

Os tipos de dados predefinidos na linguagem **Pascal**, e implementados no compilador, são as mesmas regras para a criação das variáveis simples:

- **Boolean**

- Define dois valores lógicos: FALSE e TRUE.
- Um dado do tipo booleano ocupa um byte de espaço na memória.

- **Char**

- Define os elementos do conjunto de caracteres que compõem o alfabeto ASCII, adicionados dos caracteres representados pelos códigos de 128 a 255.
- Um dado do tipo char ocupa um byte de espaço na memória.

- **Integer**

- Define os valores inteiros compreendidos no intervalo de -2.147.483.647 até 2.147.483.647.
- Um dado do tipo integer ocupa quatro bytes de espaço na memória.

- **Real**

- Define os valores reais definidos no intervalo de $3.4 \cdot (10^{-38})$ até $3.4 \cdot (10^{+38})$.
- Um dado do tipo real ocupa quatro bytes de espaço na memória.

- **String**

- Define uma cadeia de caracteres. Se nenhuma restrição de tamanho for especificada, um dado do tipo string é capaz de armazenar uma sequência contendo até 255 caracteres, onde cada caracter ocupa um byte de espaço na memória.

Uma cadeia de caracteres pode ter seu tamanho definido (contendo menos de 255 caracteres), onde o tamanho especifica o número máximo de caracteres contidos na cadeia. Essa especificação deve ser indicada entre colchetes, logo após a palavra reservada string,

Exemplo

string [6] define uma cadeia capaz de armazenar até 6 caracteres.

Uma cadeia de caracteres definida com n caracteres ocupa n bytes de espaço na memória

Tipos de dados complexos, onde as variáveis podem ser complexas também:

Vetores e Matrizes

Um vetor é uma estrutura de dados que contém um número fixo de elementos que possuem um mesmo tipo de dados, tipo esse que pode ser qualquer um dos tipos predefinidos na linguagem Pascal (integer, char, boolean ou string), um tipo *vetor*, um tipo *registro* ou ainda um tipo definido pelo usuário.

O número de elementos de um vetor é determinado pelo intervalo de indexação de elementos do vetor, que é especificado por duas constantes ordinais separadas por dois pontos, entre colchetes.

A sintaxe para definição de vetores segue o seguinte formato:

```
array [ limiteInferior .. limiteSuperior ] of tipo ;
```

Onde:

- **array** e **of** são palavras reservadas da linguagem Pascal, usadas na declaração de vetores
- limiteInferior e limiteSuperior são constantes ordinais;
- tipo define o tipo de dados de cada elemento do vetor

Exemplo. A declaração abaixo define um vetor do tipo inteiro, identificado por *Dias*:

```
Var  
  dias : array [ 1 .. 24 ] of integer;
```

Nesse vetor, os elementos estão armazenados nas posições de 1 a 24.

A referência ao elemento que está armazenado na posição *x* de um vetor é dado da seguinte forma:

```
nomeVariavel [ x ]
```

Os vetores podem ter, ainda, [mais de uma dimensão](#).

Vetores podem ter mais de uma dimensão. Nesse caso, cada nova dimensão é declarada de acordo com as regras do item anterior, e as *n* dimensões do vetor são separadas por vírgulas.

A sintaxe para definição vetores *n*-dimensionais segue o seguinte formato:

```
array [ limite1 .. limite2 , limite3 .. limite4 , ... , limiten-1 ..  
limiten ] of tipo;
```

Exemplo. A declaração abaixo define um vetor de duas dimensões, do tipo inteiro:

```
var  
  matriz : array [1 .. 10, 1.. 20] of integer ;
```