Pablo Orantes 25563

Revisión inicial de Tipos de datos y estructuras de datos

1. Tipos de datos primitivos en Julia

Julia está diseñado para el cálculo científico, por lo que su sistema de tipos numéricos es muy rico. Sus tipos primitivos principales son:

- Enteros (Integers): Soporta enteros con y sin signo de diferentes tamaños. Ejemplos: Int8, Int16, Int32, Int64, Int128. Si solo escribes x = 10, Julia asignará automáticamente el tipo Int64.
- punto flotante (Floats): Para números decimales. Soporta Float16, Float32 y Float64.
- Booleanos: El tipo Bool, que solo puede tener los valores true o false.
- Caracteres: El tipo Char, que representa un único carácter Unicode.

2. ¿Es un lenguaje fuertemente tipado?

Sí, Julia es fuertemente tipado. Esto significa que no realizará conversiones implícitas extrañas entre tipos incompatibles. Por ejemplo, si intentas sumar un número y una cadena de texto (1 + "1"), Julia te arrojará un error, a diferencia de lenguajes débilmente tipados que intentarían concatenarlo.

Además, es importante destacar que Julia es dinámicamente tipado por defecto, pero permite hacer anotaciones de tipo opcionales. Esto es clave en Julia, ya que le permite al compilador optimizar el código al máximo.

3. Principales estructuras de datos

Julia soporta de forma nativa las estructuras más utilizadas en la programación:

- Arreglos (Arrays / Vectors / Matrices): Son la columna vertebral de Julia. Pueden ser unidimensionales (Vectores), bidimensionales (Matrices) o multidimensionales. Son mutables y están altamente optimizados para operaciones matemáticas.
- Tuplas (Tuples): Colecciones ordenadas de elementos, pero a diferencia de los arreglos, son inmutables.
- Diccionarios (Dicts): Colecciones de pares clave-valor. Son ideales para búsquedas rápidas de datos.
- Conjuntos (Sets): Colecciones de elementos únicos sin un orden específico. Se usan para operaciones matemáticas de conjuntos.

5. Comparación: Julia vs. Python

Característica	Julia	Python
Rendimiento	Muy alto. Compila en tiempo real (JIT) usando LLVM. Es casi tan rápido como C sin necesidad de librerías externas.	Menor rendimiento en su núcleo. Depende de librerías escritas en C (como NumPy) para hacer cálculos matemáticos rápidos.
Indexación	Los índices de los arreglos empiezan en 1 (orientado a matemáticas).	Los índices empiezan en 0 (orientado a ciencias de la computación clásicas).
Paradigma Principal	Despacho Múltiple (Multiple Dispatch): Las funciones se comportan diferente dependiendo de los *tipos de todos sus argumentos*.	Orientación a Objetos tradicional.
Ecosistema	Fuerte en matemáticas, estadística y machine learning, pero más pequeño.	Ecosistema masivo y maduro para cualquier tipo de propósito (desarrollo web, automatización, etc.).