

The background of the slide features a wide-angle photograph of a desert landscape, likely Zion National Park, showing tall, layered red rock formations against a clear blue sky.

Introdução à Linguagem **Python**

Básico

Prof. Cláudio A. Fleury
Abr/2021

Conteúdo

- O que é Python?
- Histórico
- Filosofia da Linguagem
- Características
- Computação Científica
- Python Científico
- Interpretadores: Padrão X Interativo (IPython)
- Online ou IDE Local?
- Distribuições: Anaconda, WinPython, ...
- Hello World!

O que é Python?



pythonTM

- É uma linguagem de programação:
 - de alto nível
 - interpretada
 - interativa
 - versátil
 - de código aberto
 - de uso geral
 - legível pelos seres humanos
 - orientada:
 - a objetos
 - imperativa
 - funcional
 - estruturada/procedural*

Histórico

- Inicialmente desenvolvida por **Guido van Rossum** em 1989
- Lançada oficialmente em 1999
- Rossum
 - Trabalhava na *Centrum Wiskunde & Informatica* (CWI), Holanda, no projeto da linguagem sucessora da ABC¹ que seria capaz de lidar com exceções e interagir com o sistema operacional **Amoeba**²
 - Fã do grupo humorístico Monty **Python** - programa de comédia *Monty Python's Flying Circus*, TV inglesa BBC, 1969-1974

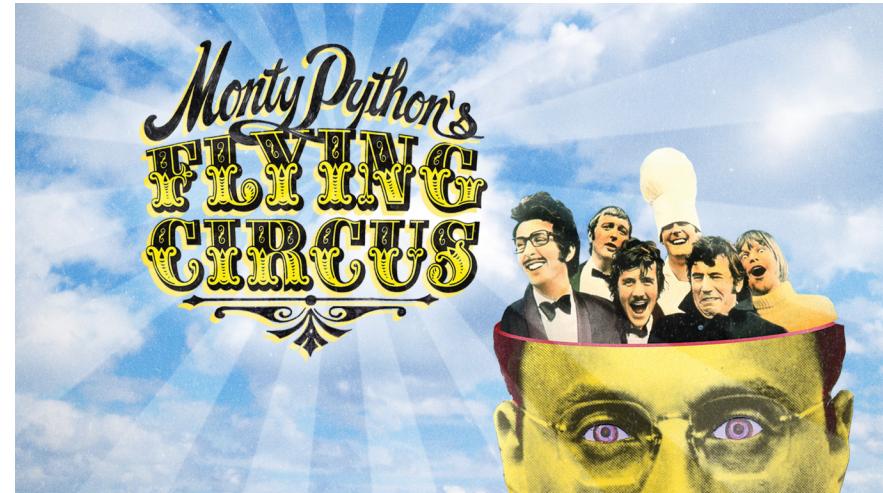


¹ baseada na ling. SETL, que é uma ling. de progr. dos anos 60, de "altíssimo nível", baseada na matemática de conjuntos

² sistema distribuído baseado em microkernel

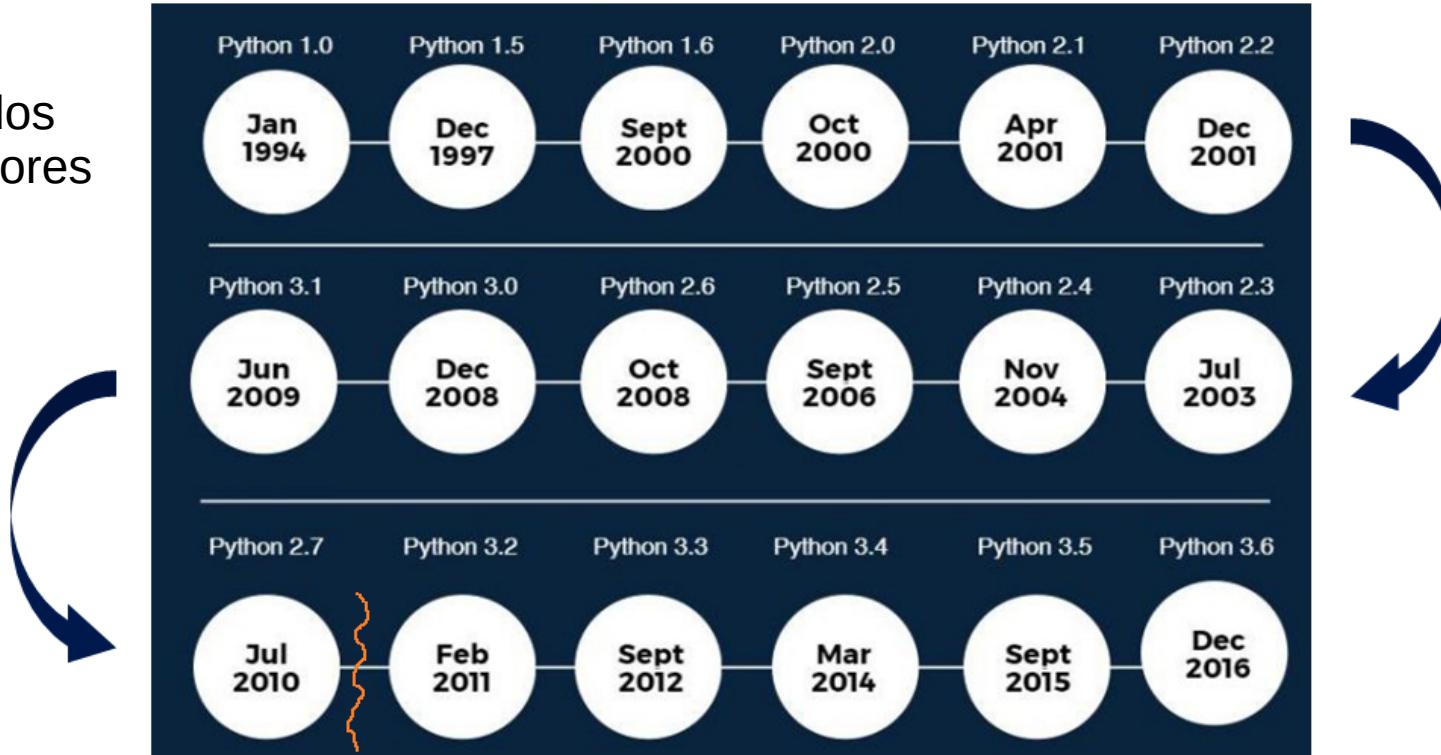
Histórico

O ícone da Ling. Python foi desenhado por sua irmã (*designer*) e mostra duas cobras estilizadas, embora Guido tenha feito uma homenagem ao grupo Monty Python



Histórico

Versões dos Interpretadores



Filosofia da Linguagem

- Representada por manifesto escrito por Guido
 - Bonito é melhor que feio
 - Explícito é melhor que implícito
 - Simples é melhor que complexo
 - Complexo é melhor que complicado
 - Legibilidade é importante

A filosofia de design minimalista do Python significa que ele enfatiza a legibilidade do código. Ele também fornece uma estrutura de sintaxe que permite aos programadores expressar conceitos em menos linhas de código do que em linguagens como C ++ e Java. Além disso, o Python fornece os meios para escrever programas que podem ser facilmente ampliados.

Filosofia da Linguagem

- O Zen do Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.

- Continuação...

- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one -- obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

Características

- A intenção de **Rossum** era projetar uma linguagem
 - com um pequeno núcleo
 - uma biblioteca padrão abrangente
 - um interpretador (respostas rápidas durante o desenvolvimento)
 - e de quebra, que fosse facilmente extensível...

Características

- Tipificação dinâmica de variáveis
- Gerenciamento automático de memória
- Biblioteca padrão (nativa) bem abrangente
- Frequentemente usada como linguagem de **script**
- Programa pode ser compilado (executável)
- Sintaxe simples, de fácil aprendizagem
- Legibilidade enfatizada, reduz custo de manutenção

Características

- **Legível e Interpretada:** Python é uma linguagem muito legível e cada instrução é traduzida individualmente e executada antes da instrução seguinte.
- **Fácil de aprender:** Aprender Python é fácil por ela ser uma linguagem expressiva e de alto nível.
- **Multiplataforma:** está disponível para execução em vários sistemas operacionais, tais como: Mac OS, MS-Windows, Linux, Unix, Oracle Solaris etc.
- **Open Source:** Python é uma linguagem de programação de código aberto.
- **Biblioteca Padrão Abrangente:** a ling. Python tem uma grande biblioteca padrão com códigos e funções úteis que podem ser usados enquanto se escreve código em Python.
- **Gratuita:** a ling. Python é gratuita para download e uso.
- **Manipulação de Exceção:** Uma exceção é um evento que pode ocorrer durante a execução do programa e que interrompe o fluxo normal do programa. A ling. Python permite o tratamento de exceções, o que significa que podemos escrever códigos menos propenso a erros e testar vários cenários que possam provocar uma exceção mais tarde.
- **Recursos Avançados:** geradores e abrangência de lista (*list comprehension*). Veremos esses recursos mais tarde.
- **Gerenciamento Automático de Memória:** a memória é limpa e liberada automaticamente. Você não precisa se preocupar em liberar memória em seus códigos.

Computação Científica

- Necessidades dos Cientistas e Engenheiros
 - Obter dados (simulação, controle de experimentos)
 - Manipular e processar dados
 - Visualizar resultados... para entender o que se está fazendo!
 - Comunicar resultados: produzir gráficos para relatórios ou publicações, montar apresentações

Computação Científica

- Soluções Existentes
 - Linguagens Compiladas: C, C++, Fortran, Pascal etc.
 - Vantagens
 - Muito rápidas. Compiladores muito otimizados. Para cálculos pesados, é difícil superar a performance dessas linguagens
 - Bibliotecas científicas muito otimizadas foram escritas para essas linguagens. Exemplos: BLAS (operações vetoriais/matriciais), LINPACK/LAPACK (álgebra linear)
 - Desvantagens
 - Uso dolorido: sem interatividade, passos obrigatórios de compilação, sintaxe com muitos símbolos (&, ::, }), gerenciamento manual de memória
 - Essas linguagens são difíceis para quem não é cientista da computação

Computação Científica

- Soluções Existentes
 - Linguagens Interpretadas: **Matlab**
 - Vantagens
 - Coleção de bibliotecas ricas (numerosos algoritmos) para diferentes domínios
 - Execução rápida porque essas bibliotecas são escritas em linguagem compilada
 - Ambiente de desenvolvimento agradável: ajuda completa e bem organizada, editor integrado, facilidades de depuração etc.
 - Apoio comercial e técnico disponíveis
 - Desvantagens
 - A linguagem base é pobre e pode se tornar restritiva para usuários avançados
 - Não gratuita

Computação Científica

- Soluções Existentes
 - Outras linguagens de script: **Scilab**, **Octave**, **Igor**, **R**, **IDL** etc.
 - Vantagens
 - Fonte aberta (*open-source*), gratuita, ou pelo menos mais barata que o Matlab
 - Alguns recursos podem ser bem avançados (estatísticas em R, figuras em Igor, etc.)
 - Desvantagens
 - Menos algoritmos que em **Matlab**, e a linguagem não é mais avançada
 - Alguns softwares são mais dedicados a um domínio.
Ex: **Gnuplot** ou **xmgrace** para desenho de curvas. Estes programas são muito poderosos, mas restritos a um único tipo de aplicação, tal como traçado gráfico (*plotting*)

Computação Científica

- Soluções Existentes
 - Linguagem **Python**
 - Vantagens
 - Bibliotecas para computação científica muito ricas (porém, um pouco menos que **Matlab**)
 - Linguagem permite a escrita de códigos estruturados e legíveis: "codificamos o que pensamos"
 - Muitas bibliotecas para tarefas além da computação científica
(gerenciamento de servidor Web, acesso à porta serial etc.)
 - Software gratuito e *open-source*, amplamente difundido, com uma comunidade vibrante
 - Desvantagens
 - Ambiente de desenvolvimento rústico, mais orientado ao público *Geek*
 - Não possui todos os algoritmos encontrados em softwares mais especializados ou *toolboxes*

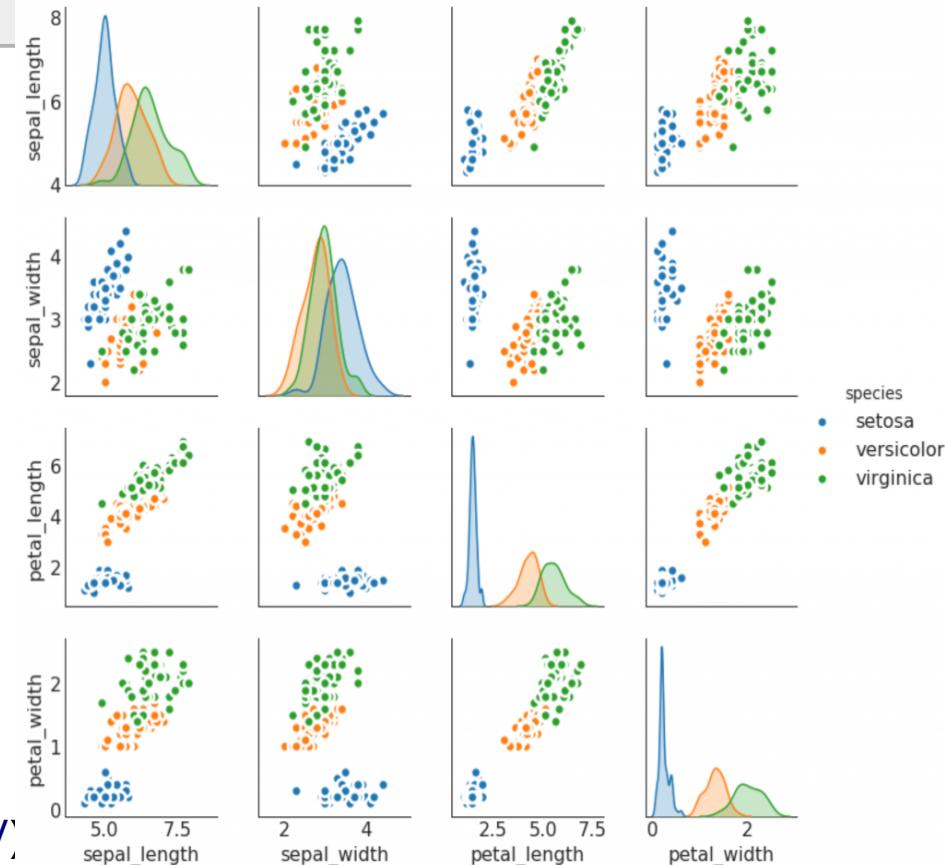
Python Científico

- Diferentemente do Matlab, Scilab ou R, Python não tem um conjunto nativo de módulos para computação científica
- Python é uma linguagem de computação genérica e moderna
 - Tipos de dados (string, int), controle de fluxo, coleções de dados (listas, dicionários), padrões etc.
 - Biblioteca padrão
- Existem vários módulos especializados ou aplicações escritas em Python: protocolos Web, estruturas (*frameworks*) Web etc. e computação científica
- Ferramentas de desenvolvimento
(testes automáticos, geração de documentação)

Python Científico

- Pacotes Científicos

- **Numpy**: fornece objetos do tipo arranjos numéricos (vetor, matriz) e funções para manipulá-los (<http://www.numpy.org/>)
- **Scipy**: rotinas de processamento de dados em alto nível. Otimização, regressão, interpolação etc. (<http://www.scipy.org/>)
- **Matplotlib**: visualização de dados em 2D, gráficos "pronto-para-publicação" (<http://matplotlib.sourceforge.net/>)
- Mayavi: visualização em 3D (<http://code.enthought.com/projects/mayavi/>)



Interpretadores: Padrão X Interativo (Ipython)

- Interpretador Python (padrão)
 - Prompts: >>> ou ... - depende do contexto da entrada do comando
 - >>> indica que o interpretador está aguardando comando para execução
 - Linha sem prompt é linha de resposta ao comando digitado
 - Prompt secundário ... sozinho significa continuação da linha anterior, para encerrá-lo use a tecla [ENTER]
- Interpretador IPython ("interativo")
 - Prompt de entrada de comando é **In[5]**:
 - Prompt de saída de dados aparece na linha seguinte com *prompt* **Out[5]**
 - O valor entre colchetes indica a quantidade de entrada/saída já realizadas

Online ou IDE Local?

- Caso você não queira instalar um ambiente Python em seu computador para escrever seus *scripts* existem alternativas *online*:
 - **Programiz:** <https://www.programiz.com/python-programming/online-compiler/>
 - **W3schools:**
https://www.w3schools.com/python/trypython.asp?filename=demo_compiler
 - **Tutorialspoint:** https://www.tutorialspoint.com/execute_python_online.php
 - **Guru99:** <https://www.guru99.com/execute-python-online.html>
 - **Ideone:** <https://ideone.com>
 - **Repl.it:** <https://replit.com>



Distribuições

- IDE's instaláveis
 - **Anaconda**: contém além do Python, muitas bibliotecas, IDE Spyder (versátil e objetiva) e kit de ferramentas para Ciência de Dados. Mais completo e fácil de ser instalado - <https://www.anaconda.com/products/individual#Downloads>
 - **Pycharm**: interface limpa e personalizável, ideal para quem está iniciando com o Python
 - Atom
 - Visual Studio Code
 - **WinPython**: IDE portátil (sem instalação) com ambiente Spyder

Ferramentas que facilitam a vida do programador:

- * interpretador — onde você roda seu programa
- * editor — onde você escreve seu *script*;
- * depurador (*debugger*) — você usa para localizar erros em seu código.
- * console --- interação direta com objetos, funções e variáveis

Hello World!

- Para exibir a famosa frase “Hello World!” basta digitar o seguinte comando no *shell* do interpretador e pressionar [Enter]:

```
>>> print ("Hello World!")
```

- Usando o repl.it (online):

The screenshot shows the repl.it interface. At the top, there are icons for a terminal, a user profile (Kaw2020), a repository name (helloworld), and a Python logo. To the right is a 'Run' button. Below the header is a code editor window containing a file named 'main.py' with the following code:

```
main.py
1 print("Hello World!")
```

To the right of the code editor is a terminal window titled 'Console'. It displays the output of the executed code:

```
Hello World!
> █
```

Fim