

O tipo de dado funciona como uma classificação da informação. As informações, de forma geral, seguem regras que possibilitam uma classificação. Devido aos diferentes tipos de informações existente, existe, na programação, os tipos de dados. Os tipos são, basicamente, as formas diferentes de representar diferentes informações. Tipo primitivo são os tipos de dados mais simples, isto é, a informação em sua forma mais primitiva.

Os tipos primitivos em C são:

O tipo char é um tipo de informação que é capaz de armazenar um caractere alfanumérico. Caracteres geralmente são armazenados em códigos. Geralmente, o char é usado para a construção de cadeias de caracteres, ou seja, um conjunto de letras que formam uma frase ou então, uma palavra.

O tipo int representa os números inteiros. Logo, pode-se armazenar tanto números positivos como também, números negativos. Normalmente, o tipo int utiliza 2 bytes de memória.

O tipo float representa os números com ponto flutuante, os números que contenham casas decimais. Normalmente, o tipo float ocupa 4 bytes de memória. São conhecidos normalmente como números reais.

O tipo double também representa os número com ponto flutuante, porém, com o tipo double, existe mais precisão do que com o tipo float. Isso porque o tipo double faz uso de 8 bytes e assim, consegue armazenar números numa faixa muito extensa.

O tipo void representa um tipo sem tipo. Podemos pensar no tipo void como sendo a representação do vazio ou então do nada. Normalmente é usado em funções que não irão retornar valores. Logo, a função retorna void, isto é, não retorna nenhum valor. Este tipo serve para indicar que um resultado não tem um tipo definido.

Modificadores podem ser aplicados a estes tipos. Estes modificadores são palavras que alteram o tamanho do conjunto de valores que o tipo pode representar.

Tabela: Todos os Tipos de dados definidos pelo Padrão ANSI C, seus tamanhos em bytes e suas faixa de valores.

<b>Tipo</b>	<b>Tamanho em Bytes</b>	<b>Faixa Mínima</b>
<b>char</b>	1	-127 a 127
<b>unsigned char</b>	1	0 a 255
<b>signed char</b>	1	-127 a 127
<b>int</b>	4	-2.147.483.648 a 2.147.483.647
<b>unsigned int</b>	4	0 a 4.294.967.295
<b>signed int</b>	4	-2.147.483.648 a 2.147.483.647
<b>short int</b>	2	-32.768 a 32.767
<b>unsigned short int</b>	2	0 a 65.535
<b>signed short int</b>	2	-32.768 a 32.767
<b>long int</b>	4	-2.147.483.648 a 2.147.483.647
<b>signed long int</b>	4	-2.147.483.648 a 2.147.483.647
<b>unsigned long int</b>	4	0 a 4.294.967.295
<b>float</b>	4	Seis dígitos de precisão
<b>double</b>	8	Dez dígitos de precisão
<b>long double</b>	10	Dez dígitos de precisão

Os tipos primitivos em Java são:

O byte aceita até 256 valores. Eles podem estar compreendidos entre  $-128 \leq x \leq 127$ . O tipo de dado byte pode assumir valores inteiros positivos ou negativos e requer 8 bits para ser implementado. O tipo de dado byte pode ser útil para economizar espaços de memória em matrizes de grandes dimensões, em casos que sejam viáveis. Variáveis tipo byte também podem ser usadas no lugar de variáveis int de tal modo que seus limites ajudem a clarear o seu código, o fato de que uma variável byte é limitada pode servir como uma forma de documentação.

O tipo de dado short pode assumir valores inteiros positivos ou negativos e requer 16 bits para ser implementado. O valor mínimo de uma variável short é de -32.768 e o valor máximo de 32.767. Tal como acontece com byte, as mesmas diretrizes se aplicam: você pode usar um short para economizar memória em arrays grandes, em situações em que as economias seja realmente importante.

O tipo de dado int pode assumir valores inteiros positivos ou negativos e requer 32 bits para ser implementado. O valor mínimo de uma variável int é de -2.147.483.648 e o valor máximo de 2.147.483.647. Para valores inteiros, esse tipo de dado é geralmente a escolha padrão, a menos que haja uma razão para escolher outro tipo.

O tipo de dado long pode assumir valores inteiros positivos ou negativos e requer 64 bits para ser implementado. O valor mínimo de uma variável long é de -9.223.372.036.854.775.808 e o valor máximo de 9.223.372.036.854.775.807. Use este tipo de dado é utilizado quando você precisa de um intervalo de valores maior do que os previstos por int.

O tipo de dado float é de precisão simples de 32 bits IEEE 754 ponto flutuante. Tal como acontece com as recomendações para byte e short, use float (em vez de double), se você precisa economizar memória em grandes conjuntos de números decimais. Este tipo de dado nunca deve ser usado para valores que precisam ser muito precisos, como moeda.

O tipo de dado double é de dupla precisão de 64 bits IEEE 754 ponto flutuante. Para valores decimais, este tipo de dado é geralmente a opção padrão.

O tipo de dado boolean tem apenas dois valores possíveis: true e false(verdadeiro/falso). Este tipo de dado representa um bit de informação, mas o seu “tamanho” não é algo que é definido com precisão.

O tipo de dado char é um caractere de 16 bits único Unicode. Ela tem um valor mínimo de '\u0000' (ou 0), e um valor máximo de '\uffff' (ou 65535).

Os tipos primitivos em Pascal são:

O tipo integer que armazenam números inteiros (naturais) cujos valores estejam entre  $-(\text{MAXINT}+1)$  e  $\text{MAXINT}$ .  $\text{MAXINT}$  é uma constante predefinida que pode variar de sistema para sistema ou de compilador para compilador. No ambiente Turbo Pascal, por exemplo,  $\text{MAXINT}$  vale 32768. Isto significa que os números neste compilador variam entre -32767 a 32768 e ocupam dois bytes (16 bits, que geram 65536 combinações possíveis de armazenamento);

O tipo byte é um subconjunto do tipo Integer que armazena valores entre 0 e 255. Este tipo ocupa 1 byte (8 bits, que geram 256 combinações);

O tipo real armazena números reais positivos e negativos (incluindo frações). No turbo Pascal este tipo ocupa 6 bytes de memória (48 bits), mas o número não é armazenado como nos outros tipos. Neste caso, o número é armazenado num formato conhecido por formato científico, onde ele é dividido em duas partes: a mantissa e o expoente. A mantissa contém números significativos do numero real automaticamente normalizadas pelo computador para um valor fracionário na faixa entre 1 e 10. O expoente é um valor inteiro positivo ou negativo que indica a posição da vírgula no número.

O tipo char vem da abreviação da palavra inglesa “character”, que significa caractere. Serve para armazenar um único caractere (uma letra, dígito ou símbolo). Ocupa 1 byte de memória, o que significa que pode armazenar 256 combinações de bits. Esse é exatamente o tamanho da tabela Americana de Códigos Padrão para a Troca de Informações (ASCII), que contém uma lista de 256 caracteres que variam entre caracteres de controle, letras, dígitos e símbolos. Cada um deles possui um código (um número) fixo. Através do número é possível se descobrir o caractere correspondente na tabela e vice-versa.

O tipo string armazena cadeias de caracteres. No fundo é o mesmo que um vetor de Chars, cujo tamanho máximo é o de 255 caracteres. Possui um byte (na posição 0) que indica quantas letras a string possui armazenada dentro dela;

O tipo boolean armazena valores lógicos que variam entre “true” (verdadeiro) ou “false” (falso). São muito utilizadas em variáveis do tipo flag e em testes condicionais.

A documentação do Python não trata os tipos de dados elementares com a nomenclatura de Tipo Primitivo, mas sim chama-os de tipos built-ins, isto é, classes que estão definidas por padrão dentro da Máquina Virtual do Python e que representam determinado tipo de informação.

Em Python, o tipo Booleano é um tipo primitivo que possui um de dois valores: verdadeiro (true) ou falso(false). Esse é um tipo fundamental. Muito comum em outras linguagens. Existe um tipo especial em python, o none. Significa não existente, não conhecido ou vazio. Python suporta tanto integers quanto floating point. Não existe declaração de tipo para distingui-los. O Python os diferencia pela presença ou ausência do ponto decimal.

Uma lista é um conjunto ordenado de valores, onde cada valor é identificado por um índice. Os valores que compõem uma lista são chamados elementos. Listas são similares a strings, que são conjuntos ordenados de caracteres, com a diferença que os elementos de uma lista podem possuir qualquer tipo. Tuplas são estruturas de dados ordenadas e que não permitem alteração após sua criação. Ou seja, é uma lista imutável. Dicionários são estruturas de dados que implementam mapeamentos. Um mapeamento é uma coleção de associações entre pares de valores.

Universidade Federal de Roraima - Curso de Ciência da Computação  
Jaqueline Miranda Neves  
Disc.: Linguagens de Programação  
Referências Bibliográficas:

<http://www.diveintopython3.net/native-datatypes.html>

<https://rubyparanovatos.wordpress.com/pascal/tutoriais/tipos-primitivos-de-dados/>

<https://dealunoparaaluno.wordpress.com/2013/04/03/os-8-tipo-de-dados-primitivos-em-java/>

Tabela retirada do site:

<http://equipe.nce.ufrj.br/adriano/c/apostila/tipos.htm>