

**Proposta de disciplina do PPGFis**  
**FIP10108 - Astronomia E Astrofísica: Introdução à Linguagem De**  
**Programação Python**

---

- **Semestre:** 2021/1
  - **Carga horária semanal:** 4
  - **Créditos:** 4
  - **Pré-requisitos:**
  - **Professor/Responsável:** José Eduardo da Silveira Costa
- 

## **Súmula**

Linguagem de programação Python. Notebooks. Markdown. Tipos básicos de dados numéricos. Operadores e operações. Identificadores. Variáveis. Entrada e saída de dados. Strings. Listas. Tuplas. Conjuntos. Dicionários. Estruturas de controle de fluxo. Funções. Módulos. NumPy arrays. Gráficos. Classes. Iteradores. Decoradores. Exceções. Ambientes Integrados de Desenvolvimento.

## **Objetivos**

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

## **Programa**

1. Apresentação. O que é Python?
2. Notebooks. Jupyter notebook. Google Colaboratory (Colab). Markdown.
3. Tipos básicos de dados: int, float, complex, str e bool. Conversões.
4. Operadores e operações. Funções matemáticas.
5. Identificadores, objetos e variáveis.

6. Entrada e saída de dados via terminal. A função print. Forms.
7. Caracteres e strings. Operações com strings. Strings e seus métodos. Formatação.
8. True e False: dados e operações booleanas. Expressões e operações lógicas.
9. Estruturas de controle: if-elif-else. Loops: for e while.
10. Listas e tuplas.
11. Conjuntos e dicionários.
12. Cópias simples, rasas e profundas.
13. Módulos e bibliotecas.
14. Funções. Funções lambda. Funções recursivas. Docstrings. Programação Modular.
15. A biblioteca NumPy. Arrays, vetores e matrizes. Funções universais. Programação Vetorial.
16. Entrada e saída de dados via arquivos. Arquivos em formato CSV. Arquivos binários. Lendo e salvando grandes volumes de dados.
17. Gráficos. Biblioteca matplotlib. Gráficos simples. Diagramas de dispersão. Gráficos em barras. Histogramas. Diagramas em pizza. Escalas lineares e logarítmicas. Grades. Legendas. Coordenadas polares.
18. Exceções. As cláusulas try-except-else-finally.
19. Classes. Membros de uma classe: atributos e métodos. Métodos especiais. Herança. Programação orientada à objetos vs. programação estruturada.
20. Iteradores.
21. Decoradores.
22. Ambientes integrados de programação (IDE). Spyder.

## **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 apresentará, de forma gradual, a linguagem de programação Python por meio de demonstrações práticas. O estudante deverá assistir cada videoaula e reproduzir as demonstrações, elaborando um notebook, o qual deverá ser compartilhado com o professor.

Em cada notebook, o estudante poderá, se quiser, adicionar comentários ou dúvidas que serão respondidas pelo professor.

Serão propostas listas de exercícios que deverão ser resolvidas e compartilhadas com o professor. Tanto os notebooks, quanto as listas poderão ter prazos para envio. Serão realizadas webconferências através do MConf para esclarecimento de dúvidas sobre os conteúdos apresentados, em datas agendadas conforme demanda. As webconferências não serão gravadas e a participação 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 terá peso 5 (cinco) na nota final;
- **notebooks**, referentes às videaulas, valendo até 10 (dez) pontos cada um. A média das notas nos notebooks terá 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:

$$N = 0.4 \times \text{média\_nas\_listas} + 0.4 \times \text{média\_nos\_notebooks} + 0.2 \times \text{nota\_na\_prova}$$

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. ISBN9788575224083.
- 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. ISBN9781484221723.
- 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.
- Link Lee, 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.