

# Lendo entrada do teclado

3 minutos

Muitos programas são interativos. Dar suporte à interatividade significa que você tem um programa que é executado de maneira diferente, dependendo da entrada. A pessoa que realiza a inserção de dados em um programa geralmente é um usuário, mas também pode ser outro programa. Há muitas maneiras de enviar entrada para um programa. Duas maneiras comuns são por meio de uma interface gráfica ou de um console.

## Entrada do usuário

Para ler a entrada do teclado, o Python fornece a função `input()`. `input()` lê o que o usuário digita no teclado e o retorna como uma cadeia de caracteres. Aqui está um exemplo que combina `input()` e `print()` para capturar o nome de uma pessoa e exibi-lo na tela:

Python

```
name = input('Enter your name:')  
print(name)
```

A cadeia de caracteres passada como um argumento para a função `input` é o prompt que o usuário verá. Neste exemplo, você está solicitando que o usuário digite o próprio nome ('Insira seu nome'). Depois que o usuário digita um nome e pressiona , a função `input` será retornada. O valor retornado da função é o texto digitado pelo usuário e esse texto é atribuído à variável chamada `name`. A variável `name` então é usada como uma entrada ou um argumento para a função `print`, que mostrará o nome que o usuário inseriu.

Você também pode chamar a função `input` sem um parâmetro:

Python

```
print('What is your name?')  
name = input()  
print(name)
```

Este programa se comportará *quase* da mesma maneira que o primeiro. A diferença é que `print` (por padrão) adiciona uma nova linha à saída.

## Lendo números como entrada

A função `input` sempre retorna o valor digitado como uma *cadeia de caracteres* (texto). Essa escolha faz sentido, porque o usuário poderá inserir qualquer valor que desejar. Mesmo que a entrada *seja* um número válido, ela ainda será retornada como um tipo de cadeia de caracteres da função `input`. Por exemplo:

Python

```
x = input('Enter a number: ')
print(type(x))
```

Executar esse código e inserir o valor "5" exibirá `<class 'str'>`, embora o valor em si seja numérico. Para transformar o valor em uma variável de inteiro verdadeira, você pode usar a função `int()`:

Python

```
x = int(input('Enter a number: '))
print(type(x))
```

Esse código produzirá `<class 'int'>` para o valor '5'. Você pode usar a função `float` da mesma forma se esperar um componente fracionário.

### Importante

E se a entrada *não for* numérica e você a passar para a função `int()`? Como você pode esperar, esse seria um erro e causaria uma falha de runtime. O programa será encerrado nesta instrução; você pode experimentá-lo por conta própria no console interativo do Python. Vamos abordar várias soluções para lidar com esses tipos de erros em módulos futuros.

## Convertendo números em cadeias de caracteres

Você também pode fazer o contrário. O método `str()` adotará um valor inteiro ou flutuante e o transformará em uma cadeia de caracteres. Será necessário chamar o método `str()` se você quiser que o exemplo de código abaixo funcione. A conversão garante que o inteiro, no respectivo formulário de cadeia de caracteres, seja concatenado à cadeia de caracteres à esquerda.

Python

```
x = 5  
print('The number is ' + str(x))
```

---

**Todas as unidades foram concluídas:**

Concluir módulo

---