# Educação Profissional Paulista

Técnico em

Desenvolvimento

de Sistemas



# Estruturas de repetição

Laço de repetição: FOR

Aula 2

Código da aula: [SIS]ANO1C1B2S14A2





#### Objetivo da Aula

• Saber como o laço *FOR* é utilizado para manipular coleções de dados, como arrays e listas, sendo essencial para operações de dados em muitas linguagens de programação.



#### Competências da Unidade (Técnicas e Socioemocionais)

- Desenvolver sistemas computacionais, utilizando ambiente de desenvolvimento;
- Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes;
- Trabalhar a criatividade e o comprometimento na resolução de problemas computacionais.



#### **Recursos Didáticos**

- Recurso audiovisual para a exibição de vídeos e imagens;
- Folhas sulfite, canetas coloridas, lápis.



#### Duração da Aula

50 minutos

## Laço FOR e coleções de dados

O laço FOR é utilizado para manipular coleções de dados, como arrays e listas, sendo essencial para operações de dados em muitas linguagens de programação.



Iteração em arrays e listas.



Manipulação de dados complexos.



Uso eficiente da memória e tempo de execução.





# Agora, vamos compreender o que são as listas:



ALURA. *Python*: crie a sua primeira aplicação. 03 Listas. Disponível em:

Acesso em: 28 mar. 2024.





© Getty Images

# Iteração em arrays e listas

Arrays e listas são estruturas de dados fundamentais em Python. Arrays geralmente são fornecidos por bibliotecas como NumPy e são otimizados para operações matemáticas; já as listas são parte integrante do Python e são mais flexíveis.

#### Exemplo básico:

numeros = [1, 2, 3, 4, 5] for numero in numeros: print(numero)

#### Uso no mercado

- Análise de dados: em data science, é comum iterar sobre listas ou arrays para processar ou analisar dados. Por exemplo, você pode ter uma lista de preços de ações e querer calcular a média.
- **Automação de processos:** em automação, você pode iterar sobre listas de tarefas ou e-mails para processá-los automaticamente.

## Manipulação de dados complexos

Manipulação de dados complexos envolve lidar com estruturas de dados como dicionários, listas de dicionários, JSON etc.

#### **Exemplo:**

```
dados = [{"nome": "Ana", "idade": 25}, {"nome": "Beto", "idade": 30}]
for item in dados:
    print(f"{item['nome']} tem {item['idade']} anos")
```

#### Uso no mercado

- **Desenvolvimento web e APIs:** manipulação de JSON para comunicação entre o servidor e o cliente.
- Ciência de dados: processamento de dados complexos para análise estatística e machine learning.



# Uso eficiente da memória e tempo de execução

Python oferece várias estratégias para otimizar o uso da memória e tempo de execução, como compreensão de listas, geradores e bibliotecas eficientes, como NumPy e Pandas.

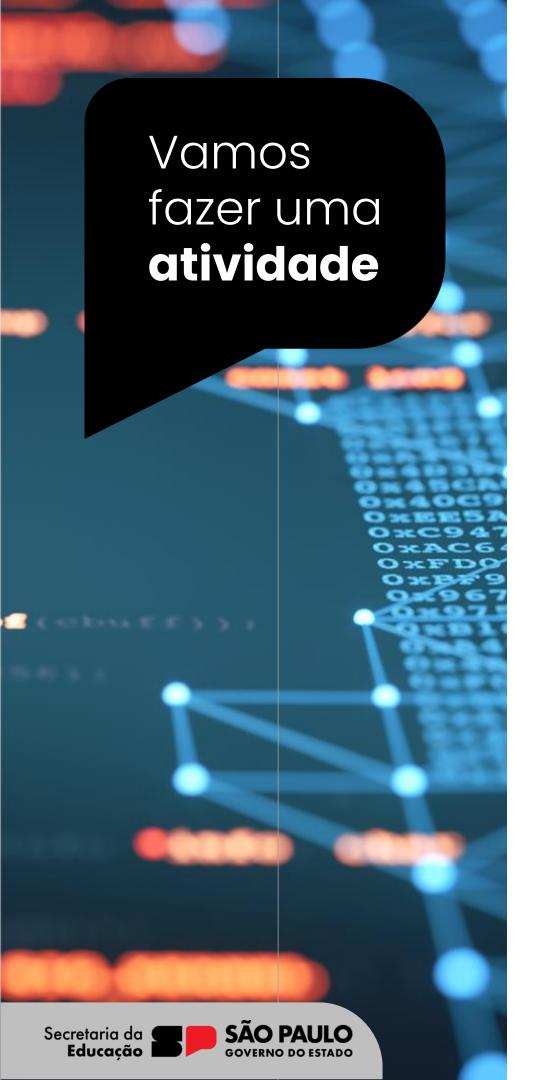
#### Exemplo com compreensão de lista:

numeros = range(1000000) quadrados = [x\*x for x in numeros]

#### Uso no mercado

- **Big data:** tratamento de grandes volumes de dados, em que a eficiência de memória e processamento é crucial.
- **Desenvolvimento de software:** otimização de algoritmos para melhorar a performance em aplicações.







### Análise de dados de vendas

**Objetivos** – Analisar e compreender um script Python pré-fornecido que realiza a análise de dados de vendas de uma loja online. O foco é identificar os componentes-chave do script e criar um fluxograma detalhado que represente o fluxo lógico das operações realizadas pelo script.

**Enunciado da atividade –** Você receberá um script Python que executa a seguinte análise em um conjunto de dados de vendas:

- Identifica os cinco produtos mais vendidos;
- Determina os meses com maior volume de vendas;
- Sua tarefa será estudar o script e criar um fluxograma que descreva cada passo do processo de análise. O fluxograma deve ilustrar claramente como o script processa os dados, incluindo a leitura dos dados, as operações de contagem e agregação e a geração dos resultados finais.



#### Análise de dados de vendas





Em grupos de até 6 pessoas

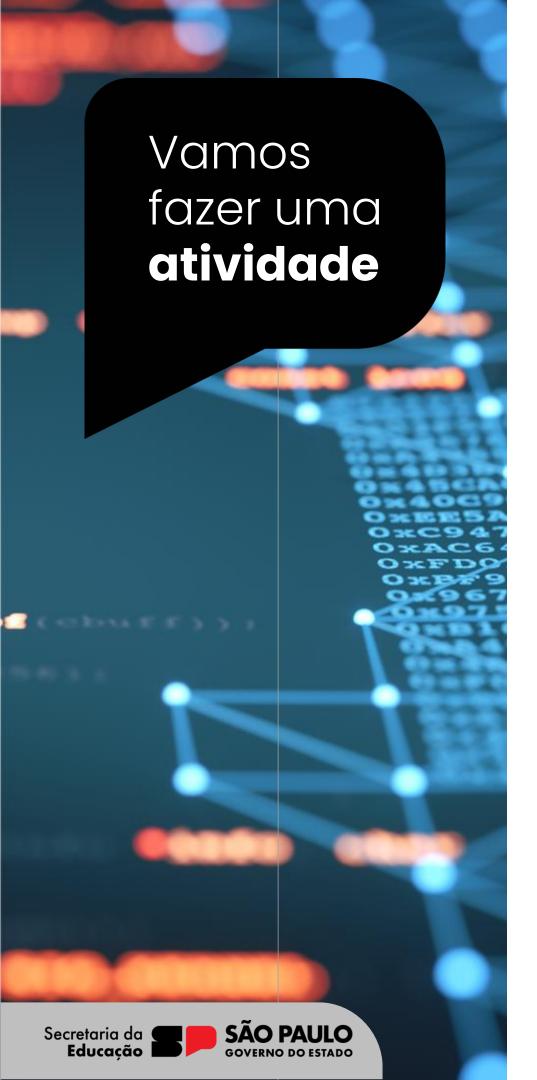




#### Análise de dados de vendas

- Estudo do script: leia e compreenda o script Python fornecido.
  Identificação de componentes-chave: identifique as principais funções e operações no script, como loops, condicionais, agregações e saídas.
- Criação do fluxograma: desenhe um fluxograma que represente visualmente o fluxo de operações do script.
  O fluxograma deve incluir:
  - Importação de módulos e definição de dados;
  - Processo de iteração e agregação de dados;
  - Cálculo dos produtos mais vendidos e análise das vendas mensais;
  - Exibição dos resultados.
- Análise crítica: reflita sobre como o script poderia ser otimizado ou modificado para atender a diferentes requisitos de análise de dados.
  - Lembre-se de enviar o relatório pelo AVA.



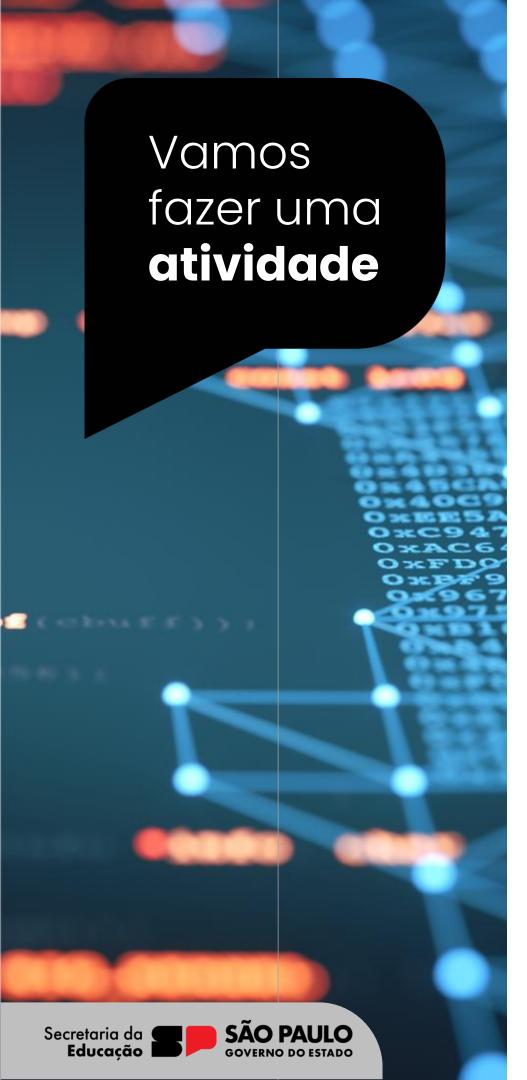


# Script base



from collections import Counter from datetime import datetime

```
# Dados de exemplo
vendas = [
  {"produto": "Teclado", "quantidade": 10, "data": "2021-03-15"},
  {"produto": "Mouse", "quantidade": 15, "data": "2021-03-17"},
  {"produto": "Monitor", "quantidade": 7, "data": "2021-04-01"},
  # ... outros dados ...
# Analisando os produtos mais vendidos
contador_produtos = Counter()
for venda in vendas:
  contador_produtos[venda["produto"]] += venda["quantidade"]
cinco_mais_vendidos = contador_produtos.most_common(5)
# Analisando vendas por mês
vendas_por_mes = Counter()
for venda in vendas:
  mes = datetime.strptime(venda["data"], "%Y-%m-%d").strftime("%Y-%m")
  vendas_por_mes[mes] += venda["quantidade"]
meses_maior_venda = vendas_por_mes.most_common()
# Resultados
print("Cinco produtos mais vendidos:", cinco_mais_vendidos)
print("Vendas por mês:", meses_maior_venda)
```



# Exemplo de resolução

Registro

Exemplo de resolução da atividade: criação do fluxograma

- 1. Início do processo Início do fluxograma.
- **2. Preparação do script -** Importação de módulos: collections. Counter e datetime. Definição dos dados de vendas: uma lista de dicionários com informações sobre as vendas.
- **3. Processamento dos dados** Inicialização dos contadores: dois contadores são criados, um para produtos (contador\_produtos) e outro para vendas por mês (vendas\_por\_mes).

Iteração sobre os dados de vendas - Para cada venda: atualização do contador de produtos (soma a quantidade vendida do produto).

Conversão da data de venda para mês e atualização do contador de vendas por mês: soma a quantidade vendida no respectivo mês.

- **4. Análise e agrupamento dos dados -** Determinação dos cinco produtos mais vendidos: usa o método most\_common(5) no contador\_produtos. Determinação dos meses com maior volume de vendas: usa o método most\_common() no vendas\_por\_mes.
- **5. Exibição dos resultados -** Exibir os cinco produtos mais vendidos. Exibir os meses com maior volume de vendas.
- **6. Fim do processo -** Fim do fluxograma.



# Hoje desenvolvemos:

A compreensão do uso da iteração em arrays e listas.

2 O entendimento sobre contextos que utilizam a manipulação de dados complexos.

3 A prática de um estudo de caso, criando os fluxos necessários para um script que apresenta o contexto do laço de repetição *FOR*.





Avançando um pouco mais em estruturas de repetição, **que tal conhecer também detalhes sobre o laço** *While*?

SHARPAX. Aula 11 – Estrutura de repetição ENQUANTO (WHILE) I Lógica de programação. Disponível em:

https://www.youtube.com/watch?v=tfAh284004w. Acesso em: 28 mar. 2024.



#### Referências da aula

ALURA. *Python*: crie a sua primeira aplicação. 03 Listas. Disponível em: <a href="https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao/task/146275">https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao/task/146275</a>. Acesso em: 28 mar. 2024.

SHARPAX. Aula 11 – Estrutura de repetição ENQUANTO (WHILE) I Lógica de programação. Disponível em: <a href="https://www.youtube.com/watch?v=tfAh284004w">https://www.youtube.com/watch?v=tfAh284004w</a>. Acesso em: 28 mar. 2024.

ZANELALO, J. *Lógica de programação* – Estruturas de repetição. PodProgramar, 25 fev. 2018. Disponível em: <a href="https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/">https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/</a>. Acesso em: 28 mar. 2024.

Identidade visual: imagens © Getty Images.



# Educação Profissional Paulista

Técnico em

Desenvolvimento

de Sistemas

