**DATATYPES IN CAIRO**

|  |  |  |
| --- | --- | --- |
| Tipo de Dato | Descripción | Rango / Ejemplo |
| felt252 o felt | Elemento de campo finito de 252 bits | let x: felt252 = 42; |
| bool | Valor booleano: true o false | let is\_valid: bool = true; |
| Tupla | Agrupación de valores de diferentes tipos | let point: (felt252, felt252) = (3, 5); |
| Array | Secuencia indexada de elementos del mismo tipo | let numbers: [felt; 3] = [1, 2, 3]; |
| Struct | Tipo de dato personalizado con campos nombrados | struct Point { x: felt252, y: felt252 } |
| Enum | Conjunto de variantes o casos nombrados | enum Color { Red: (), Green: (), Blue: () } |
| () | Tupla vacía, utilizada para representar la ausencia de un valor | let empty: () = (); |
| u8 | Entero sin signo de 8 bits | 0 a 2^8 - 1 (0 a 255) |
| u16 | Entero sin signo de 16 bits | 0 a 2^16 - 1 (0 a 65,535) |
| u32 | Entero sin signo de 32 bits | 0 a 2^32 - 1 (0 a 4,294,967,295) |
| u64 | Entero sin signo de 64 bits | 0 a 2^64 - 1 (0 a 18,446,744,073,709,551,615) |
| u128 | Entero sin signo de 128 bits | 0 a 2^128 - 1 |
| u256 | Entero sin signo de 256 bits (struct con 2 campos: low: u128, high: u128) | 0 a 2^256 - 1 |
| usize | Entero sin signo de 32 bits utilizado para indexación | 0 a 2^32 - 1 (0 a 4,294,967,295) |
| Short String | Cadena ASCII corta (hasta 31 caracteres) | let my\_first\_string = 'Hello world'; |
| ByteArray | Cadena de longitud variable | let long\_string: ByteArray = "this is a string which has more than 31 characters"; |
| T\* | Puntero a un tipo de dato T | let ptr: Point\* = &point; |
| T\*\* | Puntero a un puntero de tipo T | let ptr\_ptr: Point\*\* = &ptr; |
| (T, U) | Tipo de función que acepta un argumento de tipo T y devuelve un valor de tipo U | let add: (felt252, felt252) -> felt252 = ...; |
| T -> U | Tipo de función que acepta un argumento de tipo T y devuelve un valor de tipo U | let add: felt252 -> felt252 -> felt252 = ...; |
| Decimal | Literal numérico en base 10 | 98222 |
| Hexadecimal | Literal numérico en base 16 | 0xff |
| Octal | Literal numérico en base 8 | 0o04321 |
| Binario | Literal numérico en base 2 | 0b01 |

Tipos de datos enteros con signo (i8, i16, i32, i64, i128):

* Representan enteros con signo con diferentes rangos de valores.
* Permiten almacenar tanto valores positivos como negativos.
* El rango de valores para cada tipo de dato con signo es:
  + i8: -2^7 a 2^7 - 1 (-128 a 127)
  + i16: -2^15 a 2^15 - 1 (-32,768 a 32,767)
  + i32: -2^31 a 2^31 - 1 (-2,147,483,648 a 2,147,483,647)
  + i64: -2^63 a 2^63 - 1 (-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807)
  + i128: -2^127 a 2^127 - 1
* Elige el tipo de dato entero con signo adecuado según el rango de valores que necesites representar.
* Ten en cuenta que el bit más significativo se utiliza para representar el signo (0 para positivo, 1 para negativo).
* Los tipos de datos enteros con signo son útiles cuando necesitas trabajar con valores negativos o cuando el rango de valores puede ser tanto positivo como negativo.