

# Entrada de dados utilizando Pyhton

A entrada de dados é o processo de receber informações ou dados para um programa ou sistema computacional. Em outras palavras, é a maneira como os usuários inserem dados em um sistema para serem processados.

Esses dados podem ser de vários tipos, como números, textos, imagens, sons, entre outros. A entrada de dados é uma etapa fundamental em muitos programas, pois é por meio dela que o programa pode realizar suas operações e fornecer resultados aos usuários.

O comando "input" na linguagem de programação Python é um exemplo de como os programadores podem receber entradas de dados em seus programas.

## O comando input

O comando 'input' é utilizado para solicitar uma entrada de dados do usuário durante a execução do programa em Python. Ele permite que o usuário forneça informações importantes para a execução do código.

### Passo 1: Entendendo o comando input

O comando input é uma função built-in do Python, que solicita uma entrada do usuário na forma de uma string. Ele pode ser utilizado para receber diversos tipos de dados, como números, textos, booleanos, entre outros.

### Passo 2: Sintaxe do comando input

A sintaxe básica do comando input é a seguinte:

```
In [ ]: variavel = input("Mensagem de solitação de entrada")
```

Onde "variavel" é o nome da variável que irá armazenar a entrada do usuário e "Mensagem de solicitação de entrada" é a mensagem que será exibida para o usuário, solicitando a entrada.

### Passo 3: Exemplo de uso do comando input

Vamos ver um exemplo simples de uso do comando input. Nesse exemplo, vamos solicitar que o usuário informe o seu nome e, em seguida, vamos exibir uma mensagem de boas-vindas.

```
In [1]: nome = input("Digite o seu nome: ")
        print("Olá,", nome, "! Bem-vindo!")
```

Digite o seu nome: Professor Claudomir  
Olá, Professor Claudomir ! Bem-vindo!

Nesse exemplo, o comando "input" solicita que o usuário digite o seu nome. Em seguida, o nome é armazenado na variável "nome". Por fim, é exibida uma mensagem de boas-vindas para o usuário, utilizando o nome informado.

### Passo 4: Convertendo a entrada do usuário para outro tipo de dado

É importante lembrar que o input() sempre recebe uma string como entrada, mesmo que o usuário digite um número. Portanto, se for necessário realizar operações matemáticas com o valor digitado, é necessário convertê-lo para o tipo numérico adequado. Isso pode ser feito utilizando as funções int() ou float(), dependendo do tipo de dado desejado.

```
In [2]: idade = int(input("Digite a sua idade: "))
        altura = float(input("Digite a sua altura (em metros): "))
```

Digite a sua idade: 33  
Digite a sua altura (em metros): 1.74

Nesse exemplo, o comando "input" solicita que o usuário informe a sua idade e a sua altura. Em seguida, a função "int()" é utilizada para converter a entrada do usuário para um número inteiro, armazenado na variável "idade". Já a função "float()" é utilizada para converter a entrada do usuário para um número decimal, armazenado na variável "altura".

## Passo 5: Como tratar erros de conversão de tipos de dados?

Ao utilizar o comando "**input()**", é comum que o usuário forneça uma entrada em um formato que não é compatível com o que o programa espera. Por exemplo, se um programa espera receber um número inteiro, mas o usuário digita uma string, uma exceção "**ValueError**" será gerada. Isso ocorre porque o programa não consegue converter automaticamente a string em um número inteiro.

Para tratar esse tipo de erro, podemos utilizar o comando "**try-except**". O comando "**try**" tenta executar um bloco de código, e se uma exceção ocorrer, o bloco de código dentro do comando "**except**" será executado.

```
In [3]: try:
        idade = int(input("Digite sua idade: "))

        except ValueError:
            print("Digite uma idade válida.")
```

Digite sua idade: teste  
Digite uma idade válida.

Nesse exemplo, o programa solicita ao usuário que digite sua idade. O programa, então, tenta converter a entrada do usuário em um número inteiro utilizando a função int(). Se a entrada não puder ser convertida em um número inteiro, uma exceção ValueError será gerada e o programa imprimirá uma mensagem de erro.

## Exercícios

- 1.Crie um programa que peça ao usuário que digite seu nome e, em seguida, imprima uma mensagem de boas-vindas na tela.
- 2.Escreva um programa que solicite ao usuário que digite sua idade e, em seguida, calcule e exiba o número de dias que o usuário viveu.
- 3.Crie um programa que peça ao usuário dois números e, em seguida, imprima a soma dos dois números.
- 4.Faça um programa que peça ao usuário um número inteiro e, em seguida, exiba uma mensagem indicando se o número é par ou ímpar.
- 5.Escreva um programa que solicite ao usuário o seu salário mensal e o número de meses trabalhados. Em seguida, calcule e exiba o seu salário anual.
- 6.Peça para o usuário digitar um ângulo em graus e calcule o seu seno, cosseno e tangente utilizando as funções da biblioteca math.
- 7.Peça para o usuário digitar dois valores que representarão os catetos de um triângulo retângulo e utilize a função hypot da biblioteca math para calcular a hipotenusa desse triângulo.
- 8.Escreva um programa que peça ao usuário para digitar seu nome completo e retorne apenas o primeiro nome.
- 9.Escreva um programa que peça ao usuário para digitar uma palavra e retorne o número de caracteres que a palavra tem.
- 10.Escreva um programa que peça ao usuário para digitar seu nome completo e retorne apenas o sobrenome.
- 11.Crie um programa que peça ao usuário para digitar duas palavras e verifique se elas possuem o mesmo comprimento.

12.Crie um programa que peça ao usuário para digitar sua idade e informe se ele é maior de idade ou não.

In [ ]: