

Proposta de disciplina do PPGFis
FIP00000 - Astronomia E Astrofísica : Introdução à Linguagem De
Programação Python

- **Semestre:** 2020/2
 - **Carga horária semanal:** 4
 - **Créditos:** 4
 - **Pré-requisitos:**
 - **Professor/Responsável:** José Eduardo da Silveira Costa
-

Súmula

Linguagem de programação Python. Convenções. Code style. Notebooks. Markdown. Tipos básicos de dados. Operadores e operações. Identificadores. Variáveis. Entrada e saída de dados. Strings. Estruturas de controle de fluxo do código. Listas. Tuplas. Conjuntos. Dicionários. Funções. Módulos. NumPy arrays. Gráficos. Pandas. Data frames. Classes. Iteradores. Exceções. Ambientes integrados de programação.

Objetivos

O curso tem como principal objetivo introduzir o estudante à linguagem de programação Python, apresentando seus fundamentos e uma série de recursos complementares para que consiga utilizá-la de modo eficaz em suas atividades de pesquisa.

Programa

1. Apresentação. O que é Python? Por que usar Python como linguagem de programação? Convenções. Adotando um estilo para escrever código.
2. Notebooks, o que são? Combinando texto e código Python para criar documentos vivos. Como usar o Google Colaboratory (Colab). Jupyter Notebook. IPython. "Hello, world!": noções preliminares de Python. Inserindo texto com Markdown.

3. Tipos básicos de dados numéricos: `int`, `float` e `complex`. Operadores binários e unários. Operações aritméticas. Usando o Python como calculadora. Conversões entre os diferentes tipos de dados. Os módulos `math` e `cmath` e suas funções matemáticas.
4. Identificadores e variáveis.
5. Entrada e saída de dados via terminal. Forms.
6. Caracteres e strings. Operações com strings: extração de caracteres, concatenação, fatiamento e replicação. Strings e seus métodos. Formatação.
7. `True` e `False`: dados e operações booleanas. Operadores e expressões lógicas.
8. Controlando o fluxo do código com `if-elif-else`. Os loops `for` e `while`.
9. Listas e Tuplas.
10. Conjuntos e dicionários.
11. Cópias simples, cópias rasas e cópias profundas.
12. Funções. Funções lambda. Funções recursivas. Docstrings.
13. Módulos e bibliotecas.
14. Arrays. A biblioteca NumPy. Expandindo as capacidades numéricas do Python. Vetores e matrizes. Programação vetorial. Funções universais.
15. Entrada e saída de dados via arquivos. Arquivos em formato csv. Arquivos binários. Lendo e salvando grandes volumes de dados.
16. Gráficos. Biblioteca Matplotlib. Gráficos xy simples. Diagramas de dispersão. Gráficos em barras. Histogramas. Diagramas em pizza. Escalas lineares e logarítmicas. Legendas. Coordenadas polares.
17. Pandas. Data frames e seus métodos. Análise de dados básica.
18. Classes. Atributos e métodos. Métodos especiais. Herança. Programação orientada a objeto.
19. Iteradores.
20. Lidando com exceções. As cláusulas `try-except-else-finally`.
21. Linux Python. Distribuições Python. Configurando o Linux para programação em Python.
22. Ambientes integrados de programação (IDEs). Spyder. Programação orientada a projetos.

Método de Trabalho

Em modo de ensino remoto emergencial (ERE), as aulas presenciais são substituídas por **videoaulas assíncronas**, gravadas e disponibilizadas através do **Moodle**. Nas videoaulas, o professor apresenta de forma gradual a linguagem de programação Python por meio de demonstrações práticas. O estudante deverá assistir cada videoaula e reproduzir as demonstrações através do **Colab**, uma ferramenta gratuita, acessível através da internet e que utiliza o **Jupyter Notebook** para programação colaborativa em Python. Ao final de cada aula, o estudante terá elaborado um **notebook** com as demonstrações e anotações pessoais sobre o conteúdo apresentado, o qual deverá ser compartilhado com o professor. Em cada notebook, o estudante poderá, se quiser, adicionar comentários com suas dúvidas, os quais serão lidos e respondidos pelo professor.

Serão propostas **listas de exercícios** as quais deverão ser resolvidas e compartilhadas com o professor através do Colab. Tanto os notebooks, quanto as listas poderão ter prazos para envio.

Serão realizadas **webconferências**, através do **MConf**, para o esclarecimento de dúvidas sobre os conteúdos apresentados, utilizando os recursos colaborativos do Colab, que permitem que professor e aluno visualizem e editem o código em tempo real. As webconferências serão agendadas ao longo do semestre, ocorrendo nos dias/horários da disciplina e os links estarão disponíveis no Moodle. As webconferências não serão gravadas e a participação do estudante é opcional.

Avaliação

A avaliação será feita por meio de:

- **listas de exercícios**, valendo até 10 (dez) pontos cada uma. A média das notas nas listas tem peso 5 (cinco) na nota final;
- **notebooks**, referentes às videoaulas, valendo até 10 (dez) pontos cada um. A média das notas nos notebooks tem peso 3 (três) na nota final.
- **prova final**, valendo até 10 (dez) pontos e tendo peso 2 (dois) na nota final. Na prova final, serão propostos um ou mais problemas, para os quais o estudante deverá desenvolver códigos em Python, com base nos conhecimentos apresentados ao longo do curso.

A nota final do curso (N) será a média ponderada das três notas:

$$N = 0.5 \times \text{média_nas_listas} + 0.3 \times \text{média_nos_notebooks} + 0.2 \times \text{nota_na_prova}$$

O conceito final será atribuído conforme os critérios:

Conceito D: $N < 6.0$

Conceito C: $6.0 \leq N \leq 7.4$

Conceito B: $7.5 \leq N \leq 8.9$

Conceito A: $9.0 \leq N \leq 10$

Recuperação: caso o estudante tenha obtido conceito D, poderá realizar uma prova de recuperação, valendo até 10 (dez) pontos e tendo peso 2 (dois) no recálculo da nota final do curso.

Bibliografia

- Menezes, Nilo Ney Coutinho. *Introdução à programação com Python : algoritmos e lógica de programação para iniciantes*. São Paulo: Novatec, 2014. ISBN 9788575224083.
- Lutz, Mark. *Aprendendo Python*. Porto Alegre: Bookman, 2007. ISBN 9788577800131.
- Bernard, Joey. *Python Recipes Handbook: A Problem-Solution Approach*. Berkeley, CA: Apress, 2016. ISBN 9781484202418.
- Gerrard, Paul. *A Primer on Scientific Programming with Python*. Berkeley, CA: Apress, 2016. ISBN 9781484223857.
- Gerrard, Paul. *Learn Python*. Berkeley, CA: Apress, 2016. ISBN 9781484223857.
- Kalb, Irv. *Learn to Program with Python*. Berkeley, CA: Apress, 2016. ISBN 9781484221723.
- Langtangen, Hans Petter. *A Primer on Scientific Programming with Python*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. ISBN 9783662498873.
- Lee, Kent D. *Data Structures and Algorithms with Python*. Cham: Springer International Publishing, 2015. ISBN 9783319130729.
- LinkLee, Kent D. *Python Programming Fundamentals*. London: Springer London, 2014. ISBN 9781447166429.
- Padmanabhan, T R. *Programming with Python*. Singapore: Springer Singapore, 2016. ISBN 9789811032776.
- Shaw, Zed A. *Learn Python the hard way : a very simple introduction to the terrifyingly beautiful world of computers and code*. Upper Saddle River: Addison-Wesley, 2014. ISBN 9780321884916.
- Zhang, Yue. *An Introduction to Python and Computing Programming*. Singapore: Springer Singapore, 2015. ISBN 9789812876096.