

Lenguajes de Programación

IIC1005

PUC Chile

Avisos Varios

- Subiremos un video tutorial sobre git la próxima semana
- Recibirán el enunciado de tarea chica 1 la próxima semana
- En la ayudantía de la próxima refrescaremos algunos conceptos importantes como la consola (shell/terminal) y sus comandos más importantes, así como git y github

¿Cómo hablarle a un computador?



¿Cómo hablarle a un computador?

Un **lenguaje de programación** es un



lenguaje formal (con reglas gramaticales bien definidas)

que le proporciona al programador la capacidad de escribir (o programar)

una serie de **instrucciones** o secuencias de órdenes en forma de **algoritmos**....



¿Cómo hablarle a un computador?

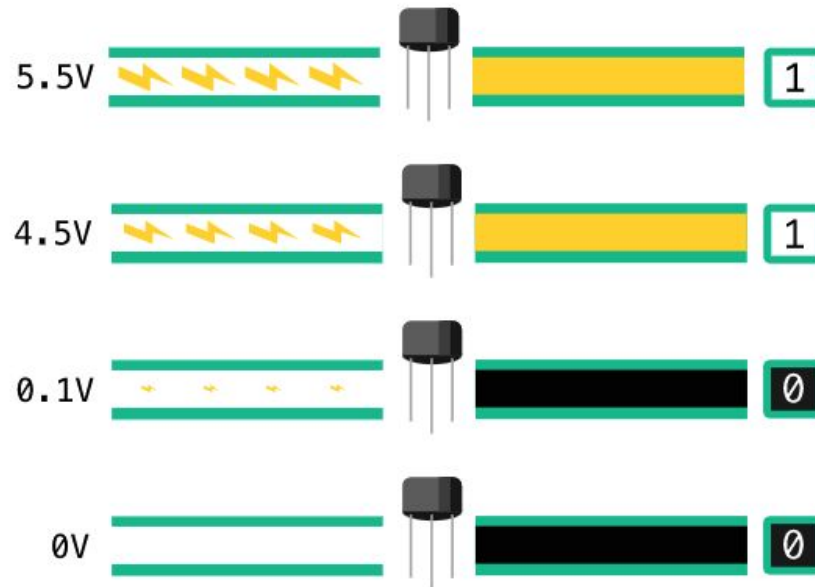


El computador: ¿habla lenguaje Python?

¿Cómo hablarle a un computador?

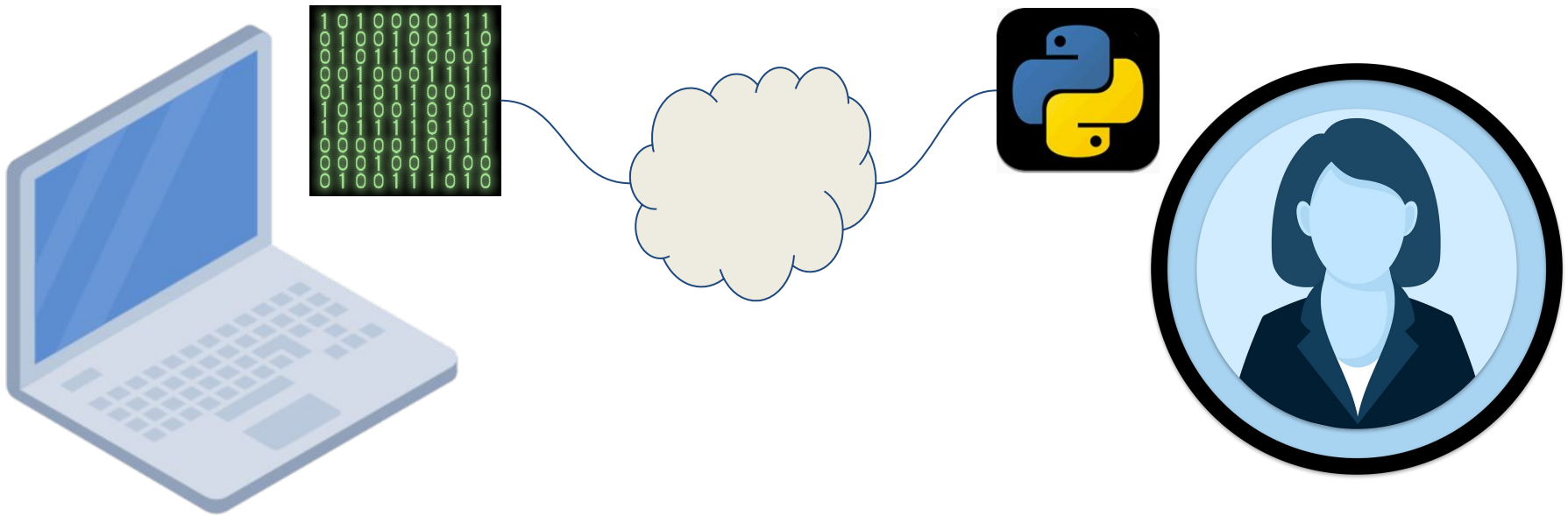


¿Cómo hablarle a un computador?

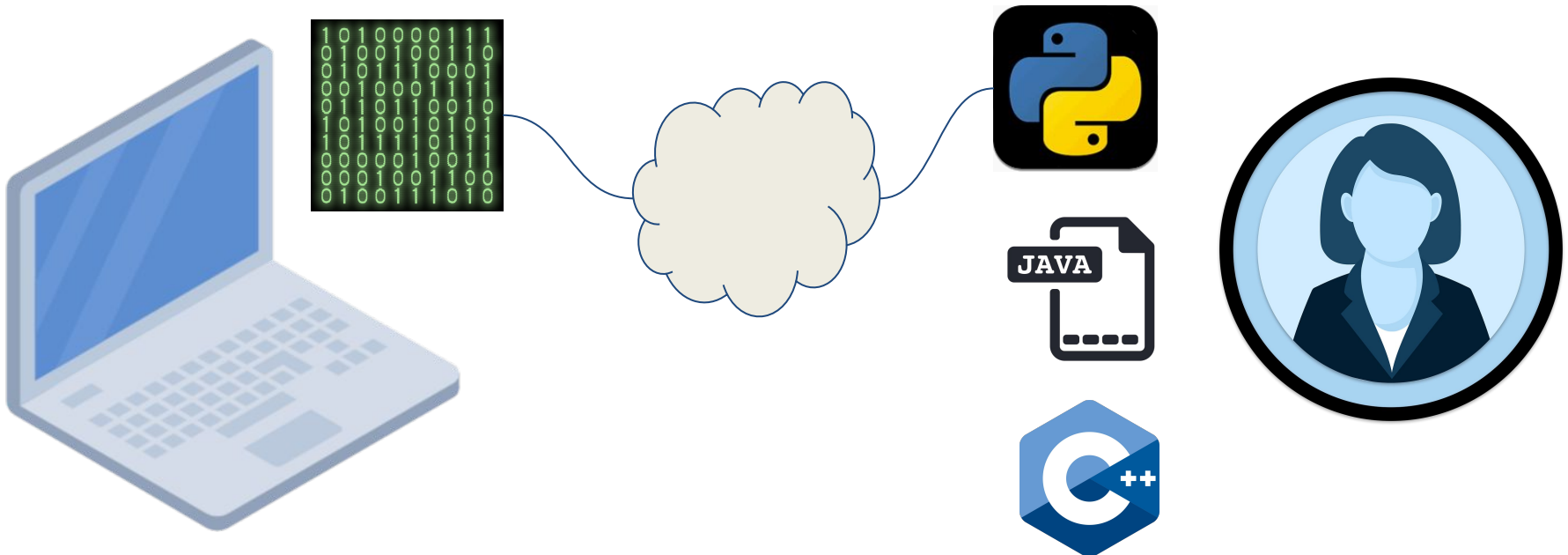


<https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:computers/xcae6f4a7ff015e7d:from-electricity-to-bits/a/from-electricity-to-bits>

¿ Qué ocurre para que mi programa en python termine en los 1s o 0s que entiende el computador, y viceversa?



¿ Qué ocurre para que mi programa en python termine en los 1s o 0s que entiende el computador, y viceversa?



Hoy: Lenguajes de Programación

- Algo de Historia, y luego ...
 - Assembly
 - Fortran
 - C
 - Prolog
 - C++
 - Java / C#
 - Objective-C
 - Python
 - Ruby
 - PHP
 - Javascript
 - R/Matlab (Computacion Cientifica)
 - SQL

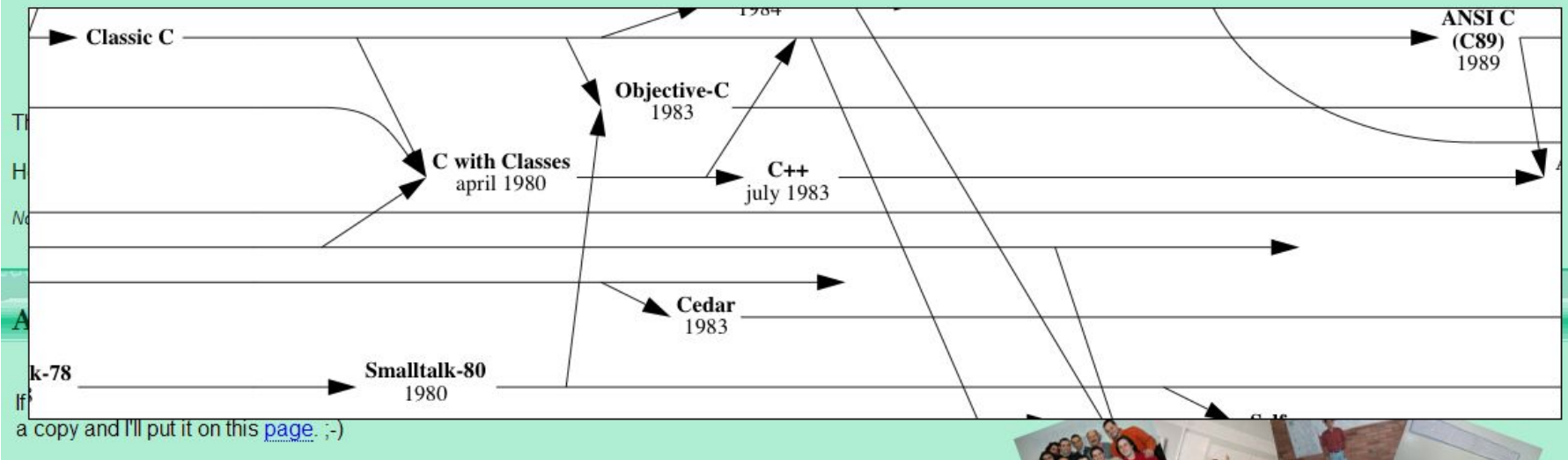
Un poquito de historia

<http://www.levenez.com/lang/>

Below, you can see the preview of the **Computer Languages History** (move on the white zone to get a bigger image):

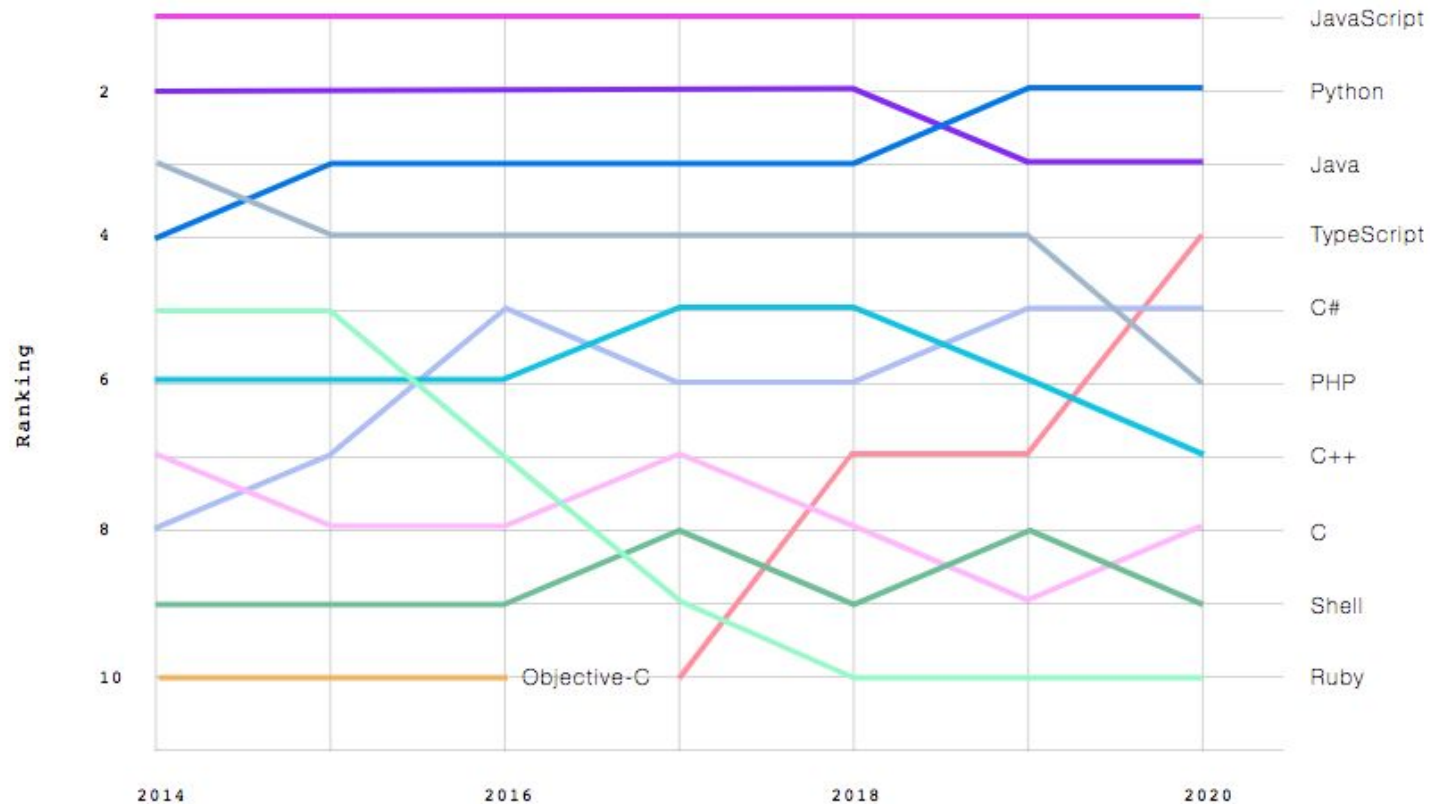


If you want to print this timeline, you can **freely** download one of the following PDF files:



Lenguajes más populares

<https://octoverse.github.com/>



Otros rankings: TIOBE

<https://www.tiobe.com/tiobe-index/>

| Mar 2021 | Mar 2020 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 2 | ⬆️ | C | 15.33% | -1.00% |
| 2 | 1 | ⬇️ | Java | 10.45% | -7.33% |
| 3 | 3 | | Python | 10.31% | +0.20% |
| 4 | 4 | | C++ | 6.52% | -0.27% |
| 5 | 5 | | C# | 4.97% | -0.35% |
| 6 | 6 | | Visual Basic | 4.85% | -0.40% |
| 7 | 7 | | JavaScript | 2.11% | +0.06% |
| 8 | 8 | | PHP | 2.07% | +0.05% |
| 9 | 12 | ⬆️ | Assembly language | 1.97% | +0.72% |
| 10 | 9 | ⬇️ | SQL | 1.87% | +0.03% |
| 11 | 10 | ⬇️ | Go | 1.31% | +0.03% |
| 12 | 18 | ⬆️⬆️ | Classic Visual Basic | 1.26% | +0.49% |
| 13 | 11 | ⬇️ | R | 1.25% | -0.01% |
| 14 | 20 | ⬆️⬆️ | Delphi/Object Pascal | 1.20% | +0.48% |
| 15 | 36 | ⬆️⬆️ | Groovy | 1.19% | +0.94% |
| 16 | 14 | ⬇️ | Ruby | 1.18% | +0.13% |
| 17 | 17 | | Perl | 1.15% | +0.24% |
| 18 | 15 | ⬇️ | MATLAB | 1.04% | +0.05% |
| 19 | 13 | ⬇️⬇️ | Swift | 0.95% | -0.28% |
| 20 | 19 | ⬇️ | Objective-C | 0.91% | +0.17% |

¿Cuánto importa la popularidad?

- Es importante, pero hay lenguajes que parecen poco populares y son muy usados en ciertas áreas:
- En Bancos y grandes compañías: COBOL
- Aplicaciones matemáticas: FORTRAN
- Etc...

Supongan este requerimiento

<<If somebody came to me and wanted to pay me a lot of money to build a large scale message handling system that really had to be up all the time, could never afford to go down for years at a time, I would unhesitatingly choose to build it in.>>

¿Qué lenguaje elegirían?

Supongan este requerimiento

<<If somebody came to me and wanted to pay me a lot of money to build a large scale message handling system that really had to be up all the time, could never afford to go down for years at a time, I would unhesitatingly choose **ERLANG** to build it in.>>

Tim Bray, director of Web Technologies at Sun Microsystems, keynote at OSCON in July 2008

ERLANG

- Primera versión de Erlang implementada en ProLog
- ¿Quién lo usa?
 - Amazon.com: Para su BD SimpleDB
 - WhatsApp: Para soportar el servicio de mensajería, con 2 millones de usuarios conectados por servidor
 - Bet365: Para el servicio de apuestas inPlay
 -

Erlang <-> Relación con ProLog

- ¿Qué tiene de especial Prolog que no tienen otros lenguajes que han usado antes?

```
-module(count_to_ten).  
-export([count_to_ten/0]).  
  
count_to_ten() -> do_count(0).  
  
do_count(10) -> 10;  
do_count(Value) -> do_count(Value + 1).
```

Relación con ProLog

- Hot Code Loading (Hot Swaping)

```
%% A process whose only job is to keep a counter.
%% First version
-module(counter).
-export([start/0, codeswitch/1]).

start() -> loop(0).

loop(Sum) ->
    receive
        {increment, Count} ->
            loop(Sum+Count);
        {counter, Pid} ->
            Pid ! {counter, Sum},
            loop(Sum);
        code_switch ->
            ?MODULE:codeswitch(Sum)
            % Force the use of 'codeswitch/1' from the latest MODULE version
    end.

codeswitch(Sum) -> loop(Sum).
```

PROLOG

- Logic programming language: asociado a lógica e inteligencia artificial.
- Lenguaje declarativo que expresa relaciones y permite realizar inferencias

```
mother_child(trude, sally).
```

```
father_child(tom, sally).
```

```
father_child(tom, erica).
```

```
parent_child(X, Y) :- father_child(X, Y).
```

```
parent_child(X, Y) :- mother_child(X, Y).
```

```
sibling(X, Y) :- parent_child(Z, X), parent_child(Z, Y).
```

?- sibling(sally, erica). Yes

LOGO

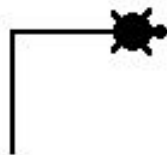
- Creado en 1969 por Seymour Papert, con propósito pedagógico



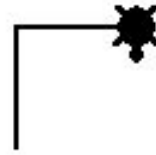
`forward 50`



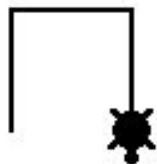
`right 90`



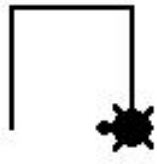
`forward 50`



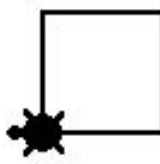
`right 90`



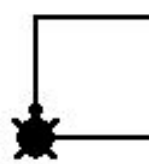
`forward 50`



`right 90`



`forward 50`



`right 90`



Por si alguien quiere probar

- Turtle academy <http://turtleacademy.com/>

[Lessons](#)[User programs](#)[Playground](#)[News](#)[About](#)[English](#)[Login/Sign Up](#)

Turtle Academy

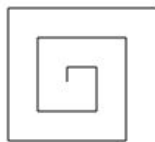
The easy way to learn programming

Turtle Academy makes it surprisingly easy to start creating amazing shapes using the LOGO language

Here are some examples for easy and fun programming

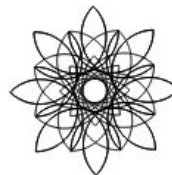
Create a Spiral

```
for [i 10 100 10] [fd :i rt 90]  
ht
```



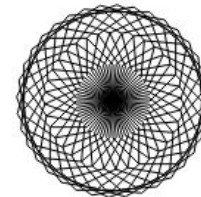
Cool flower

```
repeat 8 [rt 45 repeat 6  
[repeat 90 [fd 1 rt 2] rt 90]]  
ht
```



Crazy octagon

```
cs repeat 36 [ rt 10 repeat  
8 [ fd 25 lt 45]] ht
```



Cuándo fueron creados estos lenguajes?

- Assembly
- FORTRAN
- C
- C++
- Java
- Javascript
- Python
- C#



Cuándo fueron creados estos lenguajes?

- Assembly (1949)
- FORTRAN
- C
- C++
- Java
- Javascript
- Python
- C# (2001)



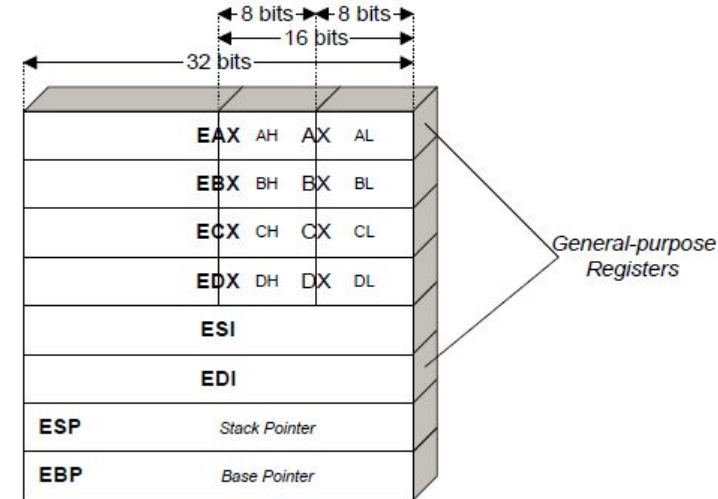
Cuándo fueron creados estos lenguajes?

- Assembly (1949)
- FORTRAN (1954)
- C (1969)
- C++ (1980)
- Java (1995)
- Javascript (1995)
- Python (1991)
- C# (2001)



Ejemplo de Assembly x86

- Así es como sumas dos números:
- Poner primer numero en registro
- Poner segundo numero en registro
- Sumar los registros
- Retornar resultado



```
;n1 db 3; n2 db 7
mov eax, 3 ; podria ser mov eax, [n1]
mov ecx, 7; podria ser mov ecx, [n2]
add eax, ecx
ret
```

```
// equivalente en C
int a = 3;
int c = 7;
a += c;
return a;
```

Considerando la Arquitectura: Assembly

- Es un lenguaje de bajo nivel (muy cerca de la máquina - hardware)
-
- Los lenguajes “Assembly” consideran directamente la arquitectura del equipo (número y tamaño de los registros) y por lo tanto no son portables a otras arquitecturas (como Java, por ejemplo)

Compiladores e Intérpretes

- **Compilador:** analiza el programa fuente y lo traduce a otro equivalente escrito en otro lenguaje (por ejemplo, en el lenguaje de la máquina). Su acción equivale a la de un traductor humano, que toma un libro y produce otro equivalente escrito en otra lengua.
- Ejemplo: Compilador de C o C++

Compiladores e Intérpretes

- **Intérprete:** que analiza el programa fuente y lo ejecuta directamente, sin generar ningún código equivalente. Su acción equivale a la de un intérprete humano, que traduce las frases que oye sobre la marcha, sin producir ningún escrito permanente.
- Ejemplo: Intérprete de Python o Ruby

Compilador vs. intérprete

- **Un intérprete** facilita la búsqueda de errores, pues la ejecución de un programa puede interrumpirse en cualquier momento para estudiar el entorno
- **Un compilador** suele generar programas más rápidos y eficientes, ya que el análisis del lenguaje fuente se hace una sola vez, durante la generación del programa equivalente.

Compiladores

- ¿Quién escribió el primer compilador y cuándo?

Compiladores

- ¿Quién escribió el primer compilador ?
 - Grace Hopper, científica estadounidense.
 - El Sistema A-0 fue escrito por Grace Hopper en 1951 y 1952 para UNIVAC I. Fue el primer compilador desarrollado para una computadora electrónica.
 - Luego inventó el B-O y el FLOWMATIC, del que se deriva el lenguaje COBOL.

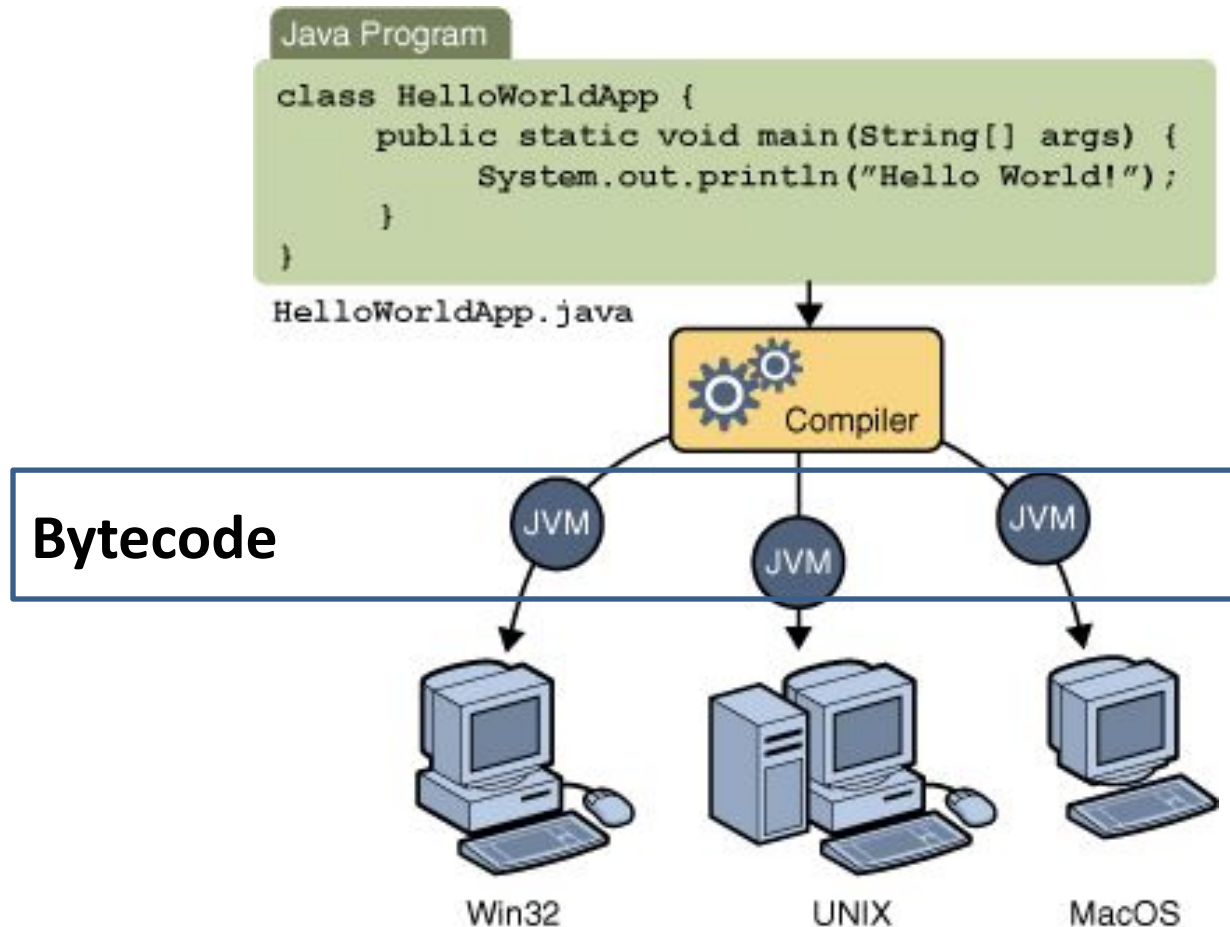


Primera diferencia: Lenguaje Compilado versus Interpretado

- Compilado: FORTRAN, Pascal, C, C++
- Interpretado: Python, Ruby
- Compilado es generalmente más rápido porque apuntan directamente a la máquina/arquitectura en la cual se ejecutan.
- Interpretado tiende a ser más portable.
- Versión de lenguajes “interpretados” es más fáciles de crear.

¿Cómo funciona JAVA?

- Con la Java Virtual Machine



POO: Hello World en C++ y Java

===== C++ =====

```
#include <iostream>
```

```
int main() {
```

```
    //comentario
```

```
    std::cout << "Hello World!";
```

```
}
```

===== JAVA =====

```
public class HelloWorld { //comentario
```

```
    public static void main(String[] args) {
```

```
    System.out.println("Hello, World");
```

```
    }
```

```
}
```

C / C++ / Java

Manejo de memoria: C

```
1#include <stdlib.h> // needed for malloc and free!  
2int *p_int = malloc(sizeof(*p_int));  
3// use p_int  
4free( p_int );
```

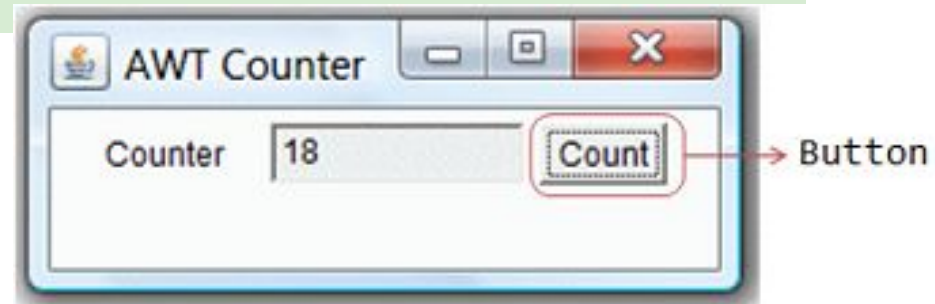
Manejo de memoria: C++

```
1int *p_int = new int;  
2// use p_int  
3delete p_int;
```

Manejo de memoria Java: Garbage Collector!

Java

```
import java.awt.Frame;  
// Using Frame class in package java.awt  
// A GUI program is written as a subclass of Frame - the top-level container  
// This subclass inherits all properties from Frame, e.g., title, icon, buttons, content-pane  
public class MyGUIProgram extends Frame {  
    // Constructor to setup the GUI components  
    public MyGUIProgram() {  
        .....  
    }  
  
    // Other methods .....  
    // The entry main() method  
    public static void main(String[] args) {  
        // Invoke the constructor (to setup the GUI) by allocating an instance  
        new MyGUIProgram();  
    }  
}
```



SQL

- **Structured Query Language** orientado especialmente para DBMS relacionales
- Cuando la información está almacenada de forma estructurada, SQL es el estándar para consultas.

Paradigmas de Programación

- Lenguajes más comunes responden a varios paradigmas

| <u>Paradigm</u> | Description | Main characteristics | Related paradigm(s) | Critics | Examples |
|-------------------|--|---|------------------------|---|--|
| <u>Imperative</u> | Computation as <u>statements</u> that <i>directly</i> change a program <u>state</u> (<u>data fields</u>) | Direct <u>assignments</u> , common <u>data structures</u> , <u>global variables</u> | | <u>Edsger W. Dijkstra</u> , <u>Michael A. Jackson</u> | <u>C</u> , <u>C++</u> , <u>Java</u> , <u>PHP</u> , <u>Python</u> |
| <u>Structured</u> | A style of <u>imperative programming</u> with more logical program structure | <u>Structograms</u> , <u>indentation</u> , either no, or limited use of, <u>goto</u> statements | Imperative | | <u>C</u> , <u>C++</u> , <u>Java</u> |
| <u>Procedural</u> | Derived from structured programming, based on the concept of <u>modular programming</u> or the <i>procedure call</i> | <u>Local variables</u> , sequence, selection, <u>iteration</u> , and <u>modularization</u> | Structured, imperative | | <u>C</u> , <u>C++</u> , <u>Lisp</u> , <u>PHP</u> , <u>Python</u> |
| <u>Functional</u> | Treats <u>computation</u> as the evaluation of <u>mathematical functions</u> avoiding <u>state</u> and <u>mutable data</u> | <u>Lambda calculus</u> , <u>compositionality</u> , <u>formula</u> , <u>recursion</u> , <u>referential transparency</u> , no <u>side effects</u> | | | <u>Erlang</u> , <u>Haskell</u> , <u>Lisp</u> , <u>Clojure</u> , <u>Scala</u> , <u>F#</u> |

Por que sería útil forzar programación funcional? $rt(x) = rt(y)$ if $x = y$

```
globalValue = 0;

integer function rq(integer x)
begin
    globalValue = globalValue + 1;
    return x + globalValue;
end

integer function rt(integer x)
begin
    return x + 1;
end
```


Si escribiéramos un loop con la $F(x)$

- Un compilador podría detectar una función dentro de un loop y optimizar la forma de referenciarla y escribirla en código de máquina SOLAMENTE si la función no depende del estado de ejecución del programa

```
While (i < 1000)  
  rq(i)
```

Depende de variable global
definida en tiempo de ejecución

```
While (i < 1000)  
  rt(i)
```

Este sí cumple con
transparencia referencial

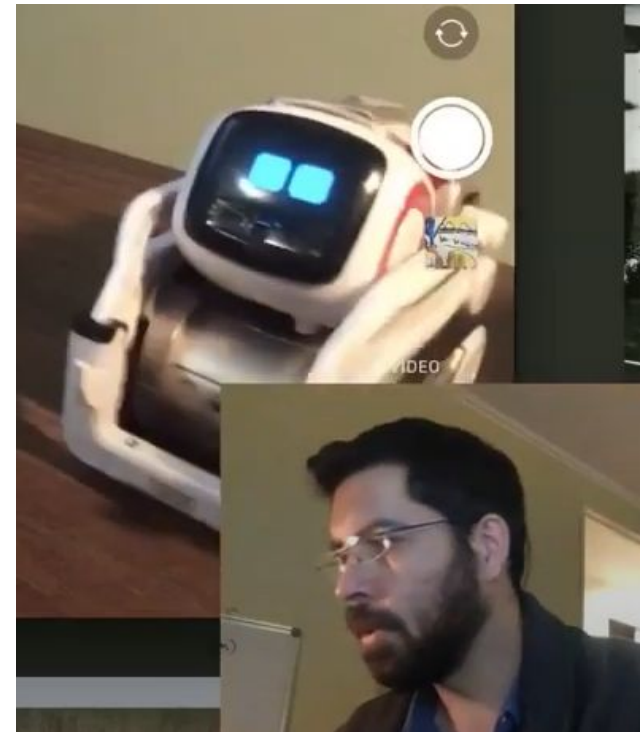
COZMO adventure time!

Cozmo iphone app

<https://youtu.be/kt-KYA7YAqg>

Python API cozmo

<https://youtu.be/AlKf4WlgK9s>



Volviendo a la realidad

- La tarea 1 grande incluye
- Incluye programación en:
 - Python
 - HTML
 - CSS
 - Javascript


¿Cómo prepararme rápidamente?

- Venir a ayudantías
- CodeCademy: Basic Web Projects
 - <https://www.codecademy.com/en/tracks/projects>
- W3Schools: tutorial javascript
 - <https://www.w3schools.com/js/default.asp>
- Libros de referencia:
 - Guia paso a paso: [You Don't Know Javascript](#)
 - Tradicional: [Javascript The Definitive Guide](#)

Gracias!

Otros lenguajes - LoLCode

- Inspirado en lolspeak
- Extension de archivo: .lol , .lols



Now with doge!

LOLCODE is an esoteric programming language inspired by the funny things that cats say on the Internet.

[Learn more about the language](#)

lci is a correct, portable, fast, and precisely documented interpreter for LOLCODE written in C.

[Download source](#)

[GitHub project page](#)

Problems? Check out the [mailing list](#) or file a [bug report](#).

LOLCODE

Hagamos un “Hola Mundo” en LOLCODE

LOLCODE 1

HAI 1.2

VISIBLE "Hai world"

KTHXBYE

LOLCODE 2

- Declarar e inicializar una variable
- Mostrarla en Pantalla

I HAS A VARIABLE ITZ <var>

LOLCODE 3

- Agregar comentarios

BTW

LOLCODE 4

- Solicitar al usuario input desde teclado

GIMMEH

LOLCODE 5

- Una bifurcación (IF)

..., O RLY?

YA RLY

...

NO WAI

...

OIC

LOLCODE 6

- CONTADOR

IM IN YR LOOP

...

IM OUTTA YR LOOP

...otros

- Switch... case

<expression>

WTF?

OMG <value literal>

<code block>

[OMG <value literal>

<code block> ...]

[OMGWTF

<code block>]

OIC