

PyBR - Guía para Principiantes

Maxwell Anderson Ielpo do Amaral

Guía Completa para Principiantes

Aprende a programar en Python usando español

Janeiro/2026

PyBR - Python Internacional
<https://github.com/maxwellamaral/pybr>

Índice

Aprende a Programar con PyBR - Guía Completa para Principiantes	2
¡Bienvenido al Mundo de la Programación! □	2
Índice	2
□ Usando la Terminal - Guía para Principiantes	3
□ Instalando Python	3
Cómo Ejecutar PyBR	3
Tu Primer Programa	4
Variables - La Memoria de la Computadora	4
Cálculos y Operaciones Matemáticas	5
Entrada y Salida de Datos	5
Tomando Decisiones - Estructuras Condicionales	6
Repitiendo Acciones - Bucles	6
Organizando el Código - Funciones	7
Funciones Avanzadas - Lambda, Filtrar y Mapear	7
Creando Objetos - Clases	7
Proyectos Prácticos	8
Consejos Finales	8
¡Felicidades! □	9

Aprende a Programar con PyBR - Guía Completa para Principiantes

Autor: Maxwell Anderson Ielpo do Amaral Traducción: AI Assistant

Publicado en Enero de 2026

¡Bienvenido al Mundo de la Programación! □

Esta guía fue creada especialmente para ti que nunca has programado antes y quieres aprender de forma fácil y en español. Con **PyBR**, aprenderás a programar usando palabras en español en lugar del inglés tradicional de Python.

Índice

1. Usando la Terminal - Guía para Principiantes
2. Instalando Python
3. Cómo Ejecutar PyBR
4. ¿Qué es la Programación?
5. Tu Primer Programa
6. Variables - La Memoria de la Computadora
7. Cálculos y Operaciones Matemáticas
8. Entrada y Salida de Datos
9. Tomando Decisiones - Estructuras Condicionales
10. Repitiendo Acciones - Bucles
11. Organizando el Código - Funciones

- 12. Funciones Avanzadas - Lambda, Filtrar y Mapear
 - 13. Creando Objetos - Clases
 - 14. Proyectos Prácticos
-

▣ Usando la Terminal - Guía para Principiantes

Si nunca has usado la **Terminal** (también llamada **Línea de Comandos**), ¡no te preocupes! Es más simple de lo que parece.

¿Qué es la Terminal?

La Terminal es una interfaz de texto donde escribes comandos para que la computadora los ejecute.

Comandos Básicos (Windows/Mac/Linux)

Acción	Windows	Mac/Linux
¿Dónde estoy?	cd	pwd
Listar archivos	dir	ls
Entrar en carpeta	cd carpeta	cd carpeta
Volver atrás	cd ..	cd ..
Limpiar pantalla	cls	clear

▣ Instalando Python

Antes de comenzar, necesitas tener **Python** instalado.

1. Abre la terminal y escribe: `python --version`
 2. Si aparece Python 3.x.x, ¡estás listo!
 3. Si no, descárgalo en python.org.
 - **Windows:** ¡Marca la casilla “Add Python to PATH” al instalar!
-

Cómo Ejecutar PyBR

Lo que necesitas

▣ **Python 3.6+** ▣ **Archivos de PyBR** (`pybr.py`)

Formas de Ejecutar

Opción 1: Modo Interactivo (REPL) Perfecto para pruebas rápidas. En la terminal:

`python pybr.py --lang es`

Verás:

PyBR - Python Internacional (Multilenguaje)
Escribe 'salir()' para terminar.
>>>

Opción 2: Ejecutar Archivos Crea un archivo `mi_programa.pybr` y ejecútalo:

```
python pybr.py mi_programa.pybr --lang es
```

Tu Primer Programa

Vamos a empezar con el clásico “¡Hola, Mundo!”:

```
imprimir("¡Hola, Mundo!")
```

Experimenta tú mismo:

```
imprimir("Mi nombre es Juan")  
imprimir("¡Estoy aprendiendo a programar con PyBR!")
```

Variables - La Memoria de la Computadora

Las **variables** son como cajas donde guardas información.

Cómo crear variables:

```
# Guardando un nombre  
nombre = "Maria"
```

```
# Guardando una edad  
edad = 25
```

```
# Usando las variables  
imprimir(nombre)  
imprimir(edad)
```

Tipos de Datos:

```
# TEXTO (string)  
ciudad = "Madrid"
```

```
# NÚMEROS ENTEROS (int)  
cantidad = 10
```

```
# NÚMEROS DECIMALES (float)  
precio = 19.99
```

```
# VERDADERO o FALSO (boolean)
```

```
es_lunes = Verdadero
esta_lloviendo = Falso
```

Cálculos y Operaciones Matemáticas

```
# SUMA (+)
suma = 10 + 5
imprimir(suma) # Muestra: 15

# RESTA (-)
diferencia = 20 - 8
imprimir(diferencia) # Muestra: 12

# MULTIPLICACIÓN (*)
producto = 6 * 7
imprimir(producto) # Muestra: 42

# DIVISIÓN (/)
resultado = 15 / 3
imprimir(resultado) # Muestra: 5.0

# POTENCIA (**)
potencia = 2 ** 3
imprimir(potencia) # Muestra: 8
```

Entrada y Salida de Datos

Salida (Mostrar información):

```
nombre = "Carlos"
imprimir(f"Mi nombre es {nombre}")
```

Entrada (Recibir información):

```
nombre = entrada("¿Cuál es tu nombre? ")
imprimir(f"¡Hola, {nombre}!")

# Para números, necesitamos convertir:
edad = entero(entrada("¿Cuál es tu edad? "))
imprimir(f"Tienes {edad} años")
```

Ejemplo: Calculadora de IMC

```
imprimir("=== Calculadora de IMC ===")
peso = flotante(entrada("Peso (kg): "))
altura = flotante(entrada("Altura (m): "))
```

```
imc = peso / (altura ** 2)

imprimir(f"Tu IMC es: {imc:.2f}")
```

Tomando Decisiones - Estructuras Condicionales

El programa toma decisiones con si, si_no (elif) y sino.

```
edad = 18

si edad >= 18:
    imprimir("Eres mayor de edad")
sino:
    imprimir("Eres menor de edad")
```

Ejemplo con SI-SI_NO-SINO (if-elif-else):

```
nota = 75

si nota >= 90:
    imprimir("¡Excelente!")
si_no nota >= 70:
    imprimir("¡Bien!")
sino:
    imprimir("Necesitas mejorar")
```

Operadores Lógicos:

- y (and)
- o (or)
- no (not)

```
si tienes_entrada y eres_mayor:
    imprimir("Puedes entrar")
```

Repitiendo Acciones - Bucles

Bucle PARA (for):

```
# Contando de 0 a 4
para i en rango(5):
    imprimir(i)
```

Bucle MIENTRAS (while):

```
contador = 0

mientras contador < 5:
```

```
imprimir(f"Contador: {contador}")
contador = contador + 1
```

Organizando el Código - Funciones

Las funciones son bloques de código reutilizables.

```
definir saludar(nombre):
    imprimir(f"¡Hola, {nombre}!")
```

```
saludar("Ana")
saludar("Pedro")
```

Funciones con Retorno:

```
definir sumar(a, b):
    retornar a + b

resultado = sumar(10, 20)
imprimir(resultado) # 30
```

Funciones Avanzadas - Lambda, Filtrar y Mapear

Funciones Lambda (Anónimas):

Funciones pequeñas de una línea.

```
doble = lambda x: x * 2
imprimir(doble(5)) # 10
```

Mapear (map) - Aplicar función a todos:

```
numeros = [1, 2, 3, 4]
cuadrados = lista(mapear(lambda x: x ** 2, numeros))
imprimir(cuadrados) # [1, 4, 9, 16]
```

Filtrar (filter) - Seleccionar elementos:

```
numeros = [1, 2, 3, 4, 5, 6]
pares = lista(filtrar(lambda x: x % 2 == 0, numeros))
imprimir(pares) # [2, 4, 6]
```

Creando Objetos - Clases

Las clases son “moldes” para crear objetos.

```

class Perro:
    definir __init__(propio, nombre, raza):
        propio.nombre = nombre
        propio.raza = raza

    definir ladrar(propio):
        imprimir(f"{propio.nombre}: ¡Guau guau!")

# Creando objetos
rex = Perro("Rex", "Labrador")
rex.ladrar()

```

Proyectos Prácticos

Proyecto 1: Lista de Tareas

```

class GestorTareas:
    definir __init__(propio):
        propio.tareas = []

    definir agregar(propio, tarea):
        propio.tareas.append(tarea)
        imprimir(f"Tarea agregada: {tarea}")

    definir listar(propio):
        imprimir("--- Mis Tareas ---")
        para i, tarea en enumerar(propio.tareas):
            imprimir(f"{i + 1}. {tarea}")

gestor = GestorTareas()
gestor.agregar("Aprender PyBR")
gestor.agregar("Practicar Python")
gestor.listar()

```

Proyecto 2: Conversor de Temperatura

```

definir celsius_a_fahrenheit(c):
    retornar (c * 9/5) + 32

temp_c = flotante(entrada("Temperatura en Celsius: "))
temp_f = celsius_a_fahrenheit(temp_c)
imprimir(f"{temp_c}°C es igual a {temp_f}°F")

```

Consejos Finales

1. **Practica todos los días:** La constancia es clave.
2. **Lee los errores:** Son pistas para arreglar tu código.

3. **Comenta tu código:** Usa # para explicar qué hace tu programa.

¡Felicidades! ☐

Has completado la guía completa de PyBR en español. ¡Ahora eres un programador!