

FastAPI Aprender

Variáveis de Ambiente



Dica

Se você já sabe o que são "variáveis de ambiente" e como usá-las, pode pular esta seção.

Uma variável de ambiente (também conhecida como "**env var**") é uma variável que existe **fora** do código Python, no **sistema operacional**, e pode ser lida pelo seu código Python (ou por outros programas também).

Variáveis de ambiente podem ser úteis para lidar com **configurações** do aplicativo, como parte da **instalação** do Python, etc.

Criar e Usar Variáveis de Ambiente

Você pode **criar** e usar variáveis de ambiente no **shell (terminal)**, sem precisar do Python:

Linux, macOS, Windows Bash

```
bash
```

Windows PowerShell

```
bash
```

fast →

💬 Criar uma variável de ambiente MY_NAME

```
$ $Env:MY_NAME = "Wade Wilson"
```

💬 Usá-la com outros programas, como

```
$ e
```

Ler Variáveis de Ambiente no Python

Você também pode criar variáveis de ambiente **fora** do Python, no terminal (ou com qualquer outro método) e depois **lê-las no Python**.

Por exemplo, você poderia ter um arquivo `main.py` com:

```
import os

name = os.getenv("MY_NAME", "World")
print(f"Hello {name} from Python")
```

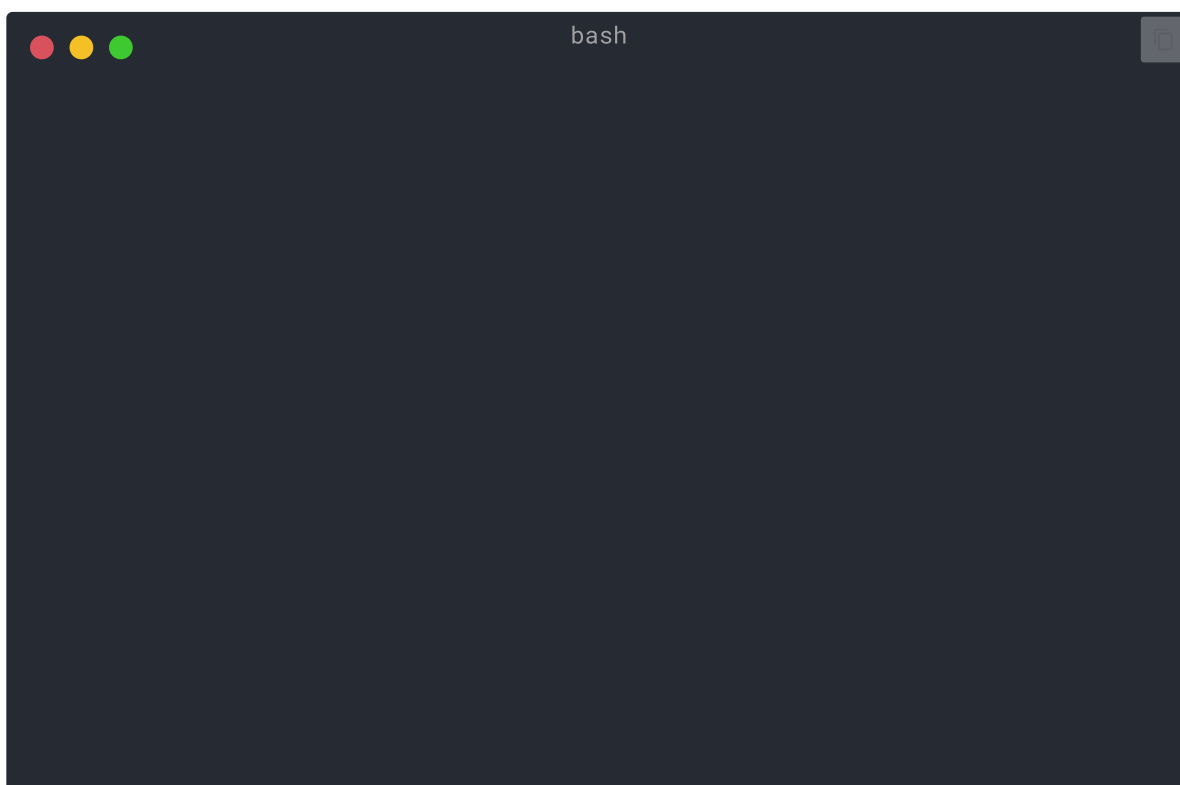
Dica

O segundo argumento para `os.getenv()` [\[↗\]](#) é o valor padrão a ser retornado.

Se não for fornecido, é `None` por padrão, Aqui fornecemos `"World"` como o valor padrão a ser usado.

Então você poderia chamar esse programa Python:

Linux, macOS, Windows Bash



Windows PowerShell



💬 Como não definimos a variável de ambiente, obtemos o valor padrão

```
Hello World from Python
```

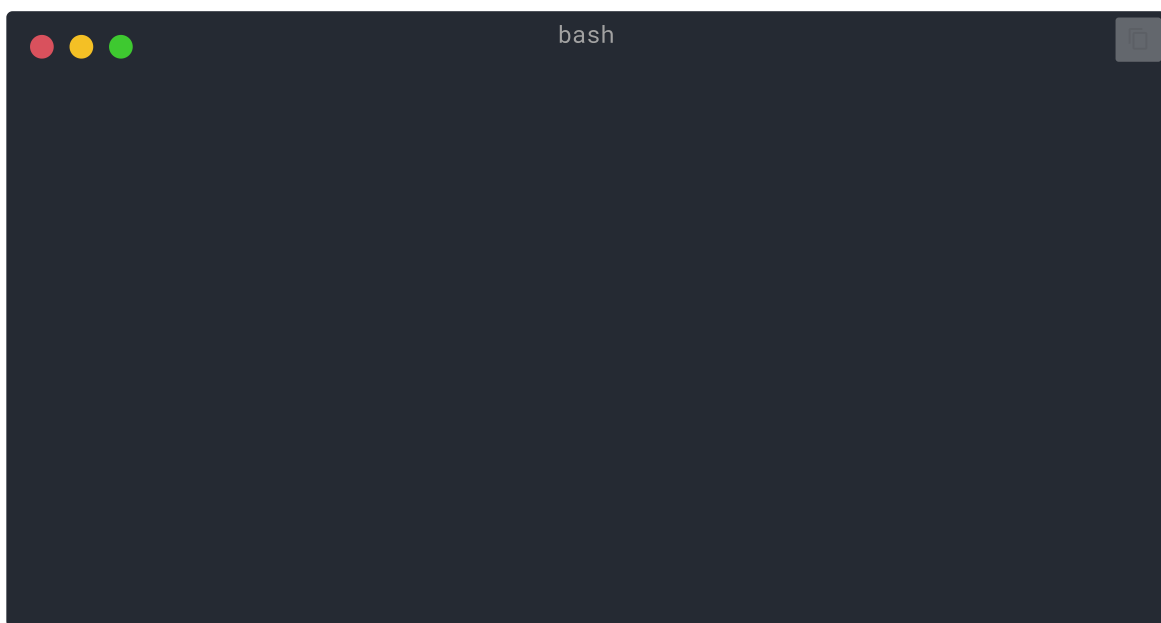
💬 Mas se criarmos uma variável de ambiente primeiro

```
$ $Env:MY_N █
```

Como as variáveis de ambiente podem ser definidas fora do código, mas podem ser lidas pelo código e não precisam ser armazenadas (com versão no `git`) com o restante dos arquivos, é comum usá-las para configurações ou **definições**.

Você também pode criar uma variável de ambiente apenas para uma **invocação específica do programa**, que só está disponível para aquele programa e apenas pela duração dele.

Para fazer isso, crie-a na mesma linha, antes do próprio programa:



Dica

Você pode ler mais sobre isso em [The Twelve-Factor App: Config](#) [↗].

Tipos e Validação

Essas variáveis de ambiente só podem lidar com **strings de texto**, pois são externas ao Python e precisam ser compatíveis com outros programas e com o resto do sistema (e até mesmo com diferentes sistemas operacionais, como Linux, Windows, macOS).

Isso significa que **qualquer valor** lido em Python de uma variável de ambiente **será uma `str`**, e qualquer conversão para um tipo diferente ou qualquer validação precisa ser feita no código.

Você aprenderá mais sobre como usar variáveis de ambiente para lidar com **configurações do aplicativo** no [Guia do Usuário Avançado - Configurações e Variáveis de Ambiente](#) ↗.

Variável de Ambiente `PATH`

Existe uma variável de ambiente **especial** chamada `PATH` que é usada pelos sistemas operacionais (Linux, macOS, Windows) para encontrar programas para executar.

O valor da variável `PATH` é uma longa string composta por diretórios separados por dois pontos `:` no Linux e macOS, e por ponto e vírgula `;` no Windows.

Por exemplo, a variável de ambiente `PATH` poderia ter esta aparência:

Linux, macOS

```
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

Isso significa que o sistema deve procurar programas nos diretórios:

- `/usr/local/bin`
- `/usr/bin`
- `/bin`
- `/usr/sbin`
- `/sbin`

Windows

```
C:\Program Files\Python312\Scripts;C:\Program Files\Python312;C:\Windows\System32
```

Isso significa que o sistema deve procurar programas nos diretórios:

- `C:\Program Files\Python312\Scripts`
- `C:\Program Files\Python312`
- `C:\Windows\System32`

Quando você digita um **comando** no terminal, o sistema operacional **procura** o programa em **cada um dos diretórios** listados na variável de ambiente `PATH`.

Por exemplo, quando você digita `python` no terminal, o sistema operacional procura um programa chamado `python` no **primeiro diretório** dessa lista.

Se ele o encontrar, então ele o **usará**. Caso contrário, ele continua procurando nos **outros diretórios**.

Instalando o Python e Atualizando o `PATH`

Durante a instalação do Python, você pode ser questionado sobre a atualização da variável de ambiente `PATH`.

Linux, macOS

Vamos supor que você instale o Python e ele fique em um diretório `/opt/custompython/bin`.

Se você concordar em atualizar a variável de ambiente `PATH`, o instalador adicionará `/opt/custompython/bin` para a variável de ambiente `PATH`.

Poderia parecer assim:

```
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/custompython/bin
```

Dessa forma, ao digitar `python` no terminal, o sistema encontrará o programa Python em `/opt/custompython/bin` (último diretório) e o utilizará.

Windows

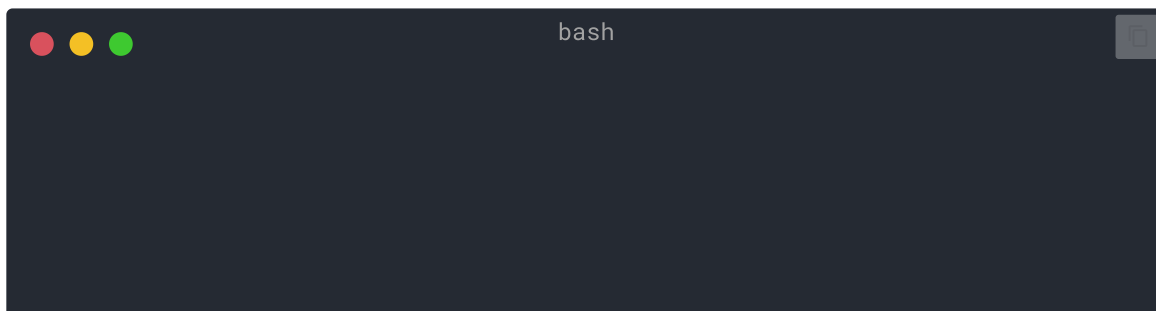
Digamos que você instala o Python e ele acaba em um diretório `C:\opt\custompython\bin`.

Se você disser sim para atualizar a variável de ambiente `PATH`, o instalador adicionará `C:\opt\custompython\bin` à variável de ambiente `PATH`.

```
C:\Program Files\Python312\Scripts;C:\Program  
Files\Python312;C:\Windows\System32;C:\opt\custompython\bin
```

Dessa forma, quando você digitar `python` no terminal, o sistema encontrará o programa Python em `C:\opt\custompython\bin` (o último diretório) e o utilizará.

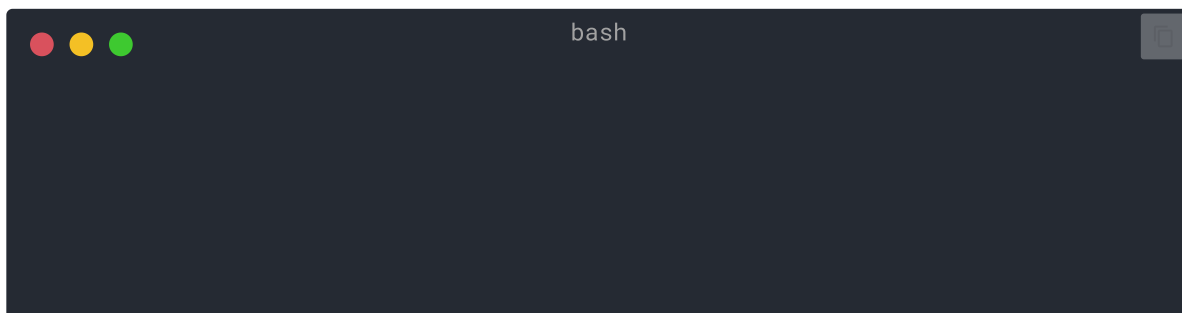
Então, se você digitar:



Linux, macOS

O sistema **encontrará** o programa `python` em `/opt/custompython/bin` e o executará.

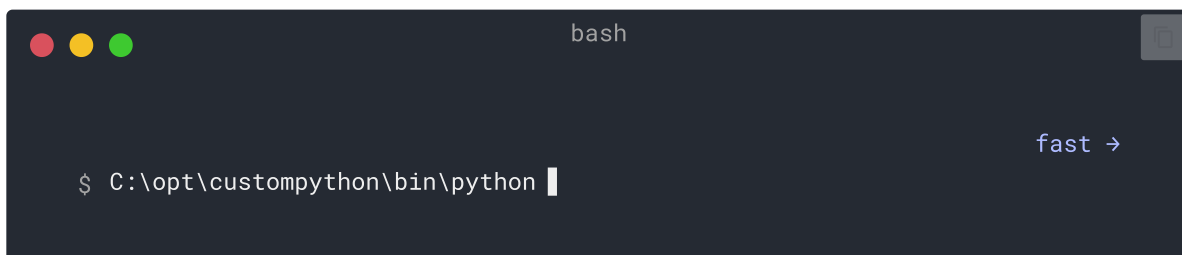
Seria aproximadamente equivalente a digitar:



Windows

O sistema **encontrará** o programa `python` em `C:\opt\custompython\bin\python` e o executará.

Seria aproximadamente equivalente a digitar:



Essas informações serão úteis ao aprender sobre [Ambientes Virtuais ↗](#).

Conclusão

Com isso, você deve ter uma compreensão básica do que são **variáveis de ambiente** e como usá-las em Python.

Você também pode ler mais sobre elas na [Wikipedia para Variáveis de Ambiente ↗](#).

Em muitos casos, não é muito óbvio como as variáveis de ambiente seriam úteis e aplicáveis imediatamente. Mas elas continuam aparecendo em muitos cenários diferentes quando você está desenvolvendo, então é bom saber sobre elas.

Por exemplo, você precisará dessas informações na próxima seção, sobre [Ambientes Virtuais](#).