

Lenguaje Ensamblador

El lenguaje ensamblador, no tiene tipos de datos en el sentido tradicional y no se puede usar la definición moderna de tipos de datos al lenguaje ensamblador. El lenguaje ensamblador, por convención y estándar, sus tipos de datos son la cantidad de bytes a usar y reservar junto a un tipo más concreto como lo son los *number*. Por ejemplo, si hacemos *memory: defb 10* estamos definiendo un byte de espacio e inicializado a 10. Es importante entender que *defb* es un tipo de dato aún más primitivo que los tipos de datos primitivos de lenguajes como *C*. Debo aclarar que aunque menciono que *defb* es un tipo de dato, estrictamente *defb* es el equivalente a una palabra reservada para un ensamblador que lo soporte como directiva. No existe un estándar para los nombres de los tipos de datos en ensamblador, pero universalmente se asume y espera que un lenguaje ensamblador tenga la capacidad, de declarar *bytes*, *words*, *reserve bytes*, *numbers* y *registers*, este último es puede ser debatido debido a que tiene características por las cuales se podría considerar un tipo de dato a pesar de no cumplir con la definición dada anteriormente. Los ensambladores, según la arquitectura, pueden o no distinguir entre un *reserve byte* y los tipos inicializados como *byte* o *word*, a pesar de esto se deben considerar un tipo de dato ya que lo que hace es reservar múltiples bytes con un valor base definido por el ensamblador, por convención no oficial se espera que los inicialice a cero (0), pero algunos ensambladores como *Roudoudou Assembler (rasm)* pueden dar la opción de cambiar le valor por defecto de inicialización.