DEPARTAMENTO DE FÍSICA NUCLEAR E ALTAS ENERGIAS

Python básico

Conteúdo

1. Introdução
2. Variáveis. Tipos de dados e operadores
3. Tipos de sequências
4. Fluxo de execução
5. Funções
6. Arquivos e Módulos
7. Programação Orientada à Objeto
8. Exceções

Introdução

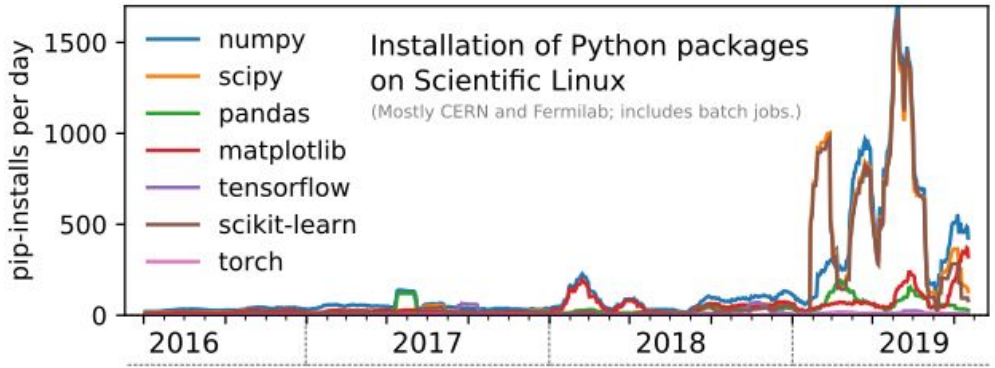
O que é Python?

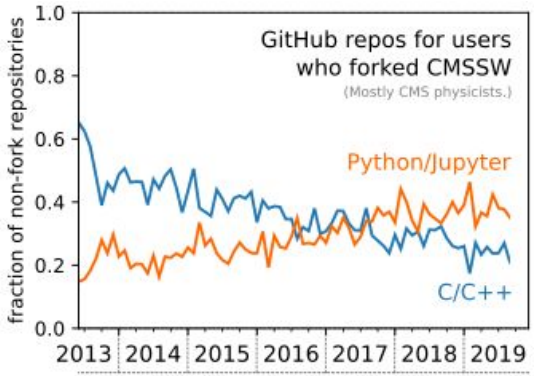
Python é uma linguagem de programação de alto nível, orientada a objetos, moderna e de propósito geral.

- Uma linguagem **interpretada** (oposto à compilação).
  - Ao contrário de C ou Fortran, por exemplo, não se compila código Python antes da execução.
  - Com isso, o código pode ser utilizado interativamente: diversos interpretadores Python estão disponíveis, a partir dos quais comandos e scripts podem ser executados.
- Software lançado **gratuitamente** sob a licença open-source.
- **Multi-plataforma**: Python é disponibilizado para todos os sistemas operacionais mais conhecidos, como Windows, Linux/Unix, MacOS X, muito provavelmente para o OS de seu smartphone, etc.
- Uma linguagem **muito fácil de se ler** com uma **sintaxe limpa** e sem caracteres específicos (*&%${…).
- Uma linguagem para a qual uma vasta variedade de pacotes de alta qualidade estão disponíveis para diversas aplicações, desde frameworks da web até computação científica.
- Uma linguagem de muito **fácil interação com outras linguagens**, em particular com C e C++.

Porque aprender Python?

- Fácil de aprender e praticar
- Fortemente usado no mercado: Google, Facebook(Instagram), Microsoft, Dropbox, Globo.com, etc.
- Utilizando em várias áreas - web, data science, devops, automação, IA e muito mais.





From Jim Privaski: https://indico.cern.ch/event/917675/

Vantagens e Desvantagens

- Vantagens:
  - A principal vantagem é a facilidade de programação, minimizando o tempo necessário para desenvolver, *debugar* (procurar por erros no programa) e manter o código.
  - Linguagem bem projetada que incentiva muitas boas práticas de programação:
  - Programação modular e orientada a objetos, bom sistema para empacotamento e reutilização de código. Isso geralmente resulta em código mais transparente, sustentável e livre de bugs.
  - Documentação totalmente integrada com o código.
  - Uma grande biblioteca padrão e uma grande coleção de pacotes complementares.
  - Muita documentação disponível!
- Desvantagens:
  - Como o Python é uma linguagem de programação interpretada e dinamicamente digitada, a execução do código python pode ser lenta em comparação com as linguagens de programação compiladas estaticamente, como C e Fortran.

Filosofia do Python

Vale citar que tanto a comunidade quanto o próprio time de desenvolvimento seguem a filosofia deixada por um de seus criadores: o Zen do Python, que pode ser acessado ao digitar o comando `import this` em um terminal interativo do Python:

```
The Zen of Python, by Tim Peters

    Bonito é melhor que feio.
    Explícito é melhor que implícito.
    Simples é melhor que complexo.
    Complexo é melhor que complicado.
    Linear é melhor do que aninhado.
    Disperso é melhor que denso.
    Legibilidade conta.
    Casos especiais não são especiais o bastante para quebrar as regras.
    Ainda que praticidade vença a pureza.
    Erros nunca devem passar silenciosamente.
    A menos que sejam explicitamente silenciados.
    Diante da ambiguidade, recuse a tentação de adivinhar.
    Deveria haver um -- e de preferência só um -- modo óbvio para fazer algo.
    Embora esse modo possa não ser óbvio a princípio a menos que você seja
    holandês.
    Agora é melhor que nunca.
    Embora nunca frequentemente seja melhor que já.
    Se a implementação é difícil de explicar, é uma má ideia.
    Se a implementação é fácil de explicar, pode ser uma boa ideia.
    Namespaces são uma grande ideia -- vamos ter mais dessas!
```

Comparação entre C++, Python e Shell

Comparação entre um simples loop sobre os elementos de uma sequência

```
#!/bin/bash
ARRAY_OF_WORDS=(this is my array of words)
for WORD in ${ARRAY_OF_WORDS[@]}
do
  echo ${WORD}
done
```

Python

```
arrayOfWords = ["this", "is", "my", "array", "of", "words"]
for word in arrayOfWords:
  print word
```

Shell (bash)

```
#include <iostream>
#include <string>
int main() {
  std::string words[6] = {"this", "is", "my", "array", "of", "words"};
  for  (int i=0; i<sizeof(words)/sizeof(words[0]); ++i) {
    std::cout << words[i] << std::endl;
  }
  return 0;
}
```

C++

Instalação do Python e Jupyter com o Anaconda

- Siga os passos do tutorial: https://minerandodados.com.br/instalar-python-anaconda/
- Lá você encontra os passos para Windows, Linux e MacOS

A linguagem de programação Python

- Python é um exemplo de uma linguagem de programação de alto nível (assim como C, C ++, Perl e Java)

- Linguagens de programação de baixo nível: "linguagens de máquina" ou "linguagens de montagem" (assembly language).

    - De forma não muito rígida, os computadores só podem executar programas escritos em linguagens de baixo nível.
    - Os programas escritos em uma linguagem de alto nível precisam ser processados antes de poderem ser executados.
    - Esse processamento extra leva algum tempo, o que é uma pequena desvantagem das linguagens de alto nível.

- As vantagens de um programa de alto nível:

    - É muito mais fácil de programar.
    - Demora menos tempo para ser escrito, é mais curto e fácil de ler, além de ter maior probabilidade de estar correto.
    - É portátil: pode ser executado em diferentes tipos de computadores com poucas ou nenhuma modificação.
    - Programas de baixo nível podem ser executados em apenas um tipo de computador e precisam ser reescritos para serem executados em outro.
    - Por conta dessas vantagens quase todos os programas são escritos em linguagem de alto nível, a não ser aplicações específicas que necessitam linguagem de baixo nível.

Interpretadores e Compiladores

São dois tipos de programas que processam linguagem de alto nível em linguagem de baixo nível:

Interpretador:

- Lê um programa de alto nível e o executa: faz o que o programa comanda.
- Processa o programa um pouco por vez, lendo linhas e realizando cálculos alternadamente.

Compilador:

- Lê o programa por completo e traduz antes de executar.
- O programa de alto nível é chamado de código-fonte e o programa traduzido é chamado de código-objeto ou executável.
- Depois que um programa é compilado, você pode executá-lo repetidamente sem mais traduções.

Python: uma linguagem interpretada

Interpretador de Python

- modo imediato (prompt)
- modo script

Interpretador de Python Modo Prompt

- No terminal: `$ python`
- Para sair: ctrl + D

```
(python-env) [sheilaamaral@iMacs-iMac ~]$ python
Python 3.9.1 | packaged by conda-forge | (default, Jan 10 2021, 02:52:42)
[Clang 11.0.0 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello world!')
hello world!
```

Python: uma linguagem interpretada

Modo script:

- execute seu programa:

```
python mypython-code.py
```

- ou torne seu programa executável adicionando a seguinte linha no topo do seu programa (altamente dependende de plataforma):
  `#!/usr/bin/env`

```
(python-env) [sheilaamaral@iMacs-iMac ~]$ python mypython-code.py
12
Podemos imprimir a frase: "Hello World!"
```

```
(python-env) [sheilaamaral@iMacs-iMac ~]$ more mypython-code.py
x = 34 - 23                     # comentário

y = "Hello"                     # outro comentário

z = 3.45

if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World!"           # concatenação de string

print (x)
print ('Podemos imprimir a frase: "%s"' % y)  #formatação de string
```

Google Colab

- O Google Colaboratory, carinhosamente chamado de Colab, é um serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python, diretamente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina.
- Para acessar o Google Colab são necessárias apenas duas condições:
  - Um navegador web com acesso à internet;
  - Uma conta no Google.

Hands-on

- Vamos para o repositório https://github.com/Analise-Dados-FAE/2021/tree/main/aula2_git_github_python/python_parte1 onde temos os notebooks da aula de hoje

References

- https://twiki.cern.ch/twiki/bin/viewauth/CMS/PyROOTHATSatLPC2020
- https://www.nevis.columbia.edu/~seligman/root-class/