Tipos de datos

Introducción a la programación | Carlos E. Cimino

Pongamos algo en claro: para la computadora cualquier dato es un número binario. La diferencia es, a alto nivel, cómo el programador interactúa con estos datos. Hay lenguajes de programación para los cuales no hay discriminación de datos. La palabra "Hola" puede ser mezclada y operada con el número 4 o el carácter 'z'. A estos lenguajes, se los llama débilmente tipados. Otros lenguajes ponen énfasis en la diferencia entre los datos, no es lo mismo un número entero que uno real, tampoco un carácter simple que una cadena de caracteres. Los lenguajes que siguen tal descripción se denominan fuertemente tipados. Como la mayoría de las cosas, no existe una manera mejor que otra. Personalmente prefiero los lenguajes fuertemente tipados, dado que, si bien parecen más dificultosos, me permiten tener un mayor control como programador y además facilitan la depuración (seguimiento y corrección de errores). De todas maneras, hay que tener en cuenta que no siempre es posible esta elección. Básicamente, existen dos claros tipos de datos diferentes: los numéricos, los alfanuméricos y los booleanos.

Datos numéricos

Cualquier dato que represente un número, positivo, negativo, fraccionario o el cero, es considerado un dato numérico. Los lenguajes fuertemente tipados hacen una clasificación aún dentro de esta categoría. No es lo mismo un número entero como el 12 que el número 459213. Recordá que la computadora guarda los datos de forma binaria. Cuando ves un 12 en la pantalla, en realidad la máquina lo representa como un 1100 en binario. Y el número 4592193 en binario se representa así: 1110000000111001101. Como verás, guardar el número 459213 requiere más espacio en memoria que el número 12. Si me pongo un poco más técnico, supongamos que cada celda de una memoria aloja 1 byte (o sea 8 bits, o sea, 8 dígitos binarios). Notarás que el número 12 cabe perfectamente en una celda, de hecho, sobran cuatro dígitos que se rellenan con ceros a la izquierda. En cambio, el número 459213 necesita 19 dígitos, lo cual demanda 5 celdas de memoria. Por este concepto es que los lenguajes fuertemente tipados discriminan entre enteros cortos (en inglés, short), enteros normales (en inglés, int) y enteros largos (en inglés, long), etc. Si el programador tiene en cuenta con qué datos numéricos va a trabajar y hace uso de los tipos correctos, el programa será más eficiente. ¿Qué sucede con los números no enteros? A estos se lo denominan, números de coma flotante o de punto flotante. También se guardan en binario, de una forma un poco más compleja que no merece la pena que explique. Que un número real ocupe más o menos espacio depende no solo de la parte entera sino también de la cantidad de dígitos decimales. Algo que cabe destacar es que no existen dígitos decimales infinitos. El número 1/3 que en número decimal es un 0.3333… (el separador decimal en la informática es el punto) tiene infinitos dígitos 3 en la teoría, pero en la práctica, la computadora guardará un número finito de ellos. Por eso muchos lenguajes fuertemente tipados diferencian entre números de coma flotantes simples o dobles, según la cantidad de cifras decimales que se pueden guardar. Si al número 1/3 lo guardamos como número flotante doble en vez de simple, tendremos más

precisión en cuanto al número real a costa de ocupar el doble de celdas de memoria. La elección depende del problema a resolver. Tampoco es lo mismo un 12 que un 12.0 ¿Te estoy mareando? No es la idea. Claro que el 12 y el 12.0 son el mismo número, sin embargo, la computadora los trata como tipos diferentes. Un 12 es un número entero, pero sabemos que todo número entero es un número real, por lo que, si se guarda el 12 como número de coma flotante, ocupará más memoria, pero permitirá operar con otros números de coma flotante.

Datos alfanuméricos

El otro conjunto de datos a analizar es el de los alfanuméricos. En esta categoría se encuadran los caracteres como las letras del alfabeto o los símbolos
Repito, la computadora guarda los datos de manera binaria. Cómo la máquina codifica los caracteres merece explicarlo en un capítulo aparte, pero digamos que tal vez hayas escuchado hablar de ASCII, acrónimo inglés de American Standard Code for Information Interchange (Código Estándar Estadounidense para el Intercambio de Información). Este código representa una tabla donde cada carácter tiene asignado un número. ¿Alguna vez has visto que si mantenés presionada la tecla ALT y tecléas el número 64 en el teclado numérico, se escribe un carácter "@" (arroba)? Tiene que ver con este código, que contiene 128 caracteres (los primeros 32 de control, no imprimibles) y cada uno de ellos ocupa 1 byte. En la versión extendida se ofrecen 128 caracteres extra.
Un tipo de dato un poco más complejo que merece ser mencionado es el de la cadena de caracteres (en inglés, string). Una cadena es un conjunto de caracteres que permiten formar una palabra, frase o párrafo. Desde la palabra "pez" hasta la biblia completa se considera una cadena de caracteres. Este tipo de dato se diferencia a los vistos hasta ahora en que no es considerado un tipo de dato primitivo, sino una estructura más compleja formada por tales. En realidad, una cadena es un vector o arreglo (array) unidimensional de caracteres. Hay lenguajes que las manejan como tal mientras que otros intentan emularlo como un dato primitivo para facilitar su tratamiento. Verás que los números también están dentro de la tabla ASCII y es aquí donde se responde la pregunta pendiente en la página 26: El número 8 sin comillas es un número entero, capaz de ser operado matemáticamente. El número '8' como carácter (en muchos lenguajes, se lo encierra entre comillas simples) es tratado como un símbolo. No sirve para operar matemáticamente, sino para ser concatenado con otros caracteres y así formar palabras. Inclusive podemos tener un "8" como cadena de caracteres (en muchos lenguajes, se lo encierra entre comillas dobles), es decir, una estructura de datos de un símbolo de longitud. Es importante tener en cuenta estas tres diferencias

Datos booleanos

Los booleanos son un tipo especial de dato que representan un valor de verdad según la lógica clásica (veremos un poco de ella más adelante). Deben su nombre a George Boole (1815 - 1864), matemático y lógico británico enunciador del álgebra que lleva su nombre. En un tipo de dato booleano solo caben dos valores posibles: VERDADERO (en inglés, true) o FALSO (en

inglés, false). Algunos lenguajes pueden representarlo como 0 para FALSO y 1 para VERDADERO, aunque es importante destacar que un 0 booleano no es igual a un 0 entero. Lo mismo para el 1. De hecho, si trabajamos con álgebra de números el resultado de 1 + 1 es 2, en cambio, en el álgebra de Boole, VERDADERO + VERDADERO da como resultado VERDADERO. Los datos booleanos son los más económicos en cuanto a espacio en memoria, dado que se pueden representar con tan solo un bit. Pero como cada celda de memoria guarda 8 bits y no se puede fraccionar, termina ocupando 1 byte.