

# HolaMundo

---

## HolaMundo en varios lenguajes de programación

En este documento se indican los pasos que deberemos seguir para ejecutar código en distintos lenguajes de programación.

En el caso de lenguajes interpretados se indica también la forma de generar un script ejecutable.

El código fuente para algunos lenguajes de programación está disponible en:

- [https://es.wikipedia.org/wiki/Hola\\_mundo#Ejemplos](https://es.wikipedia.org/wiki/Hola_mundo#Ejemplos)

Lenguajes usados:

- Bash
- Python
- PHP
- Javascript (nodejs)
- C
- C++
- Java
- Ensamblador (nasm)
- Ruby
- Go
- Rust
- Lisp

Para tener soporte para los lenguajes anteriores, debemos instalar los siguientes paquetes:

En linux, inicio -> terminal

```
sudo apt install python3 php nodejs gcc g++ openjdk-17-jdk nasm ruby  
golang rustc clisp
```

---

## Bash

### Pasos

1. Ejecutamos el intérprete.  
Este paso no es necesario aquí, puesto que ya estamos en el terminal.
2. Escribimos las sentencias y luego pulsamos INTRO.

```
echo "Hola Mundo"
```

3. Para salir del intérprete pulsamos CTRL+D.

No pulsar esta combinación si no queremos cerrar el terminal.

## Script ejecutable

1. Editamos archivo **hola.sh**

```
sudo nano hola.sh  
# esto abrirá un editor de texto sencillo
```

En el archivo, escribimos:

```
#!/usr/bin/env bash  
  
echo "Hola mundo"
```

- Para salir del editor de texto nano: **Ctrl + x**
- Aceptar con **Y** o **S**
- Pulsa **Enter** para mantener el nombre del archivo

2. Damos permisos de ejecución

```
chmod +x hola.sh
```

3. Ejecutamos

```
./hola.sh
```

---

## Python

### Pasos

1. Ejecutamos el intérprete.

```
python3
```

2. Escribimos las sentencias y luego pulsamos INTRO.

```
print ("Hola Mundo")
```

3. Para salir del intérprete pulsamos CTRL+D.

## Script ejecutable

1. Editamos archivo **hola.py** con nano:

```
#!/usr/bin/env python3  
  
print ("Hola mundo")
```

2. Damos permisos de ejecución

```
chmod +x hola.py
```

3. Ejecutamos

```
./hola.py
```

---

## PHP

### Pasos

1. Ejecutamos el intérprete.

```
php -a
```

2. Escribimos las sentencias.

```
echo "Hola mundo\n";
```

3. Para salir del intérprete pulsamos CTRL+D.

## Script ejecutable

1. Editamos archivo **hola.php** con nano:

```
#!/usr/bin/env php

<?php
    echo "Hola mundo\n"
?>
```

## 2. Damos permisos de ejecución

```
chmod +x hola.php
```

## 3. Ejecutamos

```
./hola.php
```

---

# Javascript (nodejs)

## Pasos

### 1. Ejecutamos el intérprete.

```
node
```

### 2. Escribimos las sentencias y luego pulsamos INTRO.

```
console.log('Hola mundo');
```

### 3. Para salir del intérprete pulsamos CTRL+D.

## Script ejecutable

### 1. Editamos archivo **hola.js**

```
#!/usr/bin/env node

console.log('Hola mundo');
```

### 2. Damos permisos de ejecución

```
chmod +x hola.js
```

### 3. Ejecutamos

```
./hola.js
```

---

## C

### Pasos

#### 1. Editamos archivo **hola.c**

```
#include <stdio.h>

int main()
{
    printf("¡Hola, mundo!");
    return 0;
}
```

#### 2. Compilamos y enlazamos

```
gcc -o hola hola.c      # Compilamos y enlazamos
```

#### 3. Ejecutamos

```
./hola                  # Ejecutamos
```

---

## C++

### Pasos

#### 1. Editamos archivo **hola.cpp**

```
#include <iostream>

using namespace std;

int main()
{
    cout << "¡Hola, mundo!" << endl;
}
```

```
    return 0;  
}
```

## 2. Compilamos y enlazamos

```
g++ -o hola hola.cpp    # Compilamos y enlazamos
```

## 3. Ejecutamos

```
./hola    # Ejecutamos
```

---

# Java

## Pasos

### 1. Editamos archivo **Hola.java**

```
class Hola  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hola Mundo");  
    }  
}
```

### 2. Compilamos

```
javac Hola.java    # Compilamos
```

### 3. Interpretamos y ejecutamos

```
java Hola    # Interpretamos y ejecutamos
```

# Ensamblador (nasm)

## Pasos

### 1. Editamos archivo **hola.asm**

```
section .data

msg      db "¡Hola Mundo!", 0Ah
len      equ    $ - msg

section .text

global _start

_start:
    mov     eax, 04h
    mov     ebx, 01h
    mov     ecx, msg
    mov     edx, len
    int     80h
    mov     eax, 01h
    mov     ebx, 00h
    int     80h
```

## 2. Ensamblamos y enlazamos

```
nasm -f elf64 hola.asm      # Ensamblado para ELF64
ld hola.o -o hola          # Enlazado y generación de ejecutable
```

## 3. Ejecutamos

```
./hola                      # Ejecución
```

---

# Ruby

## Pasos

### 1. Ejecutamos el intérprete.

```
ruby
```

### 2. Escribimos las sentencias y luego pulsamos INTRO.

```
puts "Hola Mundo"
```

### 3. Para salir del intérprete pulsamos CTRL+D.

## Script ejecutable

### 1. Editamos archivo **hola.rb**

```
#!/usr/bin/env ruby  
  
puts "Hola Mundo"
```

### 2. Damos permisos de ejecución

```
chmod +x hola.rb
```

### 3. Ejecutamos

```
./hola.rb
```

---

## Go

### Pasos

#### 1. Editamos archivo **hola.go**

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("Hola mundo desde Go")  
}
```

#### 2. Compilamos y enlazamos

```
go build hola.go # Compilamos y enlazamos
```

#### 3. Ejecutamos

```
./ hola # Ejecutamos
```

#### 4. Si lo deseamos, en Go también es posible interpretar el código



```
go run hola.go      # Interpretamos y ejecutamos
```

---

## Rust

### Pasos

1. Editamos archivo **hola.rs**

```
fn main() {  
    println!("¡Hola, mundo! Desde RUST ");  
}
```

2. Compilamos y enlazamos

```
rustc hola.rs      # Compilamos y enlazamos
```

3. Ejecutamos

```
./hola             # Ejecutamos
```

---

## Lisp

### Pasos

1. Ejecutamos el intérprete.

```
clisp
```

2. Escribimos las sentencias y luego pulsamos INTRO.

```
(format t "¡Hola, mundo!")
```

3. Para salir del intérprete pulsamos CTRL+D.

### Script ejecutable

1. Editamos archivo **hola.lisp**

```
#!/usr/bin/env clisp  
  
(format t "~;Hola, mundo!")
```

## 2. Damos permisos de ejecución

```
chmod +x hola.lisp
```

## 3. Ejecutamos

```
./hola.lisp
```