

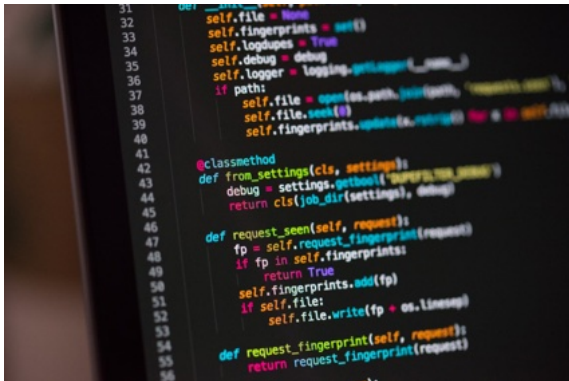
Fundamentos de Programação em Python

Visão Geral do Curso

Prof. Dr. Denis M. L. Martins

Ciência de Dados e Inteligência Artificial: 1º Semestre

Image credits: Chris Ried @Unsplash



Estudar e entender conceitos fundamentais de programação em Python, desenvolvendo lógica computacional e habilidades para criar algoritmos, manipular estruturas de dados e resolver problemas computacionais.

Why Python?

- **Popularidade Global:** Python é uma das linguagens de programação mais populares do mundo.
- **Versatilidade:** Pode ser aplicado em diversas áreas, incluindo: Ciência de Dados e Inteligência Artificial, Desenvolvimento Web e Automação.
- **Facilidade de Aprendizado:** Sintaxe simples e intuitiva, ideal para iniciantes.

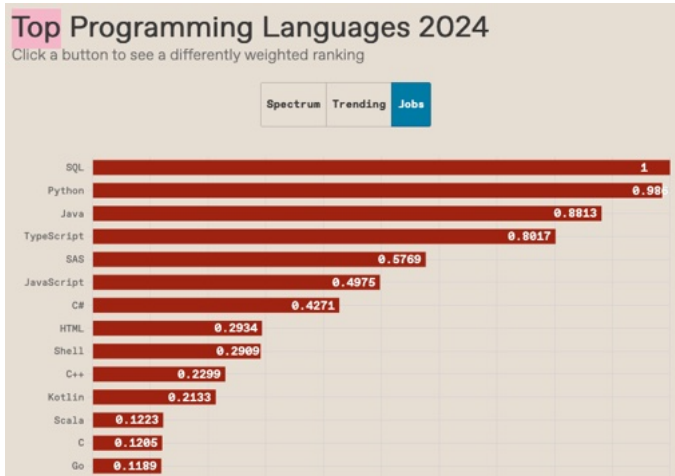


Figura 1: Top linguagens de programação segundo o IEEE Spectrum.

● Módulo 1

- ▶ Noções de algoritmo e resolução de problemas computacionais.
- ▶ Variáveis, tipos de dados e operadores.
- ▶ Estruturas de controle, condicional e de repetição.
- ▶ Funções de predefinidas e de usuário.

● Módulo 2

- ▶ Estruturas de dados: Listas e Dicionários.
- ▶ Sistemas Distribuídos.
- ▶ Módulos e bibliotecas.
- ▶ Manipulação de arquivos.
- ▶ Plots e gráficos.

Créditos: [emilwidlund @Unsplash](#)



● Módulo 1

- ▶ Noções de algoritmo e resolução de problemas computacionais.
- ▶ Variáveis, tipos de dados e operadores.
- ▶ Estruturas de controle, condicional e de repetição.
- ▶ Funções de predefinidas e de usuário.

● Módulo 2

- ▶ Estruturas de dados: Listas e Dicionários.
- ▶ Sistemas Distribuídos.
- ▶ Módulos e bibliotecas.
- ▶ Manipulação de arquivos.
- ▶ Plots e gráficos.

Créditos: [emilwidlund @Unsplash](#)



Usaremos **Jupyter Notebooks** neste curso.

- **Ambiente Interativo:** Permite a execução de código em células individuais.
- **Integração de Conteúdo:** Código, texto explicativo, fórmulas matemáticas e visualizações em um único documento.
- **Reprodutibilidade:** Facilita a reprodução de análises e experimentos.
- **Ferramenta Didática:** Amplamente utilizado em ambientes educacionais.



Jupyter header and tool bar.

A markdown cell, with title, explanation, and equation.

A code cell, setting things up with needed libraries.

A short explanation.

Code cells assigning two array variables, then making a line plot.

Figura 2: Jupyter Notebook. Créditos: Barba, Lorena A., et al. Teaching and learning with Jupyter. 2019.

Organização do Curso

Comunicação

- Toda a **comunicação** será centralizada no **Canvas**.
- Todo o material será disponibilizado lá.
- Sempre verifiquem os avisos.
- Usem o fórum de dúvidas.
- Submissão de tarefas.
- Enviem, se necessário, e-mail para:
denis.mayr@puc-campinas.edu.br



Organização do Curso

Comunicação

- Toda a **comunicação** será centralizada no **Canvas**.
- Todo o material será disponibilizado lá.
- Sempre verifiquem os avisos.
- Usem o fórum de dúvidas.
- Submissão de tarefas.
- Enviem, se necessário, e-mail para:
denis.mayr@puc-campinas.edu.br



- André Luiz Villar Forbellone e Henri Frederico Eberspächer. Lógica de programação: A construção de algoritmos e estruturas de dados com aplicações em Python. 4a edição. Editora Pearson, 2022.
- Jake VanderPlas. Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc., 2016. <https://jakevdp.github.io/PythonDataScienceHandbook/>
- Allen Downey, Jeff Elkner e Chris Meyers. Aprendendo computação com Python: <https://aprendendo-computacao-com-python.readthedocs.io/en/latest/>

Organização do Curso

Avaliação

- **Provas** (80% da nota)

- ▶ Prova 1 (P1)

- ★ Data: 09 de Abril.

- ★ Aula de dúvidas (e revisão): 02 de Abril.

- ▶ Prova 2 (P2)

- ★ Data: 11 de Junho.

- ★ Aula de dúvidas (e revisão): 04 de Junho.

- **Projeto** (20% da nota)

- ▶ Data: 14 de Maio.

- ▶ Organizado em grupos.

- ▶ Prática de implementação.

- ▶ Critérios de avaliação divulgados em Abril (após a prova P1).

- **Cálculo da Nota:** $0.8 \times (P1+P2)/2 + 0.2 \times Proj$

- **Recuperação**

- ▶ Exame sobre todo o conteúdo em 18 de Junho.

- ▶ Requisito: não ter extrapolado o limite de faltas.



Créditos: yustinustjiu @Unsplash

Organização do Curso

Avaliação

- **Provas** (80% da nota)

- ▶ Prova 1 (P1)

- ★ Data: 09 de Abril.

- ★ Aula de dúvidas (e revisão): 02 de Abril.

- ▶ Prova 2 (P2)

- ★ Data: 11 de Junho.

- ★ Aula de dúvidas (e revisão): 04 de Junho.

- **Projeto** (20% da nota)

- ▶ Data: 14 de Maio.

- ▶ Organizado em grupos.

- ▶ Prática de implementação.

- ▶ Critérios de avaliação divulgados em Abril (após a prova P1).

- **Cálculo da Nota:** $0.8 \times (P1+P2)/2 + 0.2 \times Proj$

- **Recuperação**

- ▶ Exame sobre todo o conteúdo em 18 de Junho.

- ▶ Requisito: não ter extrapolado o limite de faltas.



Créditos: yustinustjiu @Unsplash

Organização do Curso

Avaliação

- **Provas** (80% da nota)

- ▶ Prova 1 (P1)

- ★ Data: 09 de Abril.

- ★ Aula de dúvidas (e revisão): 02 de Abril.

- ▶ Prova 2 (P2)

- ★ Data: 11 de Junho.

- ★ Aula de dúvidas (e revisão): 04 de Junho.

- **Projeto** (20% da nota)

- ▶ Data: 14 de Maio.

- ▶ Organizado em grupos.

- ▶ Prática de implementação.

- ▶ Critérios de avaliação divulgados em Abril (após a prova P1).

- **Cálculo da Nota:** $0.8 \times (P1+P2)/2 + 0.2 \times Proj$

- **Recuperação**

- ▶ Exame sobre todo o conteúdo em 18 de Junho.

- ▶ Requisito: não ter extrapolado o limite de faltas.



Créditos: yustinusjiu @Unsplash

Organização do Curso

Avaliação

- **Provas** (80% da nota)

- ▶ Prova 1 (P1)
 - ★ Data: 09 de Abril.
 - ★ Aula de dúvidas (e revisão): 02 de Abril.
- ▶ Prova 2 (P2)
 - ★ Data: 11 de Junho.
 - ★ Aula de dúvidas (e revisão): 04 de Junho.

- **Projeto** (20% da nota)

- ▶ Data: 14 de Maio.
- ▶ Organizado em grupos.
- ▶ Prática de implementação.
- ▶ Critérios de avaliação divulgados em Abril (após a prova P1).

- **Cálculo da Nota:** $0.8 \times (P1+P2)/2 + 0.2 \times Proj$

- **Recuperação**

- ▶ Exame sobre todo o conteúdo em 18 de Junho.
- ▶ Requisito: não ter extrapolado o limite de faltas.



Créditos: yustinustjiu @Unsplash

Formato "Fail"¹:

- Vocês tentam resolver as questões antes da explicação teórica, o que permite diagnosticar sua compreensão inicial.
- Após a correção, os erros servem como um guia para o aprendizado, reforçando conceitos que precisam ser mais trabalhados.
- A abordagem melhora a retenção do conhecimento e incentiva um aprendizado mais ativo.

¹Conferir: Kapur, Manu, et al. "Fail, flip, fix, and feed—Rethinking flipped learning: A review of meta-analyses and a subsequent meta-analysis."Frontiers in Education. Vol. 7. Frontiers Media SA, 2022.

Questão 1: O que é um algoritmo?

- (a) Um conjunto de instruções organizadas para resolver um problema.
- (b) Um tipo especial de variável em Python.
- (c) Um erro na execução do código.
- (d) Um compilador de linguagem de programação.

Questão 1: O que é um algoritmo?

- (a) Um conjunto de instruções organizadas para resolver um problema.
- (b) Um tipo especial de variável em Python.
- (c) Um erro na execução do código.
- (d) Um compilador de linguagem de programação.

Resposta: (a)

Questão 2: O que é uma variável em programação e qual é sua função?

- Um valor fixo que não pode ser alterado.
- Um espaço de armazenamento na memória do computador que pode conter diferentes valores ao longo do tempo.
- Um comando que exibe mensagens na tela.
- Um tipo especial de dado que armazena apenas números.

Questão 2: O que é uma variável em programação e qual é sua função?

- Um valor fixo que não pode ser alterado.
- Um espaço de armazenamento na memória do computador que pode conter diferentes valores ao longo do tempo.
- Um comando que exibe mensagens na tela.
- Um tipo especial de dado que armazena apenas números.

Resposta: (b)

Questão 3: Qual das opções abaixo representa corretamente uma variável em Python?

- `int x = 10;`
- `define x 10`
- `x := 10`
- `x = 10`

Questão 3: Qual das opções abaixo representa corretamente uma variável em Python?

- `int x = 10;`
- `define x 10`
- `x := 10`
- `x = 10`

Resposta: (d)

Questão 4: Em Python, qual operador é usado para comparar se dois valores são iguais

- ☐ =
- ☐ ===
- ☐ ==
- ☐ !=

Questão 4: Em Python, qual operador é usado para comparar se dois valores são iguais

- ☐ =
- ☐ ==
- ☐ ==
- ☐ !=

Resposta: (c)

Dúvidas e Discussão

Prof. Dr. Denis M. L. Martins

denis.mayr@puc-campinas.edu.br