

# PERSPECTIVAS

Programando o Futuro

## Organizando as informações sobre os robôs sociais

Informações adicionais



## Sobre nome de variáveis

Ao escolher um nome para uma variável é preciso seguir algumas regras da linguagem Python:

1. Um nome de variável deve começar com uma letra ou um *underscore* (sublinhado).
2. Um nome de variável não pode começar com um número.
3. Um nome de variável pode conter apenas caracteres alfanuméricos e *underscore* (A-z, 0-9 e `_`) (Não é permitido utilizar espaço ou hífen).
4. Os nomes das variáveis diferenciam maiúsculas de minúsculas, assim “idade” é diferente de “Idade” e de “IDADE”. Se as três palavras forem usadas, serão três variáveis diferentes.

## Sobre a definição da variável e seu tipo

Em Python, não é necessário determinar manualmente o tipo da variável, já que o **interpretador** da linguagem é capaz de determinar o tipo da variável avaliando o seu conteúdo.

*O **interpretador** é o software que converte o código fonte (o programa escrito em uma linguagem de programação) em um código alvo (um programa em linguagem de máquina), linha por linha. Depois da conversão, o computador executará o que foi convertido. Nenhum arquivo adicional seria criado nessa conversão.*



## Principais tipos em Python

- Estes não são todos os tipos disponíveis, apenas os principais.

Tipos		Atribuição
Tipo numérico	inteiro ( <i>int</i> ) ponto flutuante ( <i>float</i> )	inteiro = 10 decimal = 3.14
Tipo sequencial	lista ( <i>list</i> ) tupla ( <i>tuple</i> )	lista = [0, 1, 2, 3] tupla = (0, 1, 2, 3)
Tipo texto	string ( <i>str</i> )	frase = "Olá Mundo!"
Tipo booleano	booleano ( <i>bool</i> )	verdadeiro = True falso = False

Nos tipos numéricos nós podemos armazenar números inteiros (*int*, de *integers*) e decimais (*floats*, de *floating-point*).

Os números decimais em Python são representados utilizando **ponto** no lugar da **vírgula**..

Variáveis do **tipo sequencial** armazenam sequências de valores arbitrários e são uma forma de armazenar múltiplos valores em uma única variável. O tipo **lista** é um dos principais tipos sequenciais utilizados em Python. Listas são declaradas utilizando colchetes, os valores dentro os colchetes são separados por vírgulas.



Outro tipo sequencial são as **tuplas**. Diferentemente das listas, tuplas são imutáveis, dessa forma não podemos alterar, adicionar ou remover itens depois que a tupla foi definida. Mas, ainda assim, é possível sobrescrevê-las. Tuplas são definidas utilizando parênteses e seu conteúdo também é separado por vírgulas.

Python possui apenas um tipo de **texto**, a *String*, que armazena uma sequência de caracteres. Strings também são imutáveis, uma vez criadas, não podem ser modificadas, apenas substituídas. Não existe um tipo específico para caracteres. Um caractere é representado por uma *string* de tamanho 1.

Por fim, temos o tipo **booleano**, este tipo é de extrema importância para a computação. A utilidade desse tipo se torna mais clara no uso de estruturas de controle de fluxo de controle e de operadores lógicos. Um tipo booleano possui apenas dois valores lógicos possíveis, *True* para verdadeiro e *False* para falso.

*Os valores lógicos True e False devem ser escritos com a primeira letra em maiúsculo.*

## Como saber o tipo da variável

Depois que as variáveis são criadas e um valor é atribuído a cada uma delas, pode ser difícil determinar o tipo de cada uma. Uma das melhores formas de evitar esse tipo de confusão é definir nomes de variáveis sempre muito claros, representando o que está armazenado nela. Por exemplo, ao ler o nome "altura\_iao", sabemos que o valor armazenado provavelmente é um tipo numérico, já que deve representar a altura do robô NAO.



Porém, só isso não é o suficiente, é importante saber como recuperar o tipo de uma variável, especialmente em programas mais complexos. Para isso, podemos utilizar o comando “type”. Este comando recebe o nome de uma variável como parâmetro e retorna (devolve) o seu tipo. Por exemplo, no comando abaixo estamos solicitando que o programa imprima na tela o tipo da variável “altura\_iao”. A resposta é o tipo “int” (de inteiro).

```
print(type(altura_iao))  
<class 'int'>
```

## Sobrescrevendo o valor da variável

Uma variável pode receber uma atribuição de valor mais de uma vez mas, quando isso ocorre, o valor anteriormente armazenado é substituído pelo novo valor declarado.

