

Variáveis e constantes são componentes fundamentais que um determinado programa irá manipular afim de realizar determinada tarefa. Uma variável representa um espaço alocado na memória do computador que irá conter um determinado dado. Essas variáveis recebem nomes, para que possam ser referenciadas ou modificadas ao longo de um algoritmo. Algumas linguagens de programação exigem que ao declararmos uma variável, seu conteúdo seja especificado e as vezes seja atribuído até mesmo um valor inicial. Esse tipo de especificação varia conforme a linguagem conforme veremos a seguir.

### Tipos de dados primitivos em C

Existem cinco tipos de dados pré-definidos em C, e são eles char, int, float, double e bool.

Int : É o tipo de dado mais comum, e sua declaração se dá através do uso da palavra reservada int. O intervalo numérico que pode ser atribuído à variável int vai de -32.768 até 32.767.

Ao realizar uma manipulação de dados com a variável, e seu conteúdo int estiver fora desse intervalo, é necessário uma declaração diferente, de modo que essa variável possa conter esse valor.

Float : Utilizada para armazenar números fracionários, e assim como a declaração de números inteiros, sua declaração se dá através do uso da palavra reservada float, contendo valores que estão entre a faixa de  $1,2e-38$  até  $1,2e+38$ .

Double : Sua declaração se dá através do uso reservada da palavra double. É utilizada na armazenagem de números reais, porém possui uma precisão maior, também chamada de precisão dupla, e pode ser usado quando o número real tiver uma precisão muito grande.

Bool : A variável booleana que representa um operador lógico só pode armazenar dois valores: true ou false, porém seu "tamanho" não é algo definido com precisão.

Char : É o tipo de dado que é utilizado para armazenar um caracter, que são geralmente armazenados conforme o padrão da tabela ASCII. O conjunto de caracteres que esse tipo de dado suporta é de 65.

## Tipos de dados em Java

Os dados primitivos ou informações mais usuais e básicas em Java podem ser classificados em três categorias: lógicos, numéricos e caracteres.

**Lógicos** : Este tipo de dado armazena um único bit de informação e pode ser representado pela palavra true ou false, e representa um operador lógico, ou seja, verdadeiro/falso, certo/errado.

### Numéricos

**Byte** : Capaz de armazenar 8 bits de informação, e valores inteiro no intervalo de  $-2^8$  até  $2^8$ , ou seja ( -128 até 127 ) é mais utilizado em casos em que seja necessário economizar espaços na memória.

**Short** : Armazena números inteiros de 16 bits que estão entre -32.768 e 32.767.

**Int** : Corresponde ao tipo de dado mais usual nos processadores atuais, e é capaz de armazenar 32 bits de informação, valores numéricos que estão entre  $-2^{32}$  até  $2^{32}$ .

**Long** : Mais utilizado em casos de valores que estão fora do intervalo usado pelo int, é capaz de armazenar 64 bits de informação

**Float** : Representa números reais de precisão simples e são capazes de armazenar 32 bits de informação.

**Double** : Utilizado para armazenar valores reais, porém com maior precisão ( precisão dupla ) que o float, sendo assim consegue armazenar 64 bits de informação.

**Char** : O tipo char guarda um caractere, e sua declaração deve ser através do uso de aspas simples. Consegue armazenar até 16 bits de dados.

## Tipos de dados em Ruby

Pelo fato de Ruby ser orientado a objeto, significa que não existem tipos de dados primitivos. Até mesmo uma variável que armazene um valor inteiro é tratado como objeto. Sendo assim, todos os tipos são objetos e com isso podemos citar os principais tipos.

**Integer** : A classe integer representa valores numéricos do tipo inteiro e representa uma classe abstrata, ou seja, é a única classe que não é possível instanciar objetos com ela. Essa classe serve apenas como modelos para as classes que herdam ou estendem dela.

**Fixnum** : É uma das subclasses de integer que representa números inteiros com limite finito. A diferença entre fixnum e integer é que a classe integer nunca é usada para atribuição, apenas para herança.

**Bignum** : Outra subclasse de integer, porém representa numeros inteiros em um intervalo quase que infinito, isso só irá depender da arquitetura do computador.

**Float** : Deriva-se da classe numeric, e é usada para representar valores fracionários independente da quantidade de casas decimais desse número.

**String** : Armazenam um conjunto de caracteres e são usada para representar textos. Porém podem ser declaradas com aspas simples ou aspas duplas.

O conteúdo de uma variável declarada com aspas simples é interpretado literalmente, já com aspas duplas serve para conter textos, expressões que podem inferir em outras expressões.

**Range** : Representa um intervalo de objetos. É declarado usando ( .. ) ou ( ... ). Por exemplo, ao criar uma lista que contenha um range 1..10, esta lista irá conter todos os valores de 1 até 10. Porém ao utilizar o range 1...10 irá excluir o limite dela, ou seja o 10.

**Array** : Também conhecida como vetor, pode armazenar objetos que podem até mesmo ser de tipos diferentes. Cada elemento está associado à um índice, que inicia em 0 e pode-se recuperar um determinado elemento.

Sua declaração é delimitada através do uso de colchetes e cada elemento deve ser separado por vírgula. Além disso, a própria linguagem Ruby possui diversos métodos para manipulação de arrays, assim como Python.

**Hash** : A ideia de hash é a mesma que a de um dicionário, em há o armazenamento de elementos ou informações, porém cada elemento contém uma informação e está associado à uma chave.

Sua declaração se dá através do uso de chaves ( {} ) e assim como no Python, o Ruby possui diversos métodos para trabalhar com manipulação de hash.