**Métodos Build-in Python**

[**abs()**](https://docs.python.org/3/library/functions.html#abs)

Retorna o valor absoluto de um número.

[**all()**](https://docs.python.org/3/library/functions.html#all)

Retorna True se todos os elementos do iterável forem verdadeiros (ou se o iterável estiver vazio).

[**any()**](https://docs.python.org/3/library/functions.html#any)

Retorna True se algum elemento do iterável for verdadeiro. Se o iterável estiver vazio, retorne False.

[**ascii()**](https://docs.python.org/3/library/functions.html#ascii)

Manipula strings. Retorna caracteres ascii em pontuações diferentes.

[**bin()**](https://docs.python.org/3/library/functions.html#bin)

Converte inteiro em string binária prefixada por 0b.

[**bool()**](https://docs.python.org/3/library/functions.html#bool)

Retorna um valor booleano, ou seja, um True ou False.

[**breakpoint()**](https://docs.python.org/3/library/functions.html#breakpoint)

Pára uma iteração em um loop for no valor desejado.

[**bytearray()**](https://docs.python.org/3/library/functions.html#func-bytearray)

Retorna um novo array de bytes.

[**bytes()**](https://docs.python.org/3/library/functions.html#func-bytes)

Retorna um novo objeto “bytes” que é uma sequência imutável de inteiros no intervalo .

[**callable()**](https://docs.python.org/3/library/functions.html#callable)

Retorna [True](https://docs.python.org/3/library/constants.html#True) se o argumento do objeto é chamável.

[**chr()**](https://docs.python.org/3/library/functions.html#chr)

Retorna a string que representa um caractere cujo ponto de código Unicode é o inteiro i .

[**classmethod()**](https://docs.python.org/3/library/functions.html#classmethod)

Transforme um método em um método de classe.

[**compile()**](https://docs.python.org/3/library/functions.html#compile)

Compile a fonte em um código ou objeto AST. Objetos de código podem ser executados por [exec()](https://docs.python.org/3/library/functions.html" \l "exec" \o "executivo)ou [eval()](https://docs.python.org/3/library/functions.html" \l "eval" \o "avaliação). source pode ser uma string normal, uma string de bytes ou um objeto AST.

[**complex()**](https://docs.python.org/3/library/functions.html#complex)

Retorna um número complexo com o valor real + imag \*1j ou converte uma string ou número em um número complexo.

[**delattr()**](https://docs.python.org/3/library/functions.html#delattr)

Este é um parente de [setattr()](https://docs.python.org/3/library/functions.html" \l "setattr" \o "setattr). Os argumentos são um objeto e uma string. A string deve ser o nome de um dos atributos do objeto. A função exclui o atributo nomeado, desde que o objeto permita.

[**dict()**](https://docs.python.org/3/library/functions.html#func-dict)

Crie um novo dicionário. O [dict](https://docs.python.org/3/library/stdtypes.html" \l "dict" \o "ditar) objeto é a classe do dicionário [dir()](https://docs.python.org/3/library/functions.html#dir)

**dir([])**

Sem argumentos, retorne a lista de nomes no escopo local atual. Com um argumento, tente retornar uma lista de atributos válidos para esse objeto

[**divmod()**](https://docs.python.org/3/library/functions.html#divmod)

Pegue dois números (não complexos) como argumentos e retorne um par de números consistindo em seu quociente e resto ao usar a divisão inteira. Com tipos de operandos mistos, aplicam-se as regras para operadores aritméticos binários.

[**enumerate()**](https://docs.python.org/3/library/functions.html#enumerate)

Retorne um objeto enumerar.

[**eval**()](https://docs.python.org/3/library/functions.html#eval)

Os argumentos são uma string e globais e locais opcionais. Se fornecido, os globais devem ser um dicionário. Se fornecido, os locais podem ser qualquer objeto de mapeamento.

[**exec()**](https://docs.python.org/3/library/functions.html#exec)

Esta função suporta a execução dinâmica de código Python. object deve ser uma string ou um objeto de código. Se for uma string, a string é analisada como um conjunto de instruções Python que é então executada (a menos que ocorra um erro de sintaxe). [1](https://docs.python.org/3/library/functions.html#id2) Se for um objeto de código, ele é simplesmente executado. Em todos os casos, espera-se que o código executado seja válido como entrada de arquivo (consulte a seção [Entrada de arquivo](https://docs.python.org/3/reference/toplevel_components.html#file-input) no Manual de Referência). Esteja ciente de que as instruções [nonlocal](https://docs.python.org/3/reference/simple_stmts.html" \l "nonlocal), [yield](https://docs.python.org/3/reference/simple_stmts.html" \l "yield)e [return](https://docs.python.org/3/reference/simple_stmts.html" \l "return) não podem ser usadas fora das definições de função, mesmo dentro do contexto do código passado para a [exec()](https://docs.python.org/3/library/functions.html" \l "exec" \o "executivo)função. O valor de retorno é None.

[**filter()**](https://docs.python.org/3/library/functions.html#filter)

Construa um iterador a partir desses elementos de iterável para os quais a função retorna true. iterável pode ser uma sequência, um contêiner que suporta iteração ou um iterador. Se function é None, a função de identidade é assumida, ou seja, todos os elementos de iterable que são falsos são removidos.

[**float()**](https://docs.python.org/3/library/functions.html#float)

Retorna um número de ponto flutuante construído a partir de um número ou string x .

[**format()**](https://docs.python.org/3/library/functions.html#format)

Converta um valor em uma representação “formatada”, conforme controlado por format\_spec . A interpretação de format\_spec dependerá do tipo do argumento value ; no entanto, há uma sintaxe de formatação padrão que é usada pela maioria dos tipos internos.

[**frozenset()**](https://docs.python.org/3/library/functions.html#func-frozenset)

Retorne um novo [frozen set](https://docs.python.org/3/library/stdtypes.html" \l "frozenset" \o "congelado) objeto, opcionalmente com elementos retirados de iterable .

[**getattr()**](https://docs.python.org/3/library/functions.html#getattr)

Retorne o valor do atributo nomeado de object . nome deve ser uma string.

[**globals()**](https://docs.python.org/3/library/functions.html#globals)

Retorne o dicionário implementando o namespace do módulo atual. Para código dentro de funções, isso é definido quando a função é definida e permanece a mesma, independentemente de onde a função é chamada.

[**hasattr()**](https://docs.python.org/3/library/functions.html#hasattr)

Os argumentos são um objeto e uma string. O resultado é Truese a string for o nome de um dos atributos do objeto, Falsecaso contrário. (Isto é implementado chamando e vendo se ele levanta um ou não.)getattr(object, name)[AttributeError](https://docs.python.org/3/library/exceptions.html" \l "AttributeError" \o "Erro de atributo)

[**hash()**](https://docs.python.org/3/library/functions.html#hash)

Retorna o valor de hash do objeto (se houver). Os valores de hash são inteiros. Eles são usados ​​para comparar rapidamente as chaves do dicionário durante uma pesquisa no dicionário. Os valores numéricos que comparam iguais têm o mesmo valor de hash (mesmo que sejam de tipos diferentes, como é o caso de 1 e 1.0).

[**help()**](https://docs.python.org/3/library/functions.html#help)

Invoque o sistema de ajuda integrado. (Esta função destina-se ao uso interativo.) Se nenhum argumento for fornecido, o sistema de ajuda interativo será iniciado no console do interpretador.

[**hex()**](https://docs.python.org/3/library/functions.html#hex)

Converta um número inteiro em uma string hexadecimal minúscula prefixada com “0x”. Se x não for um objeto Python [int](https://docs.python.org/3/library/functions.html" \l "int" \o "int), ele deve definir um \_\_index\_\_()método que retorne um inteiro.

[**id()**](https://docs.python.org/3/library/functions.html#id)

Retorna a “identidade” de um objeto.

[**input()**](https://docs.python.org/3/library/functions.html#input)

Se o argumento prompt estiver presente, ele será gravado na saída padrão sem uma nova linha à direita.

[**int()**](https://docs.python.org/3/library/functions.html#int)

Retorna um objeto inteiro construído a partir de um número ou string x , ou retorna 0se nenhum argumento for fornecido.

[**isinstance()**](https://docs.python.org/3/library/functions.html#isinstance)

Retorna Truese o argumento do objeto for uma instância do argumento classinfo ou de uma subclasse (direta, indireta ou [virtual](https://docs.python.org/3/glossary.html#term-abstract-base-class) ) dele.

[**issubclass()**](https://docs.python.org/3/library/functions.html#issubclass)

Retorna Truese class for uma subclasse (direta, indireta ou [virtual](https://docs.python.org/3/glossary.html#term-abstract-base-class) ) de classinfo .

[**iter()**](https://docs.python.org/3/library/functions.html#iter)

Retorne um objeto [iterador .](https://docs.python.org/3/glossary.html" \l "term-iterator)O primeiro argumento é interpretado de forma muito diferente dependendo da presença do segundo argumento.

[**len()**](https://docs.python.org/3/library/functions.html#len)

Retorna o comprimento (o número de itens) de um objeto.

[**list()**](https://docs.python.org/3/library/functions.html#func-list)

Em vez de ser uma função, [list](https://docs.python.org/3/library/stdtypes.html" \l "list" \o "Lista)é na verdade um tipo de sequência mutável, conforme documentado em [Listas](https://docs.python.org/3/library/stdtypes.html#typesseq-list) e [Tipos de Sequência — list, tuple, range](https://docs.python.org/3/library/stdtypes.html#typesseq) .

[**locals()**](https://docs.python.org/3/library/functions.html#locals)

Atualize e retorne um dicionário que representa a tabela de símbolos local atual.

[**map()**](https://docs.python.org/3/library/functions.html#map)

Retorna um iterador que aplica a função a cada item de iterable , gerando os resultados.

[**max()**](https://docs.python.org/3/library/functions.html#max)

Retorna o maior item em um iterável ou o maior de dois ou mais argumentos.

[**memoryview()**](https://docs.python.org/3/library/functions.html#func-memoryview)

Retorna um objeto “memory view” criado a partir do argumento fornecido.

[**min()**](https://docs.python.org/3/library/functions.html#min)

Retorna o menor item em um iterável ou o menor de dois ou mais argumentos.

[**next()**](https://docs.python.org/3/library/functions.html#next)

Recupere o próximo item do [iterador](https://docs.python.org/3/glossary.html" \l "term-iterator) chamando seu [\_\_next\_\_()](https://docs.python.org/3/library/stdtypes.html#iterator.__next__)método. Se o padrão for fornecido, ele será retornado se o iterador estiver esgotado, caso contrário, [StopIteration](https://docs.python.org/3/library/exceptions.html" \l "StopIteration" \o "Parar Iteração)será gerado.

[**object()**](https://docs.python.org/3/library/functions.html#object)

Retorne um novo objeto sem recursos.

[**oct()**](https://docs.python.org/3/library/functions.html#oct)

Converta um número inteiro em uma string octal prefixada com “0o”.

[**open()**](https://docs.python.org/3/library/functions.html#open)

Abra o arquivo e retorne um [objeto de arquivo](https://docs.python.org/3/glossary.html#term-file-object) correspondente . Se o arquivo não puder ser aberto, um [OSError](https://docs.python.org/3/library/exceptions.html" \l "OSError" \o "OSError)será gerado.

'r': aberto para leitura (padrão)

'w': aberto para escrita, truncando o arquivo primeiro

'x': aberto para criação exclusiva, falhando se o arquivo já existir

'a': aberto para escrita, anexando ao final do arquivo se existir

'b': modo binário

't': modo de texto (padrão)

'+': aberto para atualização (leitura e escrita)

[**ord()**](https://docs.python.org/3/library/functions.html#ord)

Dada uma string representando um caractere Unicode, retorne um inteiro representando o ponto de código Unicode desse caractere.

[**pow()**](https://docs.python.org/3/library/functions.html#pow)

Retorne a base ao poder exp ; se o mod estiver presente, retorne a base para a potência exp , modulo mod (computado de forma mais eficiente que ).

[**print()**](https://docs.python.org/3/library/functions.html#print)

Imprima objetos no arquivo de fluxo de texto , separados por sep e seguidos por end . sep , end , file e flush , se presentes, devem ser fornecidos como argumentos de palavra-chave.

[**property()**](https://docs.python.org/3/library/functions.html#property)

Retorna um atributo de propriedade.

[**range**()](https://docs.python.org/3/library/functions.html#func-range)

Em vez de ser uma função, [range](https://docs.python.org/3/library/stdtypes.html" \l "range" \o "variar)é na verdade um tipo de sequência imutável.

[**repr()**](https://docs.python.org/3/library/functions.html#repr)

Retorna uma string contendo uma representação imprimível de um objeto.

[**reversed()**](https://docs.python.org/3/library/functions.html#reversed)

Retorna um [iterador](https://docs.python.org/3/glossary.html" \l "term-iterator) reverso .

[**round()**](https://docs.python.org/3/library/functions.html#round)

Número de retorno arredondado para precisão de n dígitos após o ponto decimal. Se ndigits for omitido ou for None, ele retornará o inteiro mais próximo de sua entrada.

[**set()**](https://docs.python.org/3/library/functions.html#func-set)

Retorne um novo [set](https://docs.python.org/3/library/stdtypes.html" \l "set" \o "definir)objeto, opcionalmente com elementos retirados de iterable .

[**setattr()**](https://docs.python.org/3/library/functions.html#setattr)

Esta é a contrapartida de [getattr()](https://docs.python.org/3/library/functions.html" \l "getattr" \o "getattr). Os argumentos são um objeto, uma string e um valor arbitrário. A string pode nomear um atributo existente ou um novo atributo.

[**slice()**](https://docs.python.org/3/library/functions.html#slice)

Retorna um objeto [slice](https://docs.python.org/3/glossary.html" \l "term-slice) representando o conjunto de índices especificado por . Os argumentos start e step são padronizados para . Os objetos Slice possuem atributos de dados somente leitura , , e que apenas retornam os valores dos argumentos (ou seu padrão).

[**sorted()**](https://docs.python.org/3/library/functions.html#sorted)

Retorne uma nova lista ordenada dos itens em iterable .

[**staticmethod()**](https://docs.python.org/3/library/functions.html#staticmethod)

Transforme um método em um método estático.

[**str()**](https://docs.python.org/3/library/functions.html#func-str)

Retorne uma [str](https://docs.python.org/3/library/stdtypes.html" \l "str" \o "str)versão de object .

[**sum()**](https://docs.python.org/3/library/functions.html#sum)

As somas começam e os itens de um iterável da esquerda para a direita e retornam o total. Os itens do iterável são normalmente números e o valor inicial não pode ser uma string.

[**super()**](https://docs.python.org/3/library/functions.html#super)

Retorne um objeto proxy que delega chamadas de método para uma classe pai ou irmã do tipo . Isso é útil para acessar métodos herdados que foram substituídos em uma classe.

[**tuple()**](https://docs.python.org/3/library/functions.html#func-tuple)

Em vez de ser uma função, [tuple](https://docs.python.org/3/library/stdtypes.html" \l "tuple" \o "tupla)é na verdade um tipo de sequência imutável.

[**type()**](https://docs.python.org/3/library/functions.html#type)

Com um argumento, retorne o tipo de um objeto .

[**vars()**](https://docs.python.org/3/library/functions.html#vars)

Retorne o [\_\_dict\_\_](https://docs.python.org/3/library/stdtypes.html#object.__dict__)atributo para um módulo, classe, instância ou qualquer outro objeto com um [\_\_dict\_\_](https://docs.python.org/3/library/stdtypes.html#object.__dict__)atributo.

[**zip()**](https://docs.python.org/3/library/functions.html#zip)

Iterar sobre vários iteráveis ​​em paralelo, produzindo tuplas com um item de cada um.