

# **CLASIFICACIÓN DESCRIPTIVA DE LENGUAJES DE PROGRAMACIÓN**

**JESÚS TERINO  
RODRIGUEZ**

# JAVASCRIPT

Tipo: Alto nivel, Interpretado,  
Multipropósito



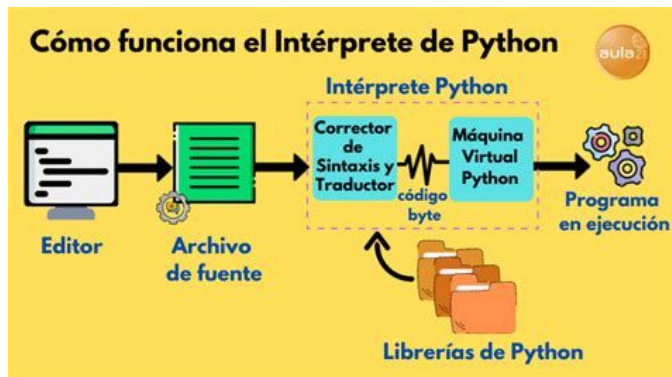
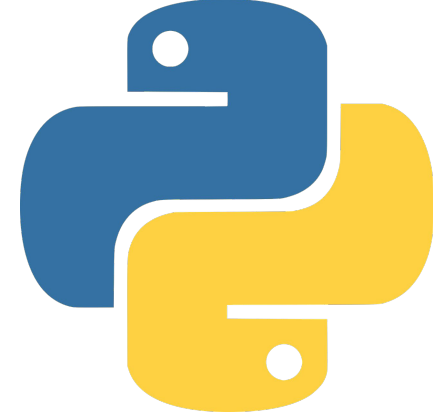
Uso: Desarrollo web, aplicaciones  
de escritorio y móviles, scripting  
en el lado del servidor



Características: Lenguaje de script para páginas web, permite interactividad en páginas web, ampliamente utilizado en desarrollo de aplicaciones web.

# PYTHON

Tipo: Alto nivel, Interpretado,  
Multipropósito



Uso: Ciencia de datos, desarrollo web,  
automatización de tareas, inteligencia  
artificial

Características: Sintaxis clara y legible, fácil de aprender, gran cantidad de librerías y frameworks.



Tipo: Alto nivel, Compilado, Orientado a Objetos



```
1 public class CreateAMethodInJava {  
2  
3     public static void main (String[] args) {  
4         // Call method with numeric literals  
5         int area=computeAreaOfRectangle(10,15);  
6         System.out.println("Area of rectangle (using literal parameters) is " + area);  
7         // Call method with variables  
8         int height=10;  
9         int width=15;  
10        area=computeAreaOfRectangle(height,width);  
11        System.out.println("Area of rectangle (using variables) is " + area);  
12    }  
13  
14    // Method definition  
15    public static int computeAreaOfRectangle (int h, int w) {  
16        return h * w;  
17    }  
18 }
```

Uso: Desarrollo de aplicaciones empresariales, aplicaciones web, Android

Características: Plataforma portátil (Write Once, Run Anywhere - WORA), robusto y eficiente en recursos.



Tipo: Alto nivel, Compilado, Orientado a Objetos

```
public static void Main()
{
    string f = @"File.txt";
    using (StreamWriter sw = File.CreateText(f))
    {sw.WriteLine("Some text");}
    using (StreamWriter sw = File.AppendText(f))
    {sw.WriteLine("Extra Text");}
    using (StreamReader sr = File.OpenText(f))
    { string s = "";
      while ((s = sr.ReadLine()) != null)
      {Console.WriteLine(s);}
    }
}
```

Uso: Desarrollo de aplicaciones de escritorio, móviles, juegos y servicios en la nube

Características: Lenguaje multiparadigma desarrollado por Microsoft, con una sintaxis similar a Java, ampliamente utilizado en la plataforma .NET.



# RUGBY

Tipo: Alto nivel, Interpretado, Orientado a Objetos



Uso: Desarrollo web, scripting, automatización

```
ana@ana-HP-Pavilion-dm4-Notebook-PC: ~  
ana@ana-HP-Pavilion-dm4-Notebook-PC:~$ irb  
2.4.0 :001 > a=[]  
=> []  
2.4.0 :002 > a.push(3)  
=> [3]  
2.4.0 :003 > a.push(3, 'a')  
=> [3, 3, "a"]  
2.4.0 :004 > a[2]  
=> "a"  
2.4.0 :005 > a.first  
=> 3  
2.4.0 :006 > 
```

Características: Conocido por su sencillez y legibilidad, Ruby es el lenguaje detrás del popular framework web Ruby on Rails.

Lenguaje	Tipo	Uso Principal	Características	Popularidad
JavaScript	Alto nivel, Interpretado	Web, Escritorio, Móvil	Interactividad web	Muy Alto
Python	Alto nivel, Interpretado	Ciencia de Datos, Web	Sintaxis clara	Muy Alto
Java	Alto nivel, Compilado	Empresarial, Web, Android	WORA, Orientado a Objetos	Muy Alto
C#	Alto nivel, Compilado	Escritorio, Móvil, Juegos, Nube	Multiparadigma, .NET	Alto
Ruby	Alto nivel, Interpretado	Web, Scripting	Sencillez, Legibilidad	Moderado
COBOL	Alto nivel, Compilado	Transacciones, Empresarial	Sistemas legados	Bajo
Lisp	Alto nivel, Interpretado	Inteligencia Artificial	Programación Funcional	Moderado
Fortran	Alto nivel, Compilado	Cálculo Numérico	Computación Científica	Moderado
Prolog	Alto nivel, Interpretado	Inteligencia Artificial, Sistemas Expertos	Programación Lógica	Bajo
Perl	Alto nivel, Interpretado	Scripting, Procesamiento de Texto	Procesamiento de Texto	Moderado

# COBOL

Tipo: Alto nivel, Compilado, Comercial



```
000024  
000025 PROCEDURE DIVISION.  
000026 0001-MAIN.  
000027     INSPECT FUNCTION REVERSE(STR-1)  
000028         TALLYING WS-LEN1 FOR LEADING SPACES.  
000029     COMPUTE WS-LEN = LENGTH OF STR-1 - WS-LEN1.  
000030     DISPLAY WS-LEN.  
000031     MOVE 1 TO I.  
000032     MOVE WS-LEN TO J.  
000033     PERFORM REV-PARA WS-LEN TIMES.  
000034     DISPLAY STR-1.  
000035     DISPLAY STR-2.  
000036     GOBACK.  
000037 REV-PARA.  
000038     MOVE STR-1(J:1) TO STR-2(I:1).  
000039     SUBTRACT 1 FROM J.  
000040     ADD 1 TO I.  
000041     EXIT.  
***** Bottom of Data *****
```

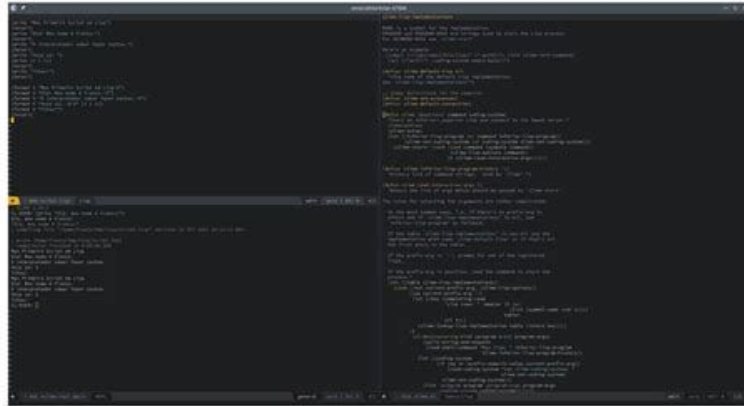
Uso: Procesamiento de transacciones, sistemas de información empresarial

Características: Diseñado para aplicaciones de negocio, menos utilizado hoy en día pero aún presente en sistemas legados.



# LISP

Tipo: Alto nivel, Interpretado, Declarativo (Funcional)



Uso: Inteligencia artificial,  
desarrollo de aplicaciones de  
software

Características: Uno de los primeros lenguajes de programación funcional, conocido por su capacidad de manipulación de listas y su poder expresivo.

# FORTRAN

Tipo: Alto nivel, Compilado, Científico

```
PROGRAM Triangle  PRINT *, "Area="
USE Circle
IMPLICIT NONE    PRINT *, "Circ
REAL :: a, b, c, Area
PRINT *, 'Welcome, please enter the&
      &lengths of
READ *, a, b, c
PRINT *, 'Triangle'
ND PROGRAM Triangle
UNCTION Area(x,y,z)
IMPLICIT NONE
REAL :: Area
REAL, INTENT( IN ) :: x, y, z
REAL :: theta, height
theta = ACOS((x**2+y**2-z**2)/(2.0*x*y))
height = x*SIN(theta); Area = 0.5*y*height
END FUNCTION
```



Uso: Cálculo numérico, computación científica

```
test_free.F90
1 MODULE test_free
2 IMPLICIT NONE
3
4 TYPE :: scale_type
5   REAL(8) :: val = 1.00
6 END TYPE scale_type
7
8 TYPE :: vector
9   INTEGER(4) :: n
10  REAL(8), POINTER, DIMENSION(:) :: v => NULL()
11 CONTAINS
12  PROCEDURE :: create => vector_create
13  PROCEDURE :: norm => vector_norm
14 END TYPE vector
15
16 TYPE, EXTENDS(vector) :: scaled_vector
17   TYPE(scale_type) :: scale
18 CONTAINS
19  PROCEDURE :: set_scale => scaled_vector_set
20  PROCEDURE :: norm => scaled_vector_norm
21 END TYPE scaled_vector
22
23 LOGICAL :: module_variable
24 CONTAINS
25
26 SUBROUTINE vector_create(self, n)
```

Características: Diseñado para cálculos numéricos, sigue siendo utilizado en áreas de alta performance computing.

# PROLOG

Tipo: Alto nivel, Interpretado, Lógico



## Query Window

```
?- likes(john, jane). ← dot necessary  
true. ← answer from prolog interpreter  
sign on  
prolog query  
prompt  
↓  
?- friends(X, Y). ← variables  
X = john,  
Y = jane ; ← type ; to get next solution  
X = jane,  
Y = john.
```

Uso: Inteligencia artificial,  
sistemas expertos, resolución de  
problemas

Características: Lenguaje declarativo basado en lógica de primer orden, utilizado para la programación lógica y la resolución de problemas simbólicos.

# PERL

Tipo: Alto nivel, Interpretado,  
Multiparadigma

```
sagar@Linuxhandbook: ~  
sagar@Linuxhandbook:~$ perl -v  
  
This is perl 5, version 34, subversion 0 (v5.34.0) built for x86_64-linux-gnu-thread-multi  
(with 50 registered patches, see perl -V for more detail)  
  
Copyright 1987-2021, Larry Wall  
  
Perl may be copied only under the terms of either the Artistic License or the  
GNU General Public License, which may be found in the Perl 5 source kit.  
  
Complete documentation for Perl, including FAQ lists, should be found on  
this system using "man perl" or "perldoc perl". If you have access to the  
Internet, point your browser at http://www.perl.org/, the Perl Home Page.  
  
sagar@Linuxhandbook:~$ _
```



Uso: Scripting,  
procesamiento de texto,  
desarrollo web (aunque  
menos popular hoy en día)

Características: Conocido como "el lenguaje de los pegamentos", Perl es potente para el procesamiento de texto y ha sido ampliamente utilizado en la administración de sistemas y la programación web.