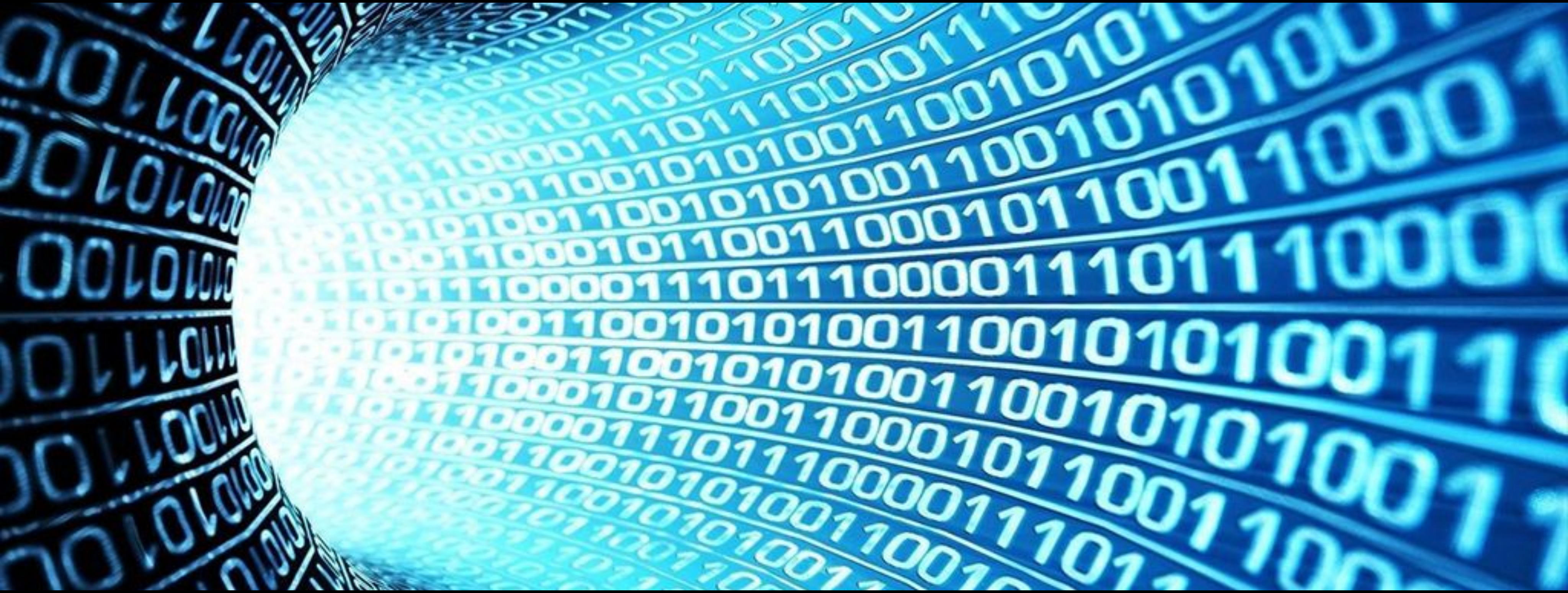
ACADEMIA DE CÓDIGO

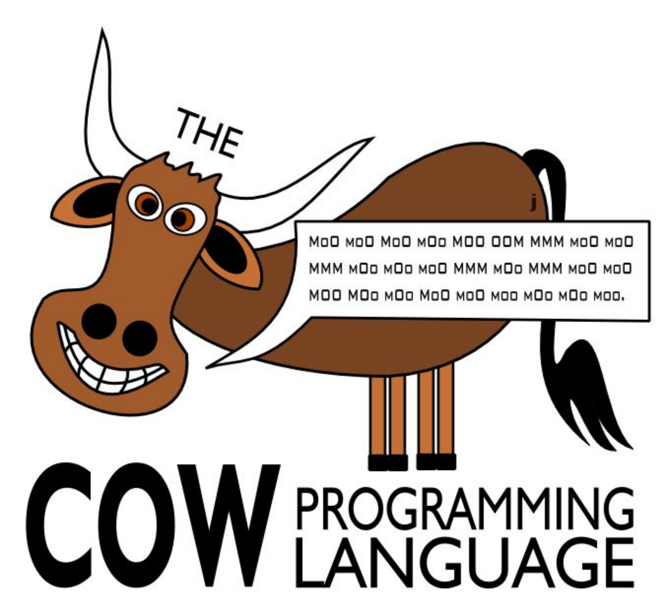


Programming Languages: A short historical overview

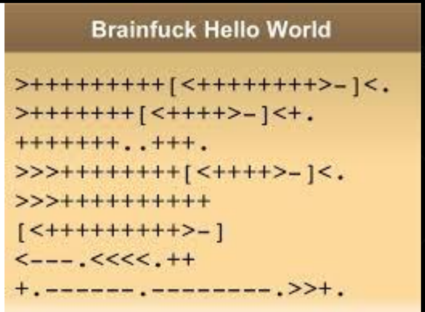
Catarina Figueiredo - May 2018

What is a programming language?

Languages designed to communicate instructions to the machine

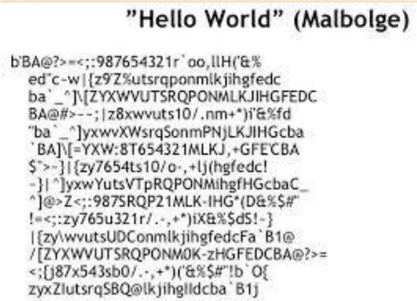


```
MoO moO MoO mOo MOO OOM MMM moO moO
MMM mOo mOo moO MMM mOo MMM moO moO
MOO MOo mOo MoO moO moo mOo mOo moo.
```



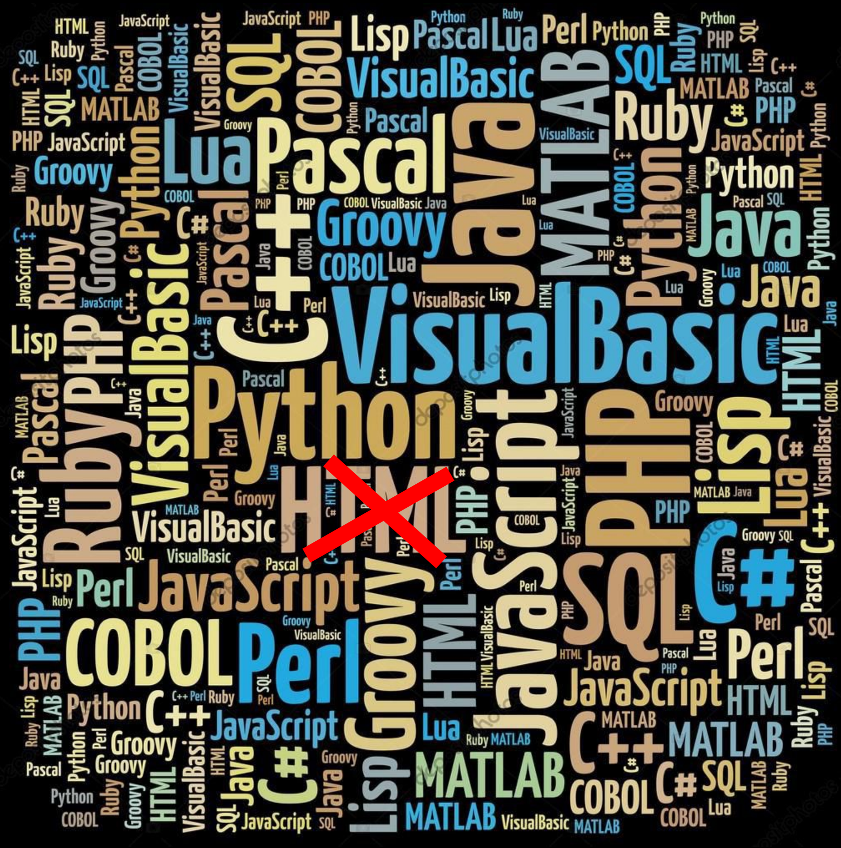
Brainfuck Hello World

```
>+++++++++[<++++++++>-]<.
>+++++++[<++++>-]<+.
+++++++..+++.
>>>++++++++[<++++>-]<.
>>>++++++++++
[<+++++++++>-]
<---.<<<<.++
+.------.--------.>>+.
```



"Hello World" (Malbolge)

```
b'BA@?>=<;:987654321r`oo,llH('&%
   ed"c-w|{z9'Z%utsrqponmlkjihgfedc
   ba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDC
   BA@#>--;|z8xwvuts10/.nm+*)i'&%fd
   "ba`_^]yxwvXWsrqSonmPNjLKJIHGcba
   `BA]\[=YXW:8T654321MLKJ,+GFE'CBA
   $">-}|{zy7654ts10/o-,+lj(hgfedc!
   -}|^]yxwYutsVTpRQPONMihgfHGcbaC_
   ^]@>Z<;:987SRQP21MLK-IHG*(D&%$#"
   !=<;:zy765u321r/.-,+*)iX&%$d5!-}
   |{zy\wvutsUDConmlkjihgfedcFa`B1@
   /[ZYXWVUTSRQPONM0K-zHGFEDCBA@?>=
   <;{j87x543sb0/.-,+*)('&%$#"!b`O{
   zyxZlutsrqSBQ@lkjihglldcba`B1j
```

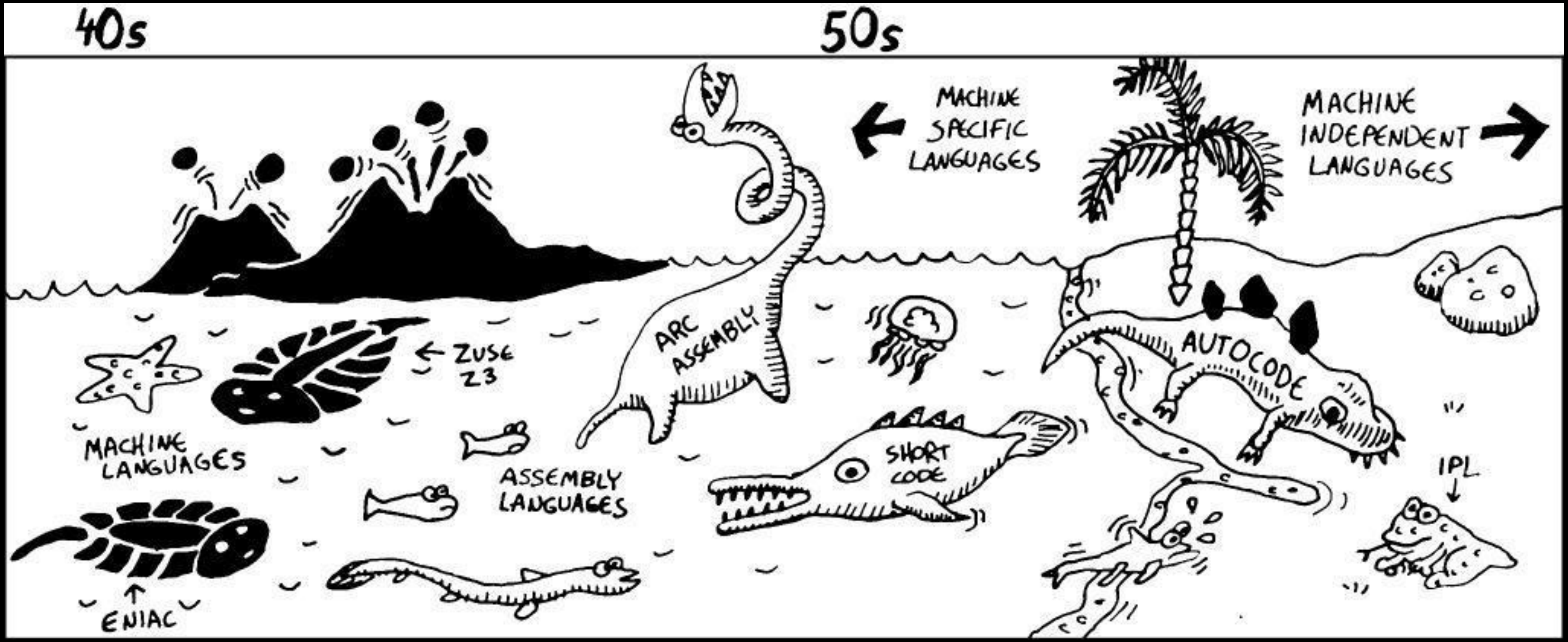


https://tinyurl.com/pkovmx8

"More is not better (or worse) than less, just different."

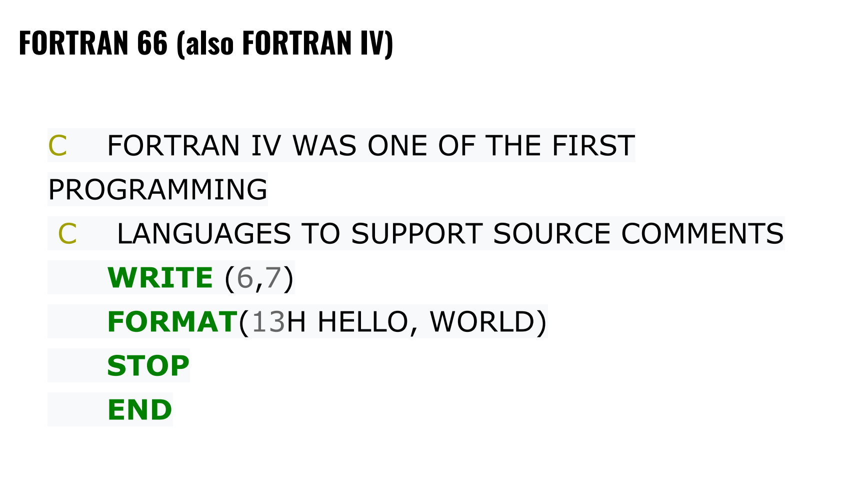
Challenge

Timeline


40s
50s
MACHINE SPECIFIC LANGUAGES
MACHINE INDEPENDENT LANGUAGES
ARC ASSEMBLY
ZUSE Z3
MACHINE LANGUAGES
ASSEMBLY LANGUAGES
SHORT CODE
AUTOCODE
IPL
ENIAC

FORTRAN (FORmula TRANslation)

- FORTRAN can be considered the first high-level language
- Developed by John Backus' group in 1957 (at IBM)
- Designed for applications of numerical-scientific type
- The first version contained many constructs that were more or less independent of a specific machine
- Introduced the first commercially available compiler
  Performance of the compiled code was an important issue



FORTRAN 66 (also FORTRAN IV)

```
C    FORTRAN IV WAS ONE OF THE FIRST
PROGRAMMING
 C    LANGUAGES TO SUPPORT SOURCE COMMENTS
     WRITE (6,7)
     FORMAT(13H HELLO, WORLD)
     STOP
     END
```

ALGOL (ALGOrithmic Languages)

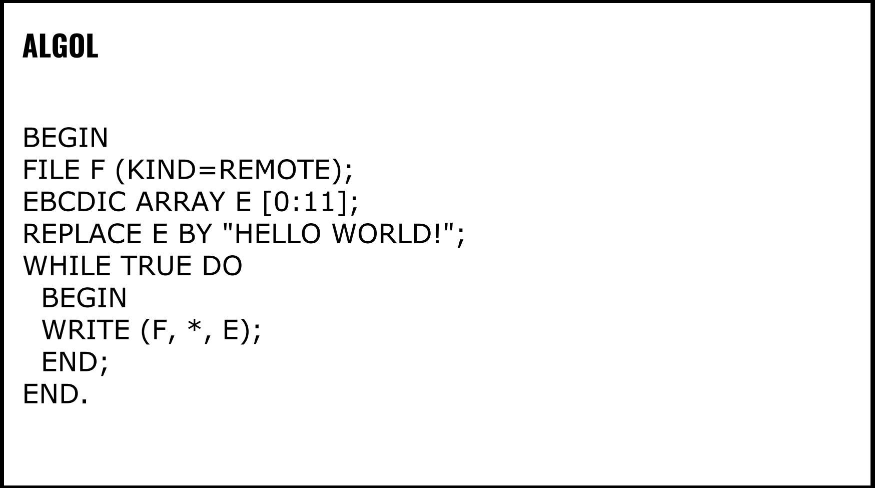
- Algol was developed jointly by a committee of European and American computer scientists in a meeting in 1958 at ETH Zurich[C1]
- Suited for expressing algorithms in general, rather than for use in specific types of applications
- Many of the concepts and constructs that are found in modern languages were introduced in Algol:

Blocks

Recursion

Type system

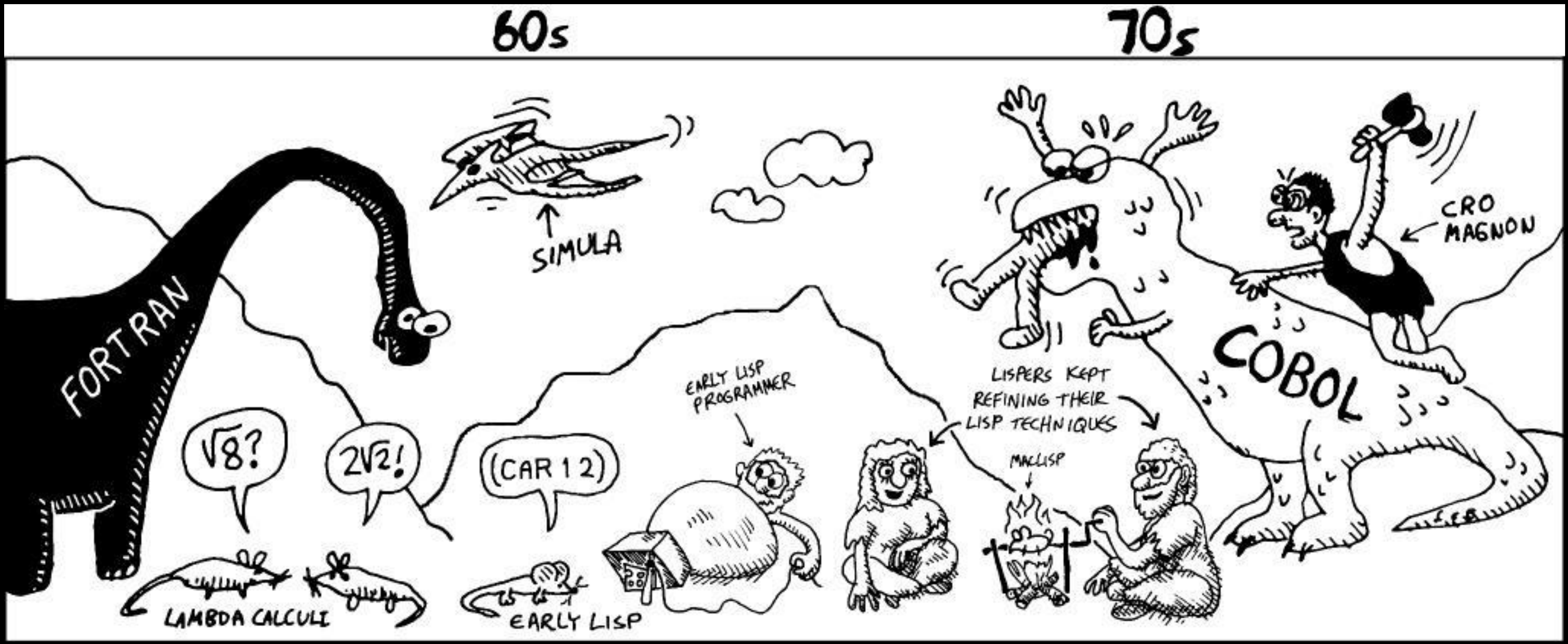
BNF



ALGOL

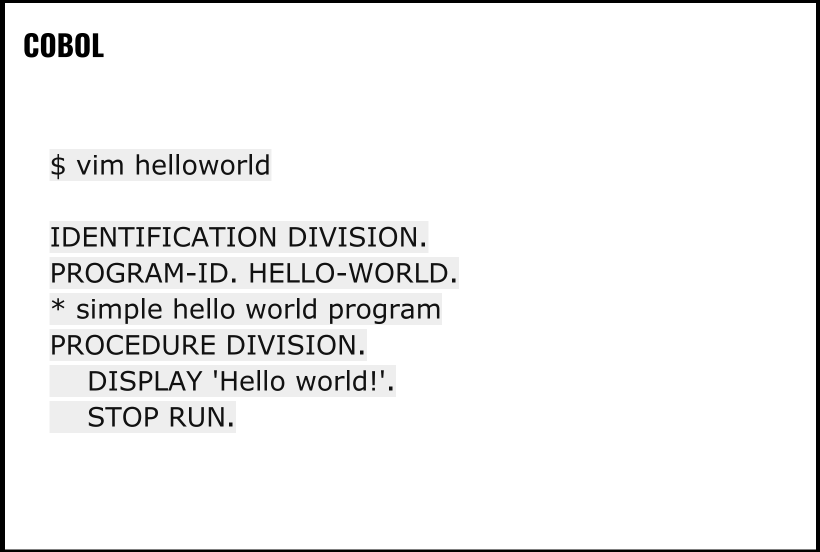
```
BEGIN
FILE F (KIND=REMOTE);
EBCDIC ARRAY E [0:11];
REPLACE E BY "HELLO WORLD!";
WHILE TRUE DO
  BEGIN
  WRITE (F, *, E);
  END;
END.
```

Timeline


60s
70s
SIMULA
FORTRAN
√8?
2√2!
(CAR 1 2)
LAMBDA CALCULI
EARLY LISP
EARLY LISP PROGRAMMER
LISPERS KEPT REFINING THEIR LISP TECHNIQUES
MACLISP
COBOL
CRO MAGNON

COBOL (Common Business Oriented Language)

- Designed in the 1960s for commercial applications
- One aim was that the syntax should be as close to the English language as possible
- Not many new programs are implemented into this language, still, around of 80% of businesses still run on COBOL



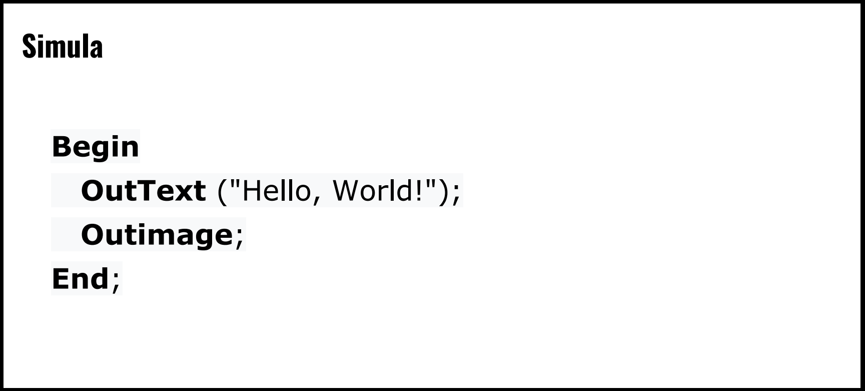
COBOL

```
$ vim helloworld

IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.
* simple hello world program
PROCEDURE DIVISION.
    DISPLAY 'Hello world!'.
    STOP RUN.
```

Simula

- The first object-oriented language
- Descendent of Algol
- Designed for simulation applications
- Developed from 1962 at the Norwegian Computing Centre
- Simula67 introduced for the first time concepts of
  - Class
  - Object
  - Subtype
  - Dynamic method dispatch
- Had a considerable influence on its successors (Smalltalk and C++)

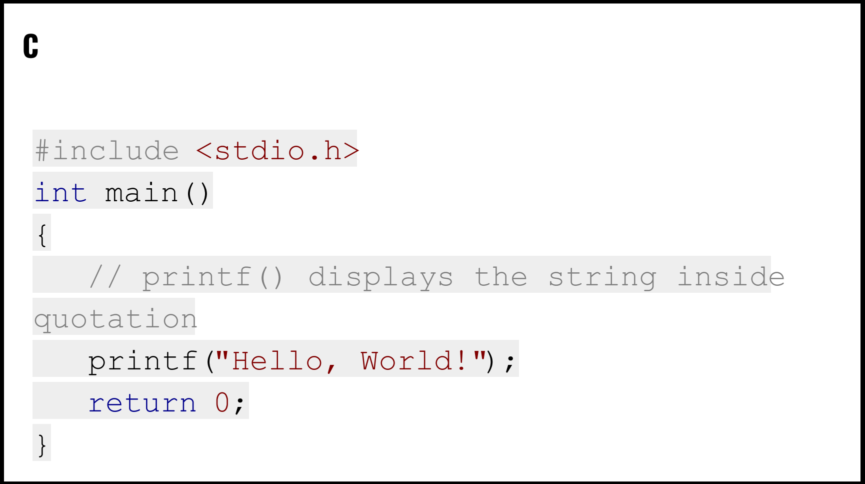


Simula

```
Begin
   OutText ("Hello, World!");
   Outimage;
End;
```

C

- Designed between 1969 and 1973 by Dennis Ritchie and Ken Thompson at Bell Labs
- Designed as a system programming language for Unix
- C offered more opportunities to access the low-level functionality of the machine and to program interactive systems than high-level languages
- Became a general purpose language
- The block structure of C considerably simplifies that found in languages of Algol family (C does not allow nested functions)

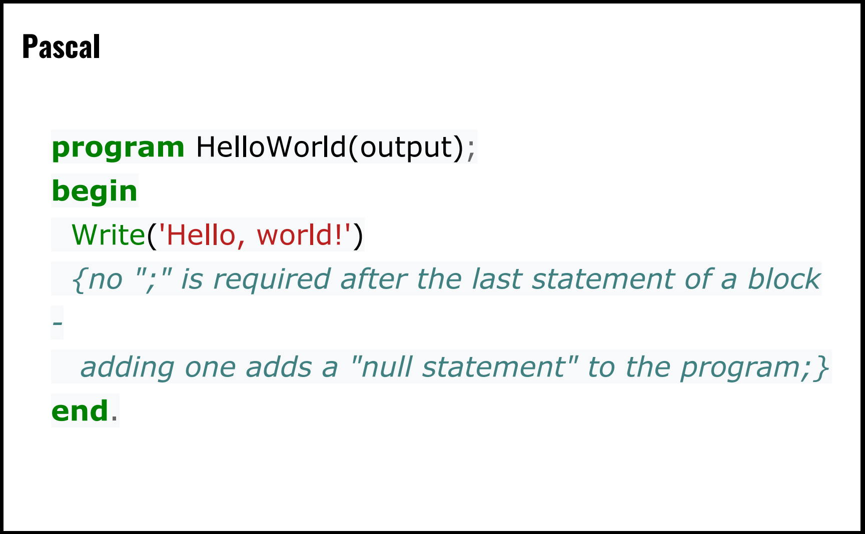


C

```
#include <stdio.h>
int main()
{
   // printf() displays the string inside
quotation
   printf("Hello, World!");
   return 0;
}
```

Pascal

- Developed around 1970 by Niklaus as an improvement and simplification of ALGOL
- Was the most used educational language right up to the end of the 1980s
- Introduced the concept of intermediate code as an instrument for program portability
- Pascal is a block-structured language in which it is possible to define functions and blocks that can be nested with arbitrary complexity
- Limit: The lack of separately compilable modules (solved in later versions like Turbo Pascal / Object Pascal)

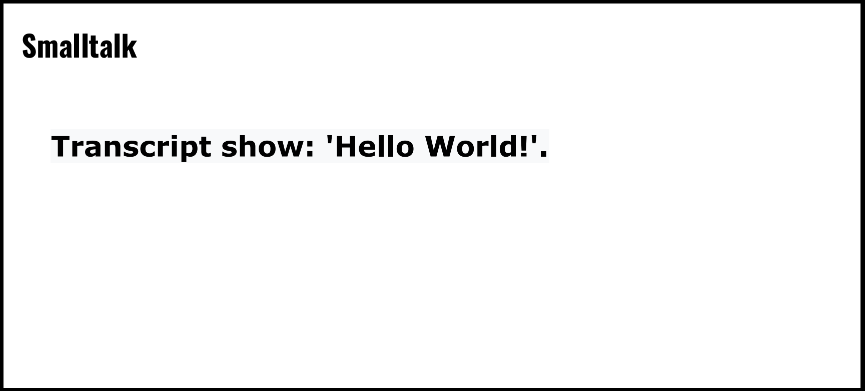


Pascal

```
program HelloWorld(output);
begin
 Write('Hello, world!')
 {no ";" is required after the last statement of a block
-
  adding one adds a "null statement" to the program;}
end.
```

Smalltalk

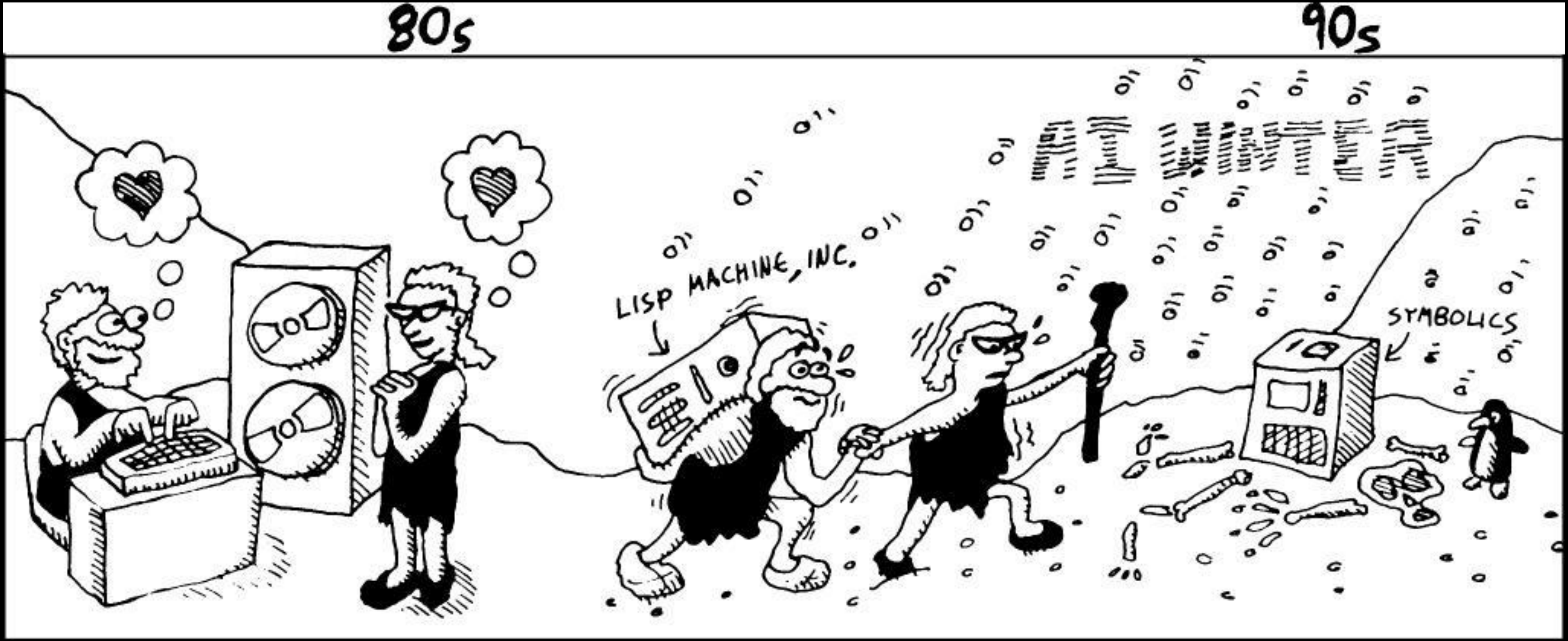
- Developed during the 1970 by Alan Kay at Xerox PARC
- Smalltalk presented programming language mechanisms for encapsulation and information hiding using the concepts of class and object
- Also precise visibility rules for classes (methods are public, instance variables are private)
- Smalltalk was designed from the start to include as primitive the concept of object rather than grafting it on to an existing language (Everything is an object)
- Designed to be a sort of “total system” which included language, programming environment and a special dedicated machine for program execution



Smalltalk

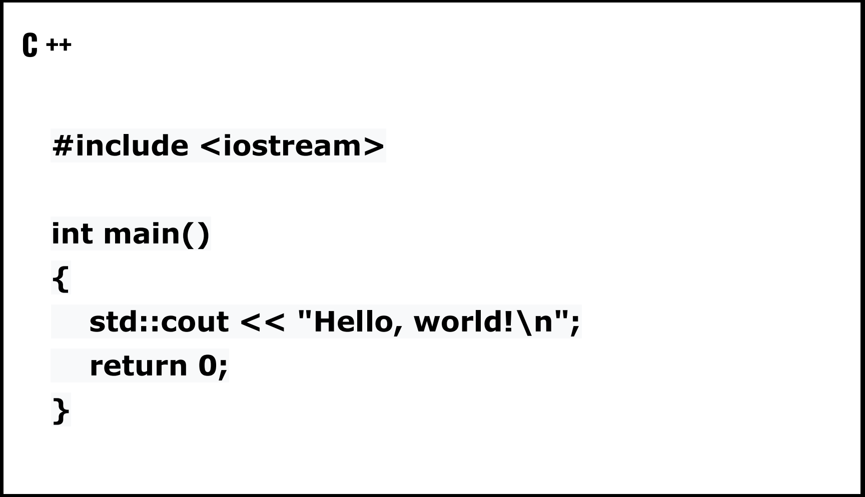
```
Transcript show: 'Hello World!'.
```

Timeline



C++

- C + object-orientation
- The first version of C++ was defined by Bjarne Stroustrup in 1980 at Bell Labs
- Classes and inheritance added to the C language without compromising efficiency and compatibility with the existing C languages
- C had to remain a subset of C++ and as such had to be acceptable to the C++ compiler
- Templates were added (parametric polymorphism)



C ++

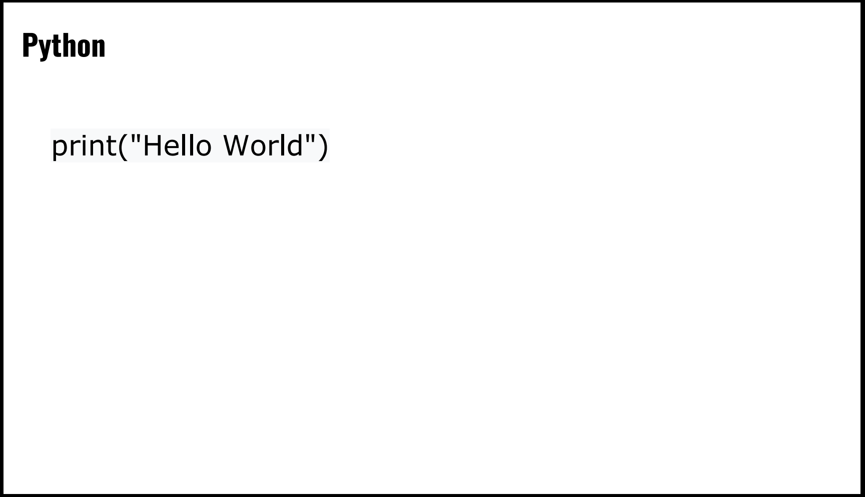
```
#include <iostream>

int main()
{
   std::cout << "Hello, world!\n";
   return 0;
}
```

Python

- Created by Guido van Rossum and first released in 1991
- Python is an interpreted high-level programming language for general-purpose programming
- Object-oriented programming fully supported
- It provides constructs that enable clear programming on both small and large scales
- Most used for automation and web development
- The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

  Beautiful is better than ugly
  Explicit is better than implicit
  Simple is better than complex
  Complex is better than complicated
  Readability counts

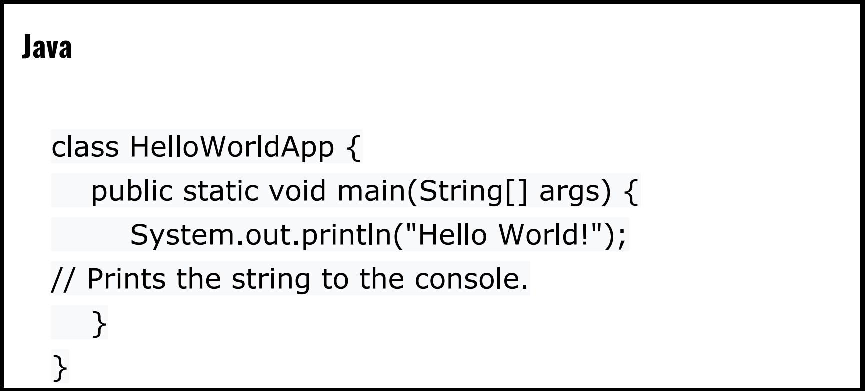


Python

```
print("Hello World")
```

Java

- The initial project, started in 1990 by Sun Microsystems, had the aim of defining a language based on a new implementation of C++, to be used in small computing devices with relatively limited power, connected to a network
- These devices were to implement a kind of browser to be used for navigation through the network
- It was later realised that the language had great potential in the world of the Web
- Two important design requirements
    1- Portability (Java Virtual Machine and associated byte code)
    2- Security (Type safety, avoidance of explicit pointer handling)

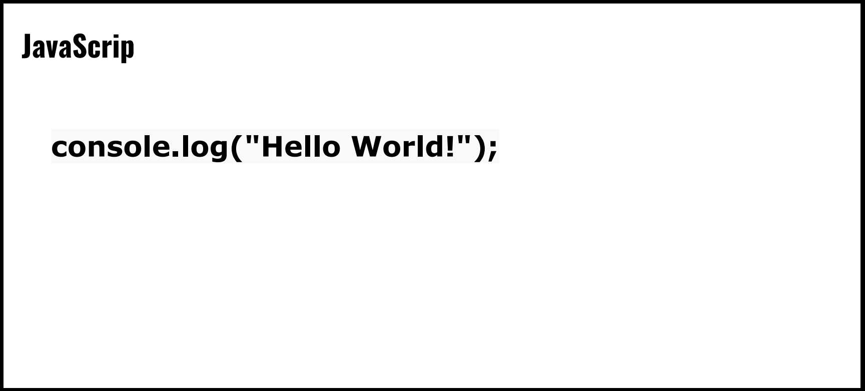


Java

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
// Prints the string to the console.
    }
}
```

JavaScript (Mocha, LiveScript)

- Developed in 1995 by Brendan Eich at Netscape
- Is not related to Java except for the syntax being derived from C
- High-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm
- JavaScript is the most commonly found programming language in the world. Mainly because it is required to be in every web browser. JavaScript is what makes the web dynamic and interactive. It was standardised under the name ECMAScript.



JavaScrip

```
console.log("Hello World!");
```

Concepts

Abstract languages – ignore the physical characteristics of the computer

Backus–Naur form or Backus normal form (BNF) - is a notation technique for context-free grammars, often used to describe the syntax of languages used in computing, such as computer programming languages, document formats, instruction sets and communication protocols

Dynamic method dispatch – selecting the correct method to run a the runtime depending on which object is receiving the corresponding message

Encapsulation – In object oriented programming languages, encapsulation is used to refer to one of two related but distinct notions, and sometimes to the combination thereof: A language mechanism for restricting direct access to some of the object's components

High-level languages – permitted the use of symbolic notation to indicate arithmetic expressions

Imperative language – language based upon a sequence of imperatives

Intermediate code - An Intermediate representation (IR) is the data structure or code used internally by a compiler or virtual machine to represent source code. An IR is designed to be conducive for further processing, such as optimization and translation

Middle-level language – Offers the opportunity to communicate more closely to the underlying machine language (C language)

Object-oriented language (OOP) – is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods

Parametric polymorphism - In programming languages and type theory, parametric polymorphism is a way to make a language more expressive, while still maintaining full static type-safety. Using parametric polymorphism, a function or a data type can be written generically so that it can handle values identically without depending on their type

Sources

https://en.wikipedia.org/wiki/A-0_System

https://en.wikipedia.org/wiki/History_of_compiler_construction

https://en.wikipedia.org/wiki/Object-oriented_programming

https://en.wikipedia.org/wiki/Parametric_polymorphism

https://en.wikipedia.org/wiki/Python_(programming_language)

https://excelwithbusiness.com/blog/say-hello-world-in-28-different-programming-languages/

https://insights.stackoverflow.com/survey/2017#technology-programming-languages
https://tinyurl.com/y7jv4nxf
https://www.techworm.net/2017/02/plankalkul-worlds-first-programming-language-computers-written-1942.html

https://www.youtube.com/watch?v=j8_NfGlta1k

https://www.youtube.com/watch?v=yhoy8jHmCOk

рахмат
danke
謝謝
ngiyabonga
شكراً جزيلاً
spas
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
спасибі
kiitos
dankie
nanni
nandri
dhanyavad
thank you
gracias
хвала
asante
manana
obrigada
tenki
bedankt
bayarlalaa
gràçie
hvala
mauruuru
köszönöm
akun
dankon
aciū
djiere dieuf
tau
mochchakkeram
murakoze
enkosi
dziękuję
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
dēkuji
obrigado
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
rahmet
terima kasih
kam sah hamnida
rahmat
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci