

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```



## Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and default arguments. [More about defining functions in Python](#)

[1](#) [2](#) [3](#) [4](#) [5](#)

# Design de Código



## O que podemos aprender com as PEPs do Python?

Daniele Nazaré

Desenvolvedora de Software



THE  
DEVELOPER'S  
CONFERENCE

## Sobre Mim

Sou Mineirinha e Bacharel em Engenharia de Computação e contribuo para a comunidade de desenvolvedores ministrando palestra em eventos. Já tive a oportunidade de palestrar na TDC, Brazil JS On the Road, Python Brazil 2019. Além disso faço parte do projeto de Computação Quântica, The Quantum Blockchain Project.

<https://br.linkedin.com/in/danielenazare>

<https://github.com/danynt14>



# Introdução



THE  
DEVELOPER'S  
CONFERENCE

"As **PEPs** (Index of Python Enhancement Proposals) são arquivos que são liberados pela comunidade para apresentar alguma funcionalidade nova, uma boa prática, um propósito ou um ambiente."

[[PEP1](#)]

# PEPs



# RFC

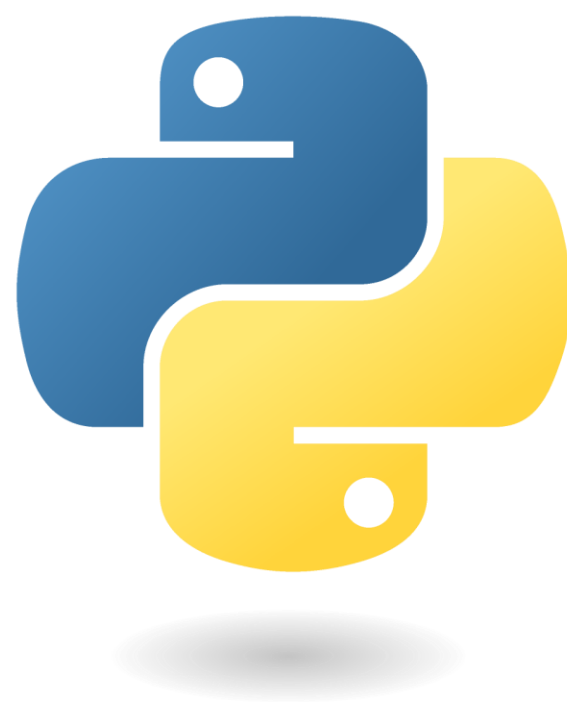


*"As PEPs são Documentos Equivalente as RFCs, Porém com Propósitos Diferentes "*



THE  
DEVELOPER'S  
CONFERENCE

Quem é responsável  
pela Curadoria das  
PEPs na comunidade?



python  
SOFTWARE FOUNDATION







THE  
DEVELOPER'S  
CONFERENCE

## PEP 8

*Guia de Estilo  
Para Código em  
Python*

## PEP 7

*Guia de Estilo  
Código em C*

# Criadores da PEP 8



"É um Documento que fornece as principais convenções de desenvolvimento padrão de código na versão vigente do Python"

[\[PEP8\]](#)



Nick Coghlan

Guido Van Rossum

Barry Warsaw

# Por que PEP 7? [[PEP7](#)]

Existem módulos e rotinas em Python escrito em C!







THE  
DEVELOPER'S  
CONFERENCE

# Quando nos convém seguir as regras?

Para manter a legibilidade do código.

Manter a consistência dentro do projeto.

Manutenibilidade do Projeto

# Quando nos convém quebrar as regras?

Quando com a aplicação da PEP8 deixar seu código menos legível.

Quando a reengenharia do código inconsistente for tão caro, levando ao ponto de ter que continuar.

Quando o código anterior mantém outra diretriz, tal que seja necessário segui-lo.

```
print 'Quando o código em Python não é compatível com a  
versão anterior!' if 2.5 != 3.9
```



THE  
DEVELOPER'S  
CONFERENCE

Ferramenta que faz análise  
sintática e léxica no código afim  
de encontrar possíveis  
estruturas que comprometam o  
funcionamento do teu código.



# Linter

## PEP8

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```



## Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

[1](#) [2](#) [3](#) [4](#) [5](#)

# Conclusão

**import this**



THE  
DEVELOPER'S  
CONFERENCE

[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
987
```



## Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

[1](#) [2](#) [3](#) [4](#) [5](#)

# Bônus

## Características da Linguagem Python



THE  
DEVELOPER'S  
CONFERENCE



# Características de Projeto da Python

- ☐ Linguagem de Altíssimo Nível
- ☐ Interpretada
- ☐ Interoperabilidade
- ☐ Tipagem Dinâmica
- ☐ Leitura fácil
- ☐ Orientada a indentação
- ☐ Case Sensitive



THE  
DEVELOPER'S  
CONFERENCE



# Características de Projeto da Python

- ☐ Script
- ☐ Legível
- ☐ Sintaxe concisa e clara
- ☐ Multiparadigma
  - ☐ Orientada a Objetos
  - ☐ Imperativa
  - ☐ Funcional
  - ☐ Procedural



THE  
DEVELOPER'S  
CONFERENCE





THE  
DEVELOPER'S  
CONFERENCE

vem = pra  
+ python

Muito  
Obrigado  
!

# Referências

- [Python]. Disponível em: <https://www.python.org/dev/> . Acessado 25 de out de 2020.
- [Python Brasil]. Disponível em: <https://wiki.python.org.br/GuiaDeEstilo> . Acessado 25 de out de 2020.
- [The Hitchhiker's Guide to Python!]. Disponível em: <https://docs.python-guide.org/>. Acessado 25 de out de 2020.
- [Kenneth Reitz & Tanya Schlusser] The Hitchhiker's Guide to Python: BEST PRACTICES FOR DEVELOPMENT. NOVATEC .
- [Wikipedia] Disponível em: <https://pt.wikipedia.org/wiki/Python>. Acessado 25 de out de 2020.