Variáveis

Aqui vemos definição, declaração, manipulação, os tipos de variáveis existentes e funções e métodos de teste de tipo.

O que são

Variáveis são espaços na memória onde, por exemplo, podemos armazenar:

* Números;
* Caracteres;
* Textos;
* Relações com outras variáveis.

Variáveis podem ser alteradas em “**tempo de execução**”, o que significa que você pode utilizar mais de uma vez a mesma variável, seja atribuindo diferente valores ou o que for. Assim sendo, dizemos que são variáveis “mutáveis”.

Como declarar

Para declarar uma variável em Python, seguimos a seguinte sintaxe:

Nome\_da\_variável = valor

Veja que:

* Utilizamos o sinal de **atribuição** "=". Este sinal é o que atribui a variável um valor. Para uma variável, lê-se "variável recebe 10" ou então "variável tem 10 atribuído a ela", por exemplo;
* Não importa a quantidade de espaços que colocamos entre o nome da variável e o valor;
* Existem nomes proibidos para variáveis, geralmente, quando se trata de nomes que já existem na sintaxe da linguagem;
* Para armazenas certos tipos de variáveis existem formas específicas de declaração.

Texto

Descrição gerada automaticamente

Entrando valores nas variáveis

De forma rápida, dizemos na computação que o seu teclado (usuário utilizando) é uma entrada de dados, já o monitor (terminal do programa) é a saída desses dados.

Desta forma, para que se entre com dados em uma variável, é necessário utilizar a função input() e para que saia dados utilizamos a função print().

### Sobre o print()

* Para a saída direta de uma variável, apenas a coloque dentro dos parênteses;
* Para a saída de apenas textos, será necessário colocar o texto dentro de aspas ou aspas duplas;
* Aspas simples permitem a **concatenação**, que nada mais é que a junção de textos através do operador **+**. Só é possível concatenar textos.

### Sobre o input()

* Ou você utiliza o input para extrair uma entrada e armazena esta entrada numa variável;

Texto

Descrição gerada automaticamente

* Ou você colocará **dentro do input() tudo aquilo que você precisa extrair de uma entrada**;

Texto

Descrição gerada automaticamente

Tipos de Variáveis

No que diz respeito ao conteúdo das variáveis, podemos classificá-las de acordo com seu tipo.

Em Python, não precisamos especificar qual o tipo de uma variável quando vamos operar com ela, basta colocar o valor que automaticamente é definido o tipo correspondente para ela.

Cada tipo de variável ocupa mais ou menos espaço, entretanto, **a memória é alocada dinamicamente**, o que significa que **o valor que uma variável ocupa depende da quantidade de elementos que está dentro dela**.

Ela opera e ocupa baseado no quanto de memória a máquina tem, ou seja, não possui um limite definido. Motivo este sendo um dentre vários que fazem Python ser uma das principais escolhas para ramos matemáticos e científicos.

### Inteiro (int)

Tipos inteiros incluem os negativos e os positivos, só não incluem números decimais.

Exemplo: -2, -1, 0, 1, 2

**Para forçar** uma variável a se **transformar em inteiro**, usamos o **int()** colocando a variável a ser transformada dentro dele.

### Decimal (float)

Tipos decimais incluem os inteiros, entretanto, com o adicional de também possuírem números decimais.

Exemplo: (-2), (-1.5), (0), (1.3), (2.2)

**Para forçar** uma variável a se **transformar em decimal**, usamos o **float()** colocando a variável a ser transformada dentro dele.

### Texto (string)

Tipos textos incluem letras únicas até textos inteiros;

**Para forçar** uma variável a se **transformar em texto**, usamos o **str()** colocando a variável a ser transformada dentro dele.

### Lógico (boolean)

Tipos lógicos só incluem um de dois valores: Verdadeiro (1) ou Falso (0).

Funções e Métodos de teste de tipo

Podemos através de alguns recursos fazer testes de tipos para descobrir qual o tipo da variável que estamos analisando.

### Função type()

A função irá **retornar o tipo da variável indicada**.

Texto

Descrição gerada automaticamente

No caso acima vai sempre retornar uma string pois está usando input() não convertido.

### Método .isinstance()

O método .isinstance() irá retornar **se a variável indicada é desta natureza**.

Texto

Descrição gerada automaticamente

Existem diversos outros métodos interessantes “is”. Verifique-os quando puder!

Operadores

O que são

Os operadores são símbolos especiais que realizam operações em operandos para produzir um resultado. Eles são utilizados para realizar cálculos, comparações e outras operações em diferentes tipos de dados.

Existem diversos tipos de operadores em Python, incluindo:

* **Operadores Aritméticos**: São utilizados para realizar operações matemáticas básicas, como adição, subtração, multiplicação, divisão etc.
* **Operadores de Atribuição**: São usados para atribuir valores a variáveis. O operador de atribuição básico do Python é o **sinal de igual**(**=**).
* **Operadores de Comparação**: Também chamados de Relacionais, são usados para comparar dois valores e retornar um **resultado booleano** que indica se a comparação é verdadeira ou falsa.
* **Operadores Lógicos**: São utilizados para realizar operações lógicas **em valores booleanos**. Os operadores lógicos mais comuns são **AND**, **OR** e **NOT**.
* **Operadores de Identidade**: São usados para verificar se dois objetos têm a mesma identidade (ou seja, se estão armazenados no mesmo local na memória).
* **Operadores de Associação**: Também chamado de Membership, São utilizados para verificar se um valor está presenta em uma sequência (como uma **lista**, **tupla**, **conjunto** ou **dicionário**).
* Operadores de Bitwise: São usados para realizar **operações bit a bit em valores inteiros**.

Operadores Aritméticos

Os operadores aritméticos são utilizados para realizar operações matemáticas básicas em números. Eles permitem realizar adições, subtrações, multiplicações, divisões, entre outras operações.

### Adição

O operador de adição em Python é o +.

Texto

Descrição gerada automaticamente

Uma curiosidade é que, além da função aritmética deste operador, podemos utilizá-lo para concatenar strings!

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

### Subtração

O operador de subtração em Python é o -.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

### Multiplicação

O operador de multiplicação em Python é o \*.

Texto

Descrição gerada automaticamente

Uma curiosidade é que além da função aritmética deste operador, podemos utilizá-lo para repetir strings!

Texto

Descrição gerada automaticamente

### Divisão

O operador de divisão em Python é o /.

Texto

Descrição gerada automaticamente

### Divisão Inteira

O operador de divisão inteira em Python é o //.

Utilizado para dividir um valor pelo outro e retomar apenas a parte inteira do resultado, descartando a parte decimal.

Texto

Descrição gerada automaticamente

### Exponenciação

O operador de exponenciação em Python é o \*\*.

Texto

Descrição gerada automaticamente

É possível também fazer qualquer tipo de raiz, basta elevar a uma potência fracionária cujo denominador seja o tipo de raiz que quer.

Texto

Descrição gerada automaticamente

### Módulo

O operador de módulo em Python é o %.

Utilizado para retornar o resto da divisão inteira entre dois valores.

Texto

Descrição gerada automaticamente

Módulos