



PYTHON
ACADEMY

FUNÇÃO INPUT E PRINT NO PYTHON: ENTRADA E SAÍDA DE DADOS

Guia de I/O: input() para entrada, print() para saída, conversão de tipos, sep/end, f-strings com print, casos práticos (calculadora, menu, validação).

Este ebook foi gerado por



Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA 

 PRIMEIRO CAPÍTULO 100% GRÁTIS

Olá **Pythonista**!

Nesse post vamos falar sobre entrada de dados, saída de dados, a função `input()` do Python, a função `print()` e **MUITO MAIS!**

Com essas funções, podemos tornar um simples script Python muito mais dinâmico e poderoso, ao possibilitar a interação com o usuário.

Veremos como utilizar a função `input` para adquirir dados de entrada passados pelo usuário, assim como mostrar informações aos usuários com a função `print`.

Então, vamos **nessa**!

Entrada de dados com a Função

`input`

Vamos começar falando sobre **entrada do usuário**.

Esse é um artifício muito comum em programação, quando precisamos que o usuário passe ao programa algum tipo de dado.

Em Python, fazemos isso utilizando a função `input()`, que é literalmente ‘entrada’ em inglês.

A função `input()` recebe como parâmetro uma string que será mostrada como auxílio ao usuário, geralmente o informando que tipo de dado o programa está aguardando receber.

Vejo um exemplo:

```
input("Escreva seu nome: ")
```

Quando fazemos isso, o terminal ou prompt de comando irá: - Mostrar o texto "Escreva seu nome: " na tela; e - Aguardar até que o usuário digite alguma informação.

Após a entrada de dados pelo usuário, o programa irá continuar sua execução, passando para as instruções seguintes.

No terminal, aparecerá algo mais ou menos assim:

```
>>> Escreva seu nome: Lucas
```

Saída de dados com a Função `print`

A função para imprimir dados em Python é a função `print()`.

Ela é responsável por mostrar valores em seu terminal:

```
print('Olá Mundo')
```

Imprime a string no terminal:

```
Olá Mundo
```

O parâmetro `sep=` da função `print`

Por padrão, quando utilizamos virgula para separar os itens, a função `print` utiliza espaços para separar cada saída.

Porém, podemos utilizar o parâmetro `sep=` para definir um caractere de separação.

Entenda no exemplo a seguir:

```
print('Dia', 'Mês', 'Ano', sep='/')
print('ontem', 'Hoje', 'Amanhã', sep='-')
print("B", "n", "n", ".", sep='a')
```

Observe a saída com os caracteres definidos:

```
Dia/Mês/Ano
ontem-Hoje-Amanhã
Banana.
```

Perceba que a função `print` incluiu o caracter indicado em `sep=` como separador dos dados a serem mostrados ao usuário!

O parâmetro `end=` da função `print`

Por padrão, a função `print` utiliza a quebra de linha (`\n`) como último caracter.

O parâmetro `end=` é responsável por alterar esse comportamento, possibilitando ao desenvolvedor trocar qual caracter será adicionado ao final do dado impresso no terminal.

Vamos entender melhor no exemplo a seguir:

```
# Exemplo com fim de linha sem nenhum caracter
print('Vamos estudar Na ', end='')
print('Python Academy')

# Exemplo com fim de linha igual à ->
print('As rosas são', end=' -> ')
print('Vermelhas')

# Exemplo com fim de linha igual à :
print("Quantidade", end=': ')
print(40)
```

Dessa forma, a saída será:

```
Vamos estudar Na Python Academy
As rosas são -> Vermelhas
Quantidade: 40
```

💡 Estou construindo o **DevBook**, uma plataforma que usa IA para criar ebooks técnicos — com código formatado e exportação em PDF. Não deixe de conferir!

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

Juntando entrada e saída com `input` e `print`

Agora que conhecemos as duas funções, podemos juntá-las para fazer programas mais completos.

Vamos armazenar o nome do usuário utilizando `input` e mostrar o dado para ele com `print`.

Podemos juntar e imprimir vários itens em `print()`, basta separá-los por vírgula, como no exemplo abaixo:

```
nome = input("Escreva seu nome: ")

print('Seu nome é:', nome)
```

Perceba que a função `input` retorna para nós aquilo que o usuário digitou. Portanto, nesse caso, o que o usuário digitar será passado à variável `nome` !

Observe a saída:

```
Escreva seu nome: Erick
Seu nome é: Erick
```

Utilizando `print` para gravar dados em arquivos

A função `print()` também funciona para gravar dados em arquivos.

Para isso, utilizamos o parâmetro `file=` da função `print`.

Também precisamos de um arquivo aberto, o que é feito utilizando-se a função `open` .

Veja o exemplo abaixo:

```
with open('arquivo.txt', 'w') as arquivo:
    print("Escreva isso dentro do arquivo,", file=arquivo)
    print("Escreva outra linha dentro do arquivo.", file=arquivo)
```

Se abrirmos o `arquivo.txt` , veremos o seguinte conteúdo:


```
Escreva isso dentro do arquivo,  
Escreva outra linha dentro do arquivo.
```

Conclusão

Vimos nesse post como utilizar as funções `input` e `print` e como utilizá-las para pegar dados **do** usuários e mostrar dados **para** o usuário.

Esse é um conhecimento básico que todo programador Python, os chamados **Pythoistas**, devem saber!

Se ficou com alguma dúvida, fique à vontade para deixar um comentário no box aqui embaixo! Será um prazer te responder! 😊

Não se esqueça de conferir!



DevBook

Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



 Syntax Highlight

 Infográficos feitos por IA

 Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

 Edite em Markdown em Tempo Real

TESTE AGORA 

 PRIMEIRO CAPÍTULO 100% GRÁTIS