

Departamento de Engenharia de Sistemas e Computação

Arquivos

Leitura e escrita de arquivos

João Araujo

Algoritmos Computacionais I

Aula 23

Roteiro

- Por que arquivos? Memória volátil *versus* armazenamento permanente
- Abrindo, lendo e fechando (Cap. 6.4)
- Escrevendo em arquivos (6.5)
- O comando `with` e tratamento de erros
- Arquivos CSV: dados em tabela

Programas e dados persistentes

- As variáveis vivem na **RAM**: desaparecem quando o programa termina
- Para guardar resultados, ler medições ou trocar dados com outros programas usamos **arquivos**
- Em Python, um arquivo de texto é visto como uma *sequência de linhas* (strings)

Exemplos

- resultados de um ensaio
- log de um sensor
- lista de alunos
- planilhas exportadas em CSV

Abrindo e lendo: `open`

Arquivo `nomes.txt`:

```
João Araujo Ribeiro
Grace Hopper
Ada Lovelace
```

```python
arquivo = open('nomes.txt', 'r')      # 'r' = read (leitura)
linhas = arquivo.readlines()          # lista de strings
arquivo.close()                       # libera o arquivo
for linha in linhas:
    print(linha)
```

Por que as linhas saem com uma linha em branco a mais?

Cada linha lida termina com `\n`, e `print` acrescenta outro. Resolva com `rstrip()`.

Fonte: `code/Cap6/prog.6.12_abrearquivo.py`

Formas de ler

```
# 1) linha por linha (não carrega o arquivo todo)
arquivo = open('nomes.txt', 'r')
for linha in arquivo:
    print(linha.rstrip())
arquivo.close()

# 2) tudo como uma lista, já sem o \n
arquivo = open('nomes.txt', 'r')
linhas = [linha.rstrip() for linha in arquivo.readlines()]
arquivo.close()

# 3) tudo em uma única string
texto = open('nomes.txt').read()
linhas = texto.split('\n')
```

Fonte: code/Cap6/prog.6.13, prog.6.14, prog.6.15 (abre arquivo)

Escrevendo: modos 'w' e 'a'

```
import math

arquivo = open('tabelaSenoCosseno.txt', 'w')     # 'w' = write
arquivo.write(f"{'Graus':>9}{'Radianos':>9}{'Seno':>9}{'Cosseno':>9}\n")
for graus in range(0, 361, 30):
    rad = math.radians(graus)
    arquivo.write(f"{graus:>9.2f}{rad:>9.2f}"
                  f"{math.sin(rad):>9.2f}{math.cos(rad):>9.2f}\n")
arquivo.close()
```

Modo	Efeito
'r'	leitura (padrão); erro se o arquivo não existir
'w'	escrita; apaga o conteúdo anterior!
'a'	escrita no final (*append*), mantendo o conteúdo

Fonte: code/Cap6/prog.6.17_tabela_senos_arquivo_f.py

`write` não acrescenta `\n`

```
arq = open('saida.txt', 'w')
arq.write('linha 1')
arq.write('linha 2')         # saiu "linha 1linha 2" !
arq.write('linha 3\n')       # \n explícito
print('linha 4', file=arq)   # print grava com \n automático
arq.close()
```

- `write` só aceita strings: converta números com `str()` ou f-string
- Esquecer de fechar (`close`) pode deixar dados sem gravar

O comando `with`

```
with open('nomes.txt', 'r') as arquivo:
    for linha in arquivo:
        print(linha.rstrip())
# aqui o arquivo já foi fechado automaticamente

with open('copia.txt', 'w') as saida:
    with open('nomes.txt', 'r') as entrada:
        for linha in entrada:
            saida.write(linha)
```

- `with` fecha o arquivo sempre, mesmo que ocorra um erro: é a forma recomendada
- O segundo exemplo é o Exercício 6.14 (cópia de arquivo)

Arquivos que não existem

```
nome = input('Nome do arquivo: ')
try:
    with open(nome, 'r') as f:
        conteudo = f.read()
except FileNotFoundError:
    print('Arquivo não encontrado:', nome)
except PermissionError:
    print('Sem permissão para ler', nome)
else:
    print(len(conteudo), 'caracteres lidos')
```

- Combina o que vimos na Aula 13 (`try`/`except`) com arquivos
- O caminho é relativo à pasta onde o programa *executa* (não à do arquivo `.py`)

Exemplo: processando um arquivo de texto

Converter o conteúdo de um arquivo para maiúsculas e gravar o resultado (Exercício 6.10):

```
with open('nomes.txt', 'r') as entrada:
    texto = entrada.read()
with open('nomes_maiusculas.txt', 'w') as saida:
    saida.write(texto.upper())
```

Variações (Exercícios 6.11 e 6.12):

- `texto.lower()` para minúsculas
- Primeira letra de cada palavra maiúscula: `texto.title()` ou processar `split()` palavra a palavra

Dados em tabela: arquivos CSV

CSV = *Comma-Separated Values*: uma linha por registro, campos separados por vírgula.

```
id_peca,maquina,operador,turno,tempo_ciclo_seg,status
1000,M1,Ana,Noite,31.6,Aprovado
1001,M1,Elis,Tarde,34.5,Aprovado
1002,M2,Bruno,Noite,24.9,Descarte
```

```python
import csv
with open('dadosproducao.csv', newline='', encoding='utf-8') as f:
    leitor = csv.DictReader(f)          # cada linha vira um dicionário
    registros = list(leitor)
print(registros[0]['maquina'], float(registros[0]['tempo_ciclo_seg']))
```

Dica: `DictReader` + o que vimos de dicionários (Aula 20) resolve muitos problemas. O módulo `csv` cuida de aspas e vírgulas dentro dos campos.

Fonte: `code/Projeto1/ler_producao_dict.py`

Gravando CSV

```
import csv
ensaios = [(5.0, 0.25), (10.0, 0.52), (15.0, 0.76)]
with open('ensaios.csv', 'w', newline='', encoding='utf-8') as f:
    escritor = csv.writer(f)
    escritor.writerow(['tensao_V', 'corrente_A'])
    escritor.writerows(ensaios)
```

- Todo programa de planilha (Excel, LibreOffice) abre CSV
- Na parte final do curso, o **Pandas** lê CSV em uma única linha: `pd.read_csv('ensaios.csv')`

Atividades

1. Escreva um programa que leia `nomes.txt` e grave em outro arquivo os nomes em ordem alfabética, um por linha.
2. Leia um arquivo de notas (uma por linha) e escreva a média, a maior e a menor em um arquivo `resumo.txt`.
3. Grave em `tabela.txt` a tabela Celsius $\rightarrow$ Fahrenheit de 0 a 100 de 10 em 10 (Exercício 6.13).
4. Leia `dadosproducao.csv` e conte quantas peças cada máquina produziu (dicionário de contagens, Aula 20).

Para praticar: exercícios do livro

Site do livro – Capítulo 6

- Leitura: **6.10, 6.11, 6.12** (conversão de maiúsculas/minúsculas)
- Escrita: **6.13** (tabela em arquivo), **6.14** (cópia de arquivo), **6.15** (pontos de π em arquivo)

Enunciados e soluções comentadas em
`https://www.araujo.eng.uerj.br/livro/exercicios/cap6/`

Exercícios adicionais (sem solução): `https://www.araujo.eng.uerj.br/livro/extras/`

Resumo da aula

- `open(nome, modo)`: `'r'`, `'w'` (apaga!) e `'a'`
- Arquivo de texto é percorrido linha a linha; `rstrip()` tira o `\n`
- `with` fecha o arquivo automaticamente; trate `FileNotFoundError`
- CSV: use o módulo `csv` (ou Pandas, nas aulas finais)
- Próxima aula: revisão, estilo e boas práticas