (File, Edit, View, Search, Tools, Documents, Help) and a toolbar with icons for file operations and editing. The code editor displays the following Scala code:

```
object HelloWorld
{
  def main(args: Array[String])
  {
    //This is a Hello World function in Scala
    println("Hello World!")
  }
}
```

2003 : Groovy



Groovy est le nom d'un langage de programmation orienté objet destiné à la plate-forme Java. Il constitue un substitut au langage Java pour cette plate-forme et est inspiré de Python, Ruby et Smalltalk.

Groovy utilise une syntaxe très proche de Java, avec des accolades, et est directement compilé, soit à la volée dynamiquement, soit classiquement avec un compilateur en bytecode. Groovy s'intègre et est entièrement compatible avec la JVM étant donné que le bytecode est le même. Il peut donc :

- utiliser les bibliothèques Java ;
- être utilisé dans des classes Java.



2003 : Groovy



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole". The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations (open, save, print), editing (undo, redo, copy, paste), and execution (run, stop, clear). The script editor contains the following code:

```
1  
2  
3  
4 println "Hello World!"
```

Below the script editor, the execution output is displayed on a yellow background. The prompt "groovy>" is followed by the command "println \"Hello World!\"". The output "Hello World!" is shown below the command and is enclosed in a red rectangular box.

```
groovy> println "Hello World!"  
Hello World!
```

2009 : GO



Go est un langage de programmation compilé et concurrent inspiré de C et Pascal. Il a été développé par **Google** à partir d'un concept initial de **Robert Griesemer**, **Rob Pike** et **Ken Thompson**.

Le langage Go a été créé pour la programmation système et a depuis été étendu aux applications, ce qui constitue la même cible que le C et surtout le C++. Il s'agit d'un langage impératif et concurrent. Sa vitesse de compilation (due à la simplicité de sa syntaxe) le fait parfois utiliser comme langage de script.



2009 : GO



```
hello.go x
1 package main
2 import "fmt"
3
4 func main() {
5     fmt.Printf("hello, world\n")
6 }
7
```

2011 : Dart

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par **Google** et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

Dart a été dévoilé lors de la conférence GOTO à Aarhus, au Danemark, du 10 au 12 octobre **2011**. Le projet a été fondé par Lars Bak et Kasper Lund³. Dart 1.0 est sorti le 14 novembre 2013, Dart est un langage orienté objet à ramasse-miettes avec une syntaxe de type C++1. Dart peut se compiler en code natif ou en JavaScript. Il prend en charge les interfaces, les mixins, les classes abstraites, les génériques réifiés et l'inférence de type



Dart

2011 : Dart

```
//Simple Dart Program

void main()
{
    print('Hello Dart');

    print("Hello Flutter");

    print("Codejitter - Hello World");

}
```

2012 : TypeScript



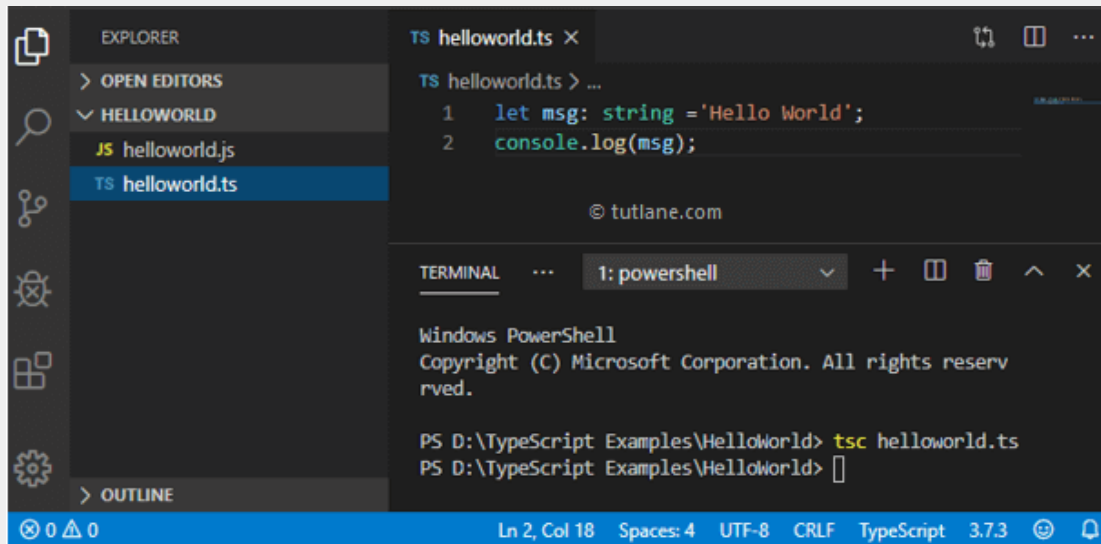
TypeScript est un langage de programmation libre et open source développé par **Microsoft** qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). Le code TypeScript est transcompilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript.

TypeScript a été rendu public en octobre **2012** (à la version 0.8), après deux ans de développement interne chez Microsoft.

The TypeScript logo, which consists of the word "TypeScript" in white text on a blue rectangular background with a subtle pattern of horizontal lines.

TypeScript

2012 : Type Script





03

TYPES DE LANGAGES

Quels sont les types de langage ?



- *Langage machine*
- *Langage assembleur (assembly language)*
- *Langage évolué*
- *Les langages de scripts (langages interprétés)*
- *Les langages compilés*
- *Langage de balisage*
- *Langage de requête*

Langage machine

- ◆ Le **langage machine** appelé aussi **langage binaire**, c'est avec ce langage que fonctionnent les ordinateurs. Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations (texte, images etc.).
- ◆ Dans *le langage machine*, le programmeur doit entrer chaque commande et toutes les données sous forme **binaire** c'est à dire sous forme de suite d'**octets**, chaque octet comportant une succession de 8 **bits**, chaque **bit** étant représenté par le 1 ou le 0

Langage assembleur

- ◆ *Le langage assembleur* est un langage de bas niveau proche du *langage machine* qui peut être directement interprété par le microprocesseur de l'ordinateur tout en restant lisible par un humain.
- ◆ L'assembleur a été créé pour faciliter le travail des programmeurs. Il consiste à représenter les combinaisons de bits employées en *langage binaire* par des symboles faciles à retenir : Pour chaque instruction exprimée en *langage machine*, le programmeur code ses programmes en *langage assembleur*, ceux-ci sont ensuite transcrits par un logiciel appelé assembleur en *langage machine*, puis exécutés par l'ordinateur.

Langage évolué ou langage de haut niveau

- ◆ On désigne par *langage évolué* tous les langages se situant au dessus des *langages de bas niveau (langage machine, assembleur)*. Le *langage évolué* est un langage qui accomplit beaucoup pour un minimum de code et d'effort de programmation, il y en a tout un paquet et je cite à titre d'exemple : Pascal, Java, C, C++, C#, Visual Basic (ou VB), Delphi, Python, Perl, PHP, JavaScript,
- ◆ *Les langages de scripts (langages interprétés)* : Un script est une suite de commandes destinées à être exécutées dans leur ordre d'apparition ligne par ligne, instruction par instruction, exemples : JavaScript, PHP etc.
- ◆ *Les langages compilés* : Ces langages sont traduits en *langage machine* avant leur exécution par un programme compilateur. Ce qui se traduit par une plus grande rapidité d'exécution . Quelques exemples : Java, Python, C, C++, Pascal etc.

Langage de balisage



- ◆ Ces langages reposent sur ce qu'on appelle des balises ou tags, ces derniers sont des étiquettes avec lesquelles on peut étiqueter des données (mots, texte etc.) pour produire un effet chez eux tant en sens (leurs donner du sens : ceci est un paragraphe, ceci est un titre, citation etc.) qu'en rendu visuel (italique, gras, couleur du texte etc.), on peut étiqueter des données en les encadrant par ces balises, parmi ce type de langages il y a le *HTML (HyperText Markup Language)*

Langages de requêtes



- ◆ Ces langages qualifient le plus souvent les langages propres aux bases de données, ils sont représentés notamment par le SQL(*langage structuré de requêtes*), il permet tout simplement de gérer une base de données par exemple l'interroger, y insérer des données ou en supprimer d'autres, lui demander de faire ressortir des données selon des critères que je lui fixe. On traitera ce langage de manière exhaustive lors du cours PHP/MYSQL



04

TYPES DE PROGRAMMATIONS

Quels sont les types de programmations ?



- ◆ Il existe 4 types principaux de programmation :
- la programmation impérative
 - la programmation orientée objets
 - la programmation fonctionnelle
 - la programmation logique

Programmation impérative



- ◆ La programmation impérative est le type de programmation le plus fréquent. Elle consiste à lister une séquence d'ordres à l'ordinateur.
- ◆ Le code va ensuite être exécuté ligne par ligne jusqu'à la fin de son exécution.
- ◆ Les langages en programmation impérative sont le C, le COBOL, le Small Talk ou encore le Perl. En plus des langages plus connus comme **Python, JavaScript et PHP**.

Programmation structurée



- ◆ La programmation structurée fait partie de la famille de la programmation impérative. Elle utilise un nombre limité de structures de contrôle (if, while, ...), ce qui a pour but de réduire le nombre d'erreurs.
- ◆ Ce paradigme de programmation était populaire dans les années 1970. Mais il est beaucoup moins utilisé aujourd'hui car l'utilisation de goto et d'autres concepts structurés rend le code moins contrôlable (avec des résultats inattendus).
- ◆ Quelques exemples de langages structurés sont **Fortran, Ada ou Pascal**.

Programmation procédurale



- ◆ La programmation procédurale fait partie de la famille de la programmation impérative. Elle a la particularité de fonctionner avec des appels de procédures. Les procédures peuvent être appelées à tout moment et une procédure peut aussi s'appeler elle-même (récursion).
- ◆ Le paradigme procédurale est similaire à la programmation fonctionnelle sauf qu'elle fonctionne avec des procédures plutôt qu'avec des fonctions.
- ◆ Les langages de programmation procédurale sont **BASIC, C, COBOL, Fortran, Pascal, Perl, VBScript**.

Programmation orientée objet (POO)



- ◆ La programmation orientée objet (ou POO) est un paradigme dans lequel on va créer puis utiliser des objets. Un objet est comme une structure qui représente une entité. On peut ensuite ajouter des méthodes et des caractéristiques à ces objets.
- ◆ En programmation par objet, de nombreux concepts ont vu le jour comme par exemple la notion de classe, le polymorphisme ou encore l'héritage.
- ◆ Les langages de programmation orientée objets sont **Smalltalk**, **C++**, **Objective C**, **Python**, **PHP**, **Java**, **Ruby**, **C#**.

Programmation fonctionnelle



- ◆ La programmation fonctionnelle fait partie de la famille de la programmation déclarative. Elle se caractérise principalement par la possibilité de déclarer et d'appeler des fonctions à l'intérieur d'autres fonctions.
- ◆ Les langages fonctionnels sont , Haskell, , Scala. Le paradigme de programmation fonctionnel était très apprécié, il est aussi possible de coder de manière fonctionnelle avec des langages comme **JavaScript** ou encore **Python**.

Programmation logique



- ◆ La programmation logique fait partie de la famille de la programmation déclarative. Elle est généralement utilisée dans l'intelligence artificielle. Il s'agit d'une programmation basée sur le calcul de prédicats.
- ◆ En mathématiques, on utilise les prédicats pour trouver des solutions. En informatique on reprend ces concepts pour que l'ordinateur résolve les problèmes.
- ◆ Le langage qui utilise le paradigme de programmation logique par excellence est Prolog. On peut aussi utiliser la programmation logique en **Python**.

Conclusion



L'histoire des langages de programmation est passionnante. Qui aurait pensé qu'un algorithme du milieu du XIXe siècle ouvrirait la voie à la société axée sur la technologie dans laquelle nous vivons aujourd'hui. Des premiers codes machine au code sophistiqué lisible par l'homme qui alimente nos technologies préférées aujourd'hui, les langages de programmation ont parcouru un long chemin. Ce qui est certain, la programmation informatique continuera à se développer comme elle l'a fait au cours des 150 dernières années et il est passionnant de voir ce que l'avenir nous réserve.

RESSOURCES

- <https://www.superprof.fr/blog/la-bible-des-informaticiens/>
- <https://www.eoris.fr/une-histoire-de-linformatique>
- https://fr.wikipedia.org/wiki/Histoire_des_langages_de_programmation
- <https://sup-fc.univ-fcomte.fr/ActuAdaLovelace>
- https://fr.wikipedia.org/wiki/Nombre_de_Bernoulli
- <https://dailygeekshow.com/ada-lovelace-programmation-informatique/>
- <https://fr.wikipedia.org/wiki/Fortran>
- <https://www.mcours.net/cours/pdf/leilcllic2/leilcllic197.pdf>
- <https://fr.vikidia.org/wiki/Lisp>
- [https://fr.wikipedia.org/wiki/Algol_\(langage\)](https://fr.wikipedia.org/wiki/Algol_(langage))
- [https://fr.wikipedia.org/wiki/Pascal_\(langage\)](https://fr.wikipedia.org/wiki/Pascal_(langage))
- <https://fr.wikipedia.org/wiki/Smalltalk>
- https://en.wikibooks.org/wiki/Ada_Programming/Basic
- [https://fr.wikipedia.org/wiki/Ada_\(langage\)](https://fr.wikipedia.org/wiki/Ada_(langage))
- <https://fr.wikipedia.org/wiki/C%2B%2B>
- <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/langage-haskell/>
- [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage))
- <https://www.codingame.com/playgrounds/34845/le-php---les-bases-du-langage/introduction>
- <https://fr.wikipedia.org/wiki/PHP>
- https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- <https://fr.wikipedia.org/wiki/Ruby>
- [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))
- <https://learn.microsoft.com/fr-fr/dotnet/csharp/tour-of-sharp/>

RESSOURCES



- [https://fr.wikipedia.org/wiki/Groovy_\(langage\)](https://fr.wikipedia.org/wiki/Groovy_(langage))
- <https://www.techno-science.net/glossaire-definition/Groovy-langage.html>
- [https://fr.wikipedia.org/wiki/Go_\(langage\)](https://fr.wikipedia.org/wiki/Go_(langage))
- <https://interstices.info/naissance-des-langages-de-programmation/>
- <https://www.bocasay.com/fr/evolutions-tendances-langages-developpement-web/>
- <https://devskiller.com/blog/history-of-programming-languages/>
- https://plus.wikimonde.com/wiki/Liste_de_programmes_Hello_world
- <https://www.buvetteetudiants.com/cours/administrateur/html-css/langages-informatiques.php>
- <https://www.superprof.fr/blog/historique-programmes-informatiques/>