

Fase 2 - Tipos de datos y estructuras de datos

Julia

Julia es un lenguaje de programación de tipado dinámico, esto pues no es necesaria la declaración del tipo explícitamente, es importante mencionar que se permite el multiple dispatch lo cual optimiza bastante en tiempo de ejecución.

Tipos de datos primitivos

- **Float:** El numero indica la cantidad de bits Float16, Float32, Float64
- **Int:** El número indica la cantidad de bits Int8, Int16, Int32, Int64, Int128
- **UInt:** Son enteros sin signo, el número indica la cantidad de bits UInt8, UInt16, UInt32, UInt64, UInt128
- **Bool:** De 8 bits true, false
- **Char:** De 32 bits
- **String**

También es posible definir nuestro propio dato primitivo, de esta forma:

primitive type nombre bits **end**

Otros tipos de datos

Racionales: Los racionales, $\text{Rational}\{T\}$, se caracterizan por el operados `//` .

Complejos: Los complejos, $\text{Complex}\{T\}$, y la constante que se utiliza para representarlo es **im** .

(Martinez, 2022)

(Nissly, 2023)

Estructura de Datos

- **Tuple:** Se trata de un contenedor inmutable, esto quiere decir que no se puede modificar después de ser instanciado y se crea con paréntesis separando los valores por comas, puede contener diferentes tipos de datos.
- **Name tuple:** Es bastante parecido aun tuple, pero se diferencia porque se nombra cada valor y se puede acceder a ellos por su nombre, no por el índice como en las tuplas.

- **Range:** Se trata de una secuencia numérica la cual por defecto la recorre de uno en uno pero es posible modificarlo, estos valores se generan al ejecutarlo.
- **Array:** Es una colección ordenada la cual puede contener varios objetos, por defecto los elementos son del mismo tipo, pero se puede cambiar para que sean distintos.
- **Pair:** Contiene únicamente dos valores el first y second, estos se relacionan entre si.
- **Dictionary:** Es una colección donde se relacionan keys y valores, se puede pensar como una colección de pairs, los diccionarios son mutables.
- **Sets:** Es una colección de elementos únicos, es decir no existen valores repetidos, también son llamados conjuntos y se pueden hacer operaciones como unión, intersección y diferencia.

(Martinez, 2022)

(Alonso et al, s.f)

Diferencias y comparación con otros lenguajes

Este lenguaje se me hace bastante parecido a Python pues es fácil de usar y entender, me parece bastante interesante que en Julia es posible realizar varias operaciones matemáticas complejas fácilmente, algo que en Python no es potente si no se usa librerías. Me gusta que en Julia no es necesario el punto y coma en cada línea, Julia me parece un lenguaje bastante bueno.

Referencias

Martinez, A. (2022). *Introducción al lenguaje de programación Julia*. Github.

<https://github.com/alexini-mv/introduccion-julia?tab=readme-ov-file#julia-esencial>

Nissly, J. (2023). *Understanding the Julia Type System*. GLC. <https://blog.glcs.io/julia-type-system#heading-primitive-types>

Alonso, L. Huijzer, R. Storopoli, J. (s.f.). *Native Data Structures*. Julia Data Science.

https://juliadatascience.io/data_structures