

# Introducción a la programación



# ¿Qué es programar?

Programar consiste en crear una serie de **órdenes** dirigidas a un ordenador o dispositivo para que lleve a cabo las **tareas** que le encomendemos, con el objetivo de solucionar un determinado problema.

La programación es el instrumento que permite la ejecución de las **tareas automatizadas** de un sistema informático.

Se utilizan los lenguajes de programación, a través de las cuales se crearán los **programas** que **contienen las instrucciones** que se dan a la máquina para que ésta ejecute determinada acción.

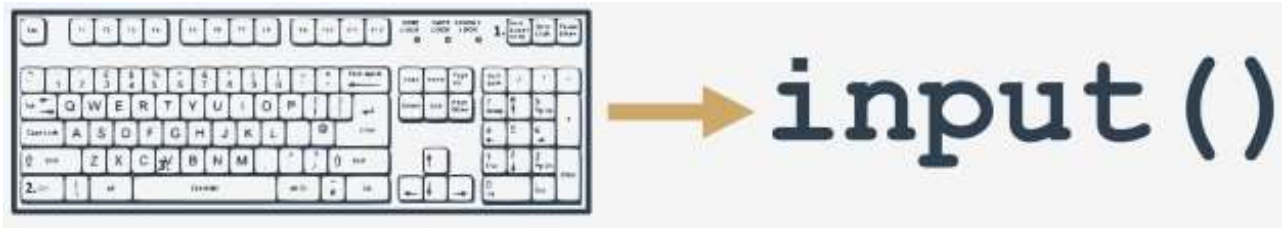
# ¿Qué es un programa?

- Un programa es una secuencia de instrucciones que han sido escritas en un lenguaje de programación concreto, entendibles por el ordenador, y que permiten realizar un trabajo o resolver un problema.
- Si no sabemos resolver este problema, no podremos escribir el programa. De ahí que un buen programador siempre comienza su trabajo con un análisis y comprensión exhaustivos del problema que pretende resolver.

Las formas de resolver un mismo problema no son únicas, ante un problema un programador podrá escribir diferentes programas que solucionen de distinta forma el problema planteado. Pese a esto lo que sí podemos afirmar es que, en la codificación de cualquier programa, de forma general, se pueden distinguir las siguientes partes:



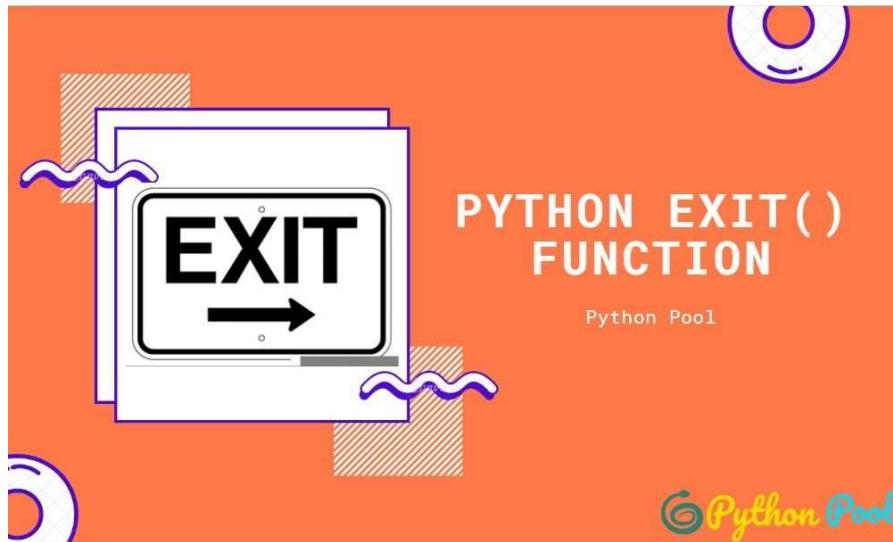
- Entrada de datos: instrucciones que recogen datos de un dispositivo o periférico para ser almacenados en la memoria principal a la espera de su proceso posterior.



Comando input para introducir datos del teclado

- Proceso: parte del programa que engloba las instrucciones encargadas de procesar los datos recogidos anteriormente. Los resultados se almacenan de nuevo en la memoria principal.

- Salida de resultados: instrucciones del programa que recogen los resultados obtenidos en la fase anterior y los envían a los dispositivos, periféricos, etc de salida de la información.



# Lenguajes de programación

Podemos definir lenguaje de programación como un **conjunto de caracteres y reglas** que permiten crear un programa , para introducir y tratar la información en un ordenador. Son lenguajes artificiales creados con el objetivo de conseguir la comunicación entre el humano y la máquina, o entre máquinas.

Se pueden clasificar atendiendo a varios criterios:



# Clasificación según nivel de abstracción (de menor a mayor abstracción)

- Lenguaje máquina: Es el lenguaje de programación que entiende directamente el ordenador. Utiliza el **sistema binario**, es decir, únicamente utiliza dos dígitos, el **0** y el **1**. Con estos dos únicos dígitos, conocidos como bits, forma las cadenas binarias (combinaciones de ceros y unos) con las que la CPU de una máquina procesa la información.

```
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
11001010 11001010 11110101 00101011
11001010 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
```



Cualquier cosa que queramos que realice el ordenador debe expresarse con este código de ceros y unos, que **es el lenguaje que realmente entiende y utiliza el microprocesador** de una ordenado o dispositivos. **El lenguaje máquina fue el primer lenguaje de programación.**

Los primeros programadores programaban en lenguaje máquina, con números binarios, pero la complejidad de las operaciones realizadas y la facilidad de cometer errores derivó en la necesidad de crear lenguajes de programación más parecidos al lenguaje humano, más sencillos y rápidos de manejar y de entender.

De esta forma, para escribir un programa de forma más sencilla y fácil de recordar, se utilizan los **lenguajes de bajo y alto nivel**. En ellos la sucesión de unos y ceros es sustituida por letras o palabras, mucho más fáciles de recordar y utilizar, por su mayor semejanza con el lenguaje humano, generalmente el inglés.

- Lenguajes de Programación de Bajo Nivel (de abstracción):  
 Lenguaje **Ensamblador**, del inglés Assembler.

A este grupo pertenecen los **primeros lenguajes creados intentando sustituir el lenguaje máquina por uno más similar utilizado por el hombre**. Se denominan frecuentemente lenguajes ensambladores

```

INICIO  BSF      STATUS,RPO      ; ENTRA EN EL BANCO 1
        MOVLW   0X1F             ; MUEVE 1Fh A W
        MOVWF   TRISA            ; CONFIGURA EL PUERTO A COMO

ENTRADA
        MOVLW   0X00             ; MUEVE 00 A W
        MOVWF   TRISB            ; CONFIGURA EL PUERTO B COMO SALIDA
        BCF     STATUS,RPO       ; SALE DEL BANCO 1
        CLRF    PORTB            ; LIMPIA EL PUERTO B

OTRA
CERO    BTFSC    PORTA,4          ; PREGUNTA SI EL PIN 4 DEL PUERTO A I

        GOTO     OTRA            ; SALTA A LA ETIQUETA OTRA
        MOVLW   0X18             ; MUEVE 18h A W
        MOVWF   PORTB            ; CARGA W EN EL PUERTO B
        CALL    RETARDO          ; LLAMA A LA ETIQUETA RETARDO
        BTFSC    PORTA,4          ; PREGUNTA SI EL PIN 4 DEL PUERTO A I

CERO    GOTO     OTRA1           ; SALTA A LA ETIQUETA OTRA1

        MOVLW   0X24             ; MUEVE 24h A W
        MOVWF   PORTB            ; CARGA W EN EL PUERTO B
        CALL    RETARDO          ; LLAMA A LA ETIQUETA RETARDO
  
```

Programar en ensamblador  
 es MUY DIFÍCIL

Más fáciles de utilizar que el lenguaje máquina, pero **son específicos de cada procesador**, lo que obliga a reescribir los programas hechos con este tipo de lenguaje si nos llevamos el programa a otro ordenador (poca portabilidad).

Constituidos por instrucciones de lenguaje máquina escritas de forma simbólica, mediante códigos nemotécnicos (grupos de 3-4 caracteres que simbolizan órdenes o tareas a realizar) fáciles de leer y recordar (ejemplo: ADD (sumar), MOV (mover), ...).

Un programa escrito en este lenguaje no es directamente ejecutable en la máquina. Debe de ser traducido a un programa equivalente en lenguaje de máquina, capaz de ser ejecutado.



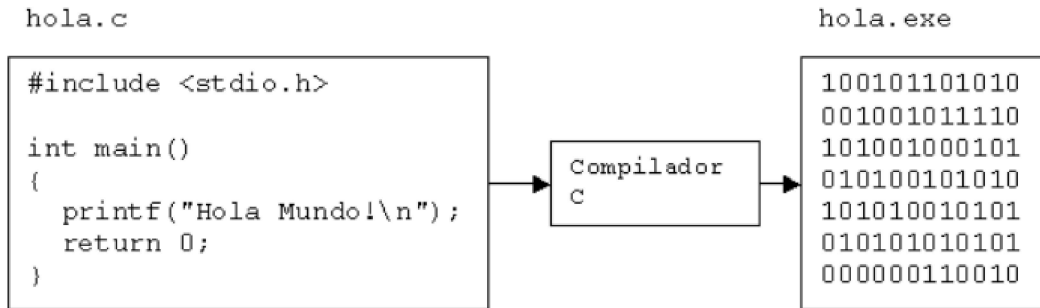
- Lenguajes de Programación de Alto Nivel: son más fáciles de aprender y utilizar porque son los que más se parecen al lenguaje natural, ya que se usan palabras o comandos del lenguaje humano (generalmente inglés). Por ello son los más utilizados por los programadores. Además los lenguajes de alto nivel son independientes de la máquina, por lo que se pueden usar en cualquier computador con muy pocas modificaciones o sin ellas.

- Existen muchos lenguajes de programación de alto nivel con sus diferentes versiones y lenguajes derivados. Entre ellos tenemos el HTML, , C, C++, SQL, JAVA, FORTRAN, Python, Pascal, LOGO, Processing, ...
- El primero es FORTRAN que significa Traductor de Formulas (del inglés FORMula TRANslating) desarrollado investigadores de IBM en los años 50.
- Los lenguajes de alto nivel necesitan un programa **compilador** o un programa **intérprete** que lo traduzca a lenguaje máquina.



El **compilador** y el **intérprete** son programas cuya finalidad es "traducir" las instrucciones enviadas a una máquina en un lenguaje de alto nivel a un lenguaje que el equipo pueda entender, es decir, lenguaje máquina.

**Compiladores:** Programas especiales que procesan las instrucciones escritas en un lenguaje de programación de alto nivel y las convierte a lenguaje de máquina. El compilador traduce el programa completo a código máquina antes de empezar a ejecutarlo, a lo que se llama **código objeto**, y si no hay errores, genera un archivo ejecutable.



Los compiladores son programas muy grandes que permiten la comprobación de errores. Una vez compilado el programa, podemos ejecutarlo, es decir, hacer que se inicie la carga del programa y comiencen a realizarse las acciones programadas.

### Programa en alto nivel (código fuente)

Inicio  
 $x=1$   
 $y=1$   
resultado =  $x+y$   
Mostrar resultado  
Fin



### Ventajas de un lenguaje compilado:

- a) El programa ejecutable no incluye el código fuente, por lo tanto, el código compilado es más seguro.
- b) Un programa compilado se ejecuta más rápidamente que un programa que está siendo interpretado ya que la traducción se realiza directamente a código máquina.

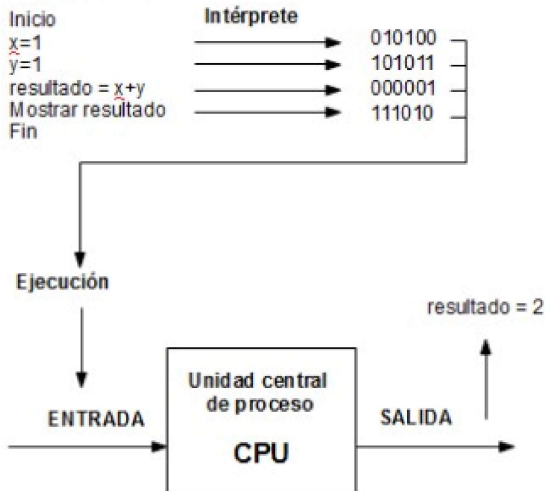
### Las desventajas:

- a) El código objeto necesita crearse para obtener el archivo ejecutable final, y si el programa es muy largo, el proceso de compilación puede ser muy lento.
- b) El código fuente debe estar libre de errores para que el archivo ejecutable se genere.
- c) El archivo ejecutable generado por el compilador solo funcionará en el sistema operativo en el que fue creado.



**Intérpretes:** Un intérprete es un programa que, en tiempo de ejecución, transcribe a lenguaje máquina y ejecuta una por una, en la secuencia descrita en el programa fuente, el conjunto de instrucciones del mismo (sin generar un programa objeto). De esta forma, cada vez que se ejecuta el programa fuente, éste deberá ser interpretado de nuevo; cada vez que se ejecuta una sentencia ésta debe ser interpretada de nuevo.

**Programa en alto nivel  
(código fuente)**



El programa intérprete traduce y ejecuta las instrucciones del programa línea a línea, siguiendo la secuencia real del programa. Cuando detecta un error, la traducción y ejecución del programa se detienen, e indica un código de error.

Las ventajas del utilizar un intérprete son:

- a) Al ejecutarse línea a línea, resulta ser más fácil de depurar y comprobar errores.
- b) Si el programa es muy largo, un intérprete resulta ser más rápido que un compilador porque no tiene que traducir todo el programa para ejecutarlo.

Las desventajas de utilizar un intérprete son:

- a) El código fuente se requiere para poder ejecutar un programa.
- b) Los intérpretes son más lentos que los programas compilados o ejecutables