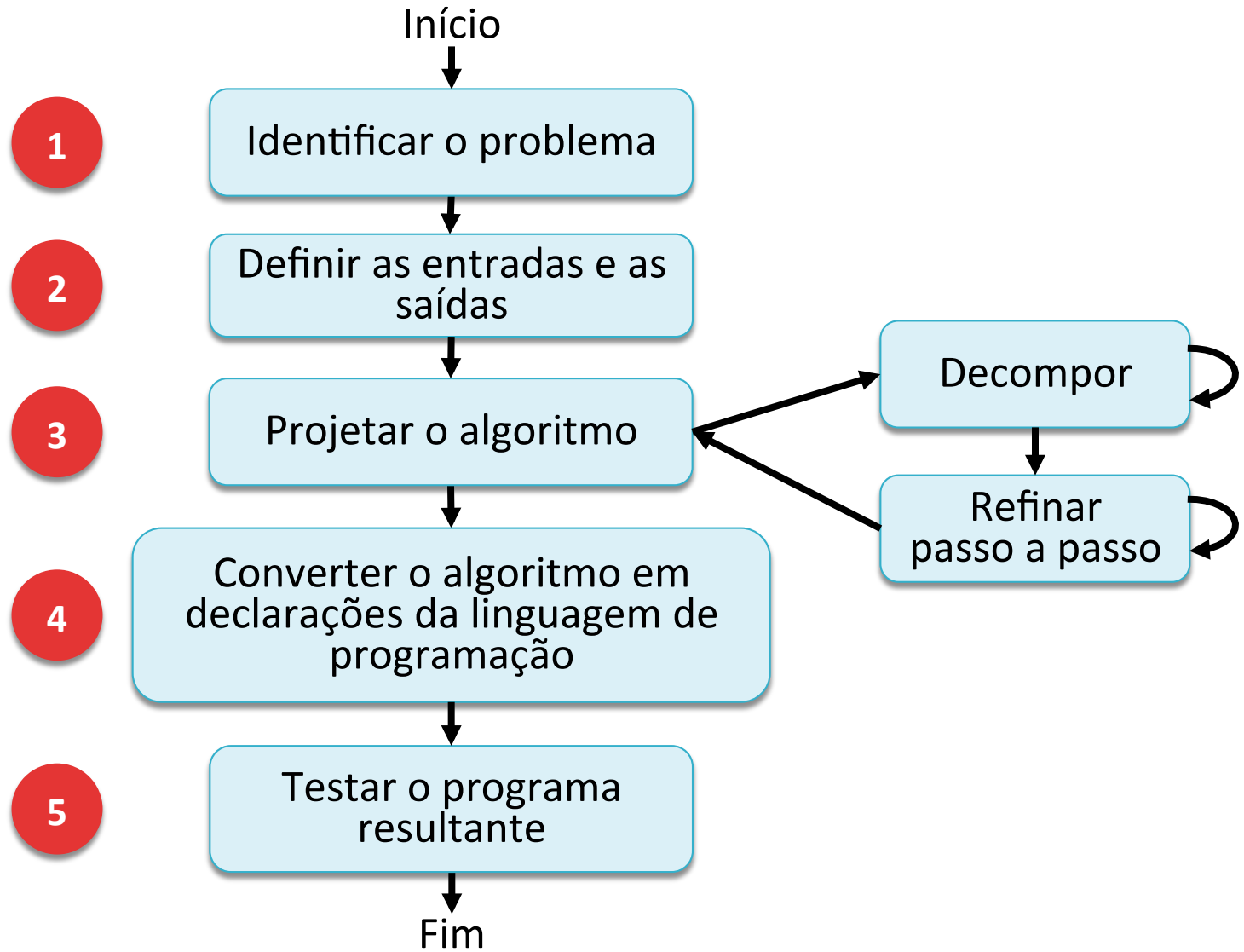


**IEC037**

**Introdução à Programação de  
Computadores**

**Aula 14 – Arquivos em Python**

# Resolução de Problemas Algorítmicos



# Conteúdo



Introdução a arquivos



Operações básicas



Manipulando números

# Conteúdo



Introdução a arquivos



Operações básicas



Manipulando números

# Arquivos

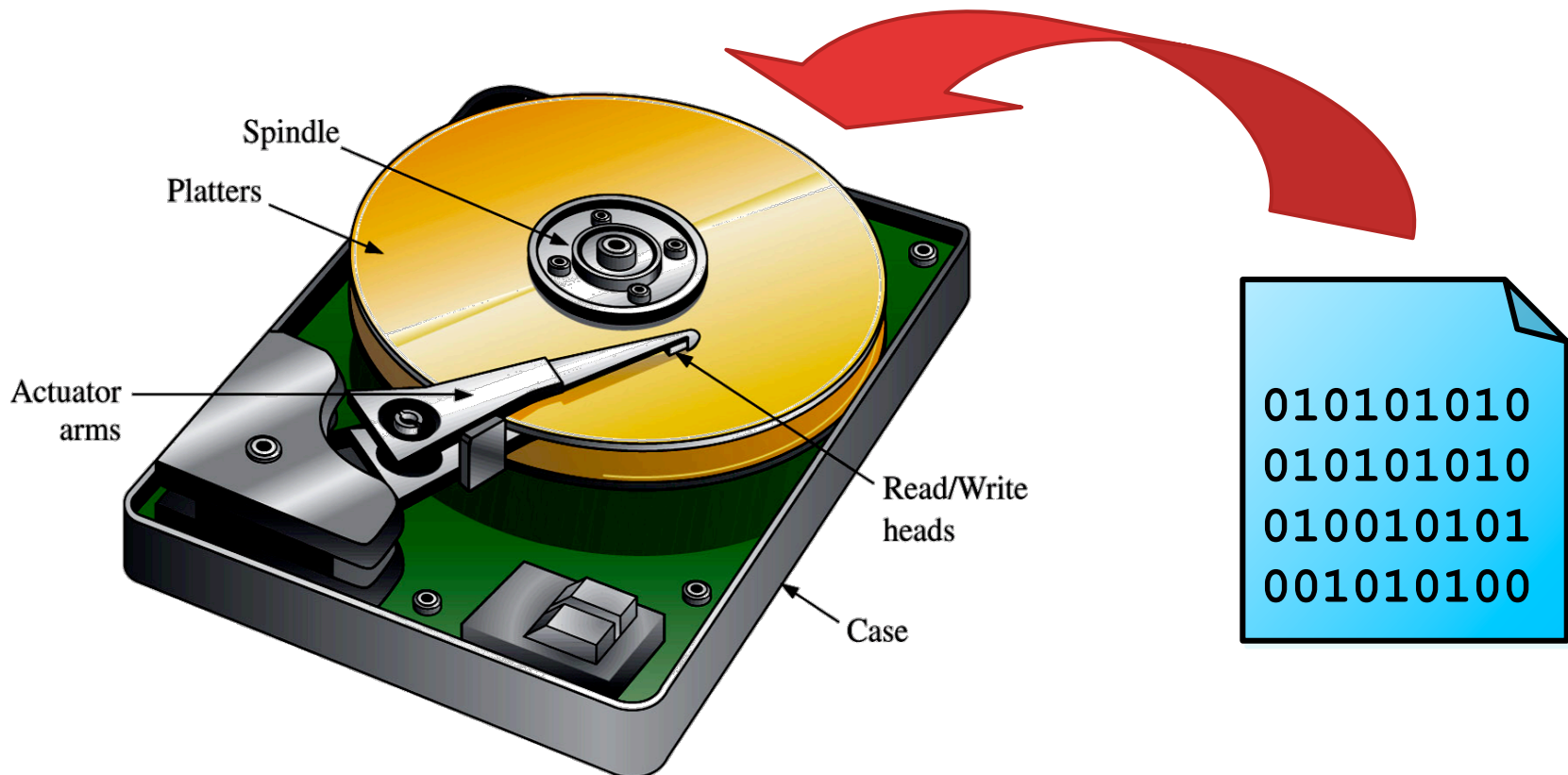
## :: Motivação

- Nem todos os problemas dependem unicamente de dados fornecidos via teclado.
- Alguns problemas requerem manipulação de dados armazenados em **arquivos**.
  - ▣ Exemplo: leituras de uma sonda, medições de um equipamento.
- Além disso, os dados armazenados nas variáveis de um script são **temporários**: eles são **perdidos** quando o programa termina.

# Arquivos

## :: Características

- Um arquivo (*file*) é uma **sequência de bytes** que reside em uma área de armazenamento (Ex: disco rígido, *pen drive*, CD/DVD).



# Arquivos

## :: Características

- Arquivos são **identificados** por um nome.
- O nome de um arquivo pode ter uma **extensão** que indica o tipo do conteúdo do arquivo.

**arquivo.ext**

- Podem armazenar **grande quantidade** de informação (não têm tamanho preestabelecido).
- Dados são **persistentes**: podem ser recuperados após interrupção do fornecimento de energia.

# Arquivos

## :: Tipos

### Arquivo texto

- Coleção de **caracteres** que podem ser mostrados diretamente na tela ou modificados por editores de texto simples.
- **Exemplos:** código Python, texto simples, etc.

### Arquivo binário

- Sequência de **bits** sujeita às convenções do programa de origem, não legível diretamente.
- **Exemplos:** arquivos executáveis ou compactados.



# Conteúdo



Introdução a arquivos



**Operações básicas**



Manipulando números

# Operações básicas



Abertura



Fechamento



Escrita



Leitura

# Abertura de arquivo

## :: Função **open**

- A abertura de uma arquivo **liga uma variável** do script ao espaço em disco ocupado pelo arquivo.
- A função `open` realiza essa operação, exigindo os seguintes parâmetros:
  - ▣ **Nome** e **extensão** do arquivo a ser aberto
  - ▣ **Modo** de operação (leitura ou escrita)

```
<variável> = open("arquivo.ext", <modo>)
```

# Abertura de arquivo

## :: Modos

Modo	Significado
<b>r</b>	Leitura apenas
<b>w</b>	Escrita, apagando o conteúdo existente
<b>a</b>	Escrita, acrescentando após conteúdo existente
<b>rb</b>	Leitura de arquivos binários
<b>wb</b>	Escrita de arquivos binários, apagando o conteúdo existente
<b>ab</b>	Escrita de arquivos binários, acrescentando após conteúdo existente

# Abertura de arquivo

## :: Exemplos

```
arq1 = open("números.txt", r)
```

Abre arquivo `numeros.txt` somente leitura (`r`) e guarda referência na variável `arq1`

```
arq2 = open("imagem.png", wb)
```

Abre o arquivo binário `imagem.png` para escrita com perda de conteúdo anterior (`wb`) e guarda referência na variável `arq2`

```
arq3 = open("medicoes.txt", a)
```

Abre o arquivo `medicoes.txt` para escrita sem perda de conteúdo anterior (`a`) e guarda referência na variável `arq3`

# Funções × Métodos

- Apenas a operação de abertura de arquivos utiliza uma **função** Python.
- As demais operações (fechamento, escrita, leitura) utilizam **métodos** Python, aplicados sobre o nome da variável que aponta para o local do arquivo.

# Fechamento de arquivo

## :: Método **close**

- Antes de encerrar o programa, certifique-se de **fechar** todos os arquivos abertos.
- Cada arquivo aberto **consome recursos** do computador.
- O fechamento de arquivos visa:
  - ▣ **Liberar** recursos computacionais (memória, processamento)
  - ▣ Preservar **integridade** do arquivo
  - ▣ Aplicar **alterações**

# Fechamento de arquivo

## :: Exemplos

```
arq1.close()
```

Fecha arquivo referenciado pela variável **arq1**

```
saida.close()
```

Fecha arquivo referenciado pela variável **saida**



# Escrita de dados

## :: Método **write**

- O método **write** escreve uma **string de caracteres** em um arquivo.
- Antes disso, o arquivo deve estar aberto em **modo de escrita** (**w** ou **a**).

```
<arquivo>.write("Eu amo Python.\n")
```

# Escrita de dados

## :: Exemplo com **strings**

```
# Abre arquivo
```

```
arq = open("filosofos.txt", "w")
```

```
arq.write("Socrates\n")
```

```
arq.write("Platao\n")
```

```
arq.write("Aristoteles\n")
```

```
# Fecha arquivo
```

```
arq.close()
```



# Escrita de dados

## :: Exemplo com **lista**

```
# Abre arquivo
arq = open("filosofos2.txt", "w")

flst = ["Socrates", "Platao", "Aristoteles"]

for i in range(len(flst)):
    arq.write(flst[i] + "\n")

# Fecha arquivo
arq.close()
```



# Leitura de dados

## :: Métodos

- Método **readline**

- ▣ Lê **uma única** linha do arquivo aberto e guarda-a em uma variável do tipo **string**.

- Método **read**

- ▣ Lê **todas** as linhas do arquivo aberto e guarda-as em uma variável do tipo **string**.

- Método **readlines**

- ▣ Lê **todas** as linhas do arquivo aberto e guarda-as em uma **lista** de strings.

# Leitura de dados

## :: Método **readlines**

```
# Abre arquivo
arq = open("filosofos.txt", "r")

for linha in arq.readlines():
    print(linha)

# Fecha arquivo
arq.close()
```

Escreve todas as linhas do  
arquivo na saída



# Leitura de dados

## :: Método **readline**

```
# Abre arquivo
arq = open("filosofos.txt", "r")

for linha in arq.readline():
    print(linha)

# Fecha arquivo
arq.close()
```

Escreve apenas os caracteres da primeira linha do arquivo



# Leitura de dados

## :: Método **read**

```
# Abre arquivo
arq = open("filosofos.txt", "r")

for linha in arq.read():
    print(linha)

# Fecha arquivo
arq.close()
```

Escreve caracteres de cada  
linha do arquivo



# Conteúdo



Introdução a arquivos



Operações básicas



Manipulando números



# Escrevendo dados numéricos

- Para escrever números em um arquivo, você deve primeiro **convertê-los em strings**, para depois usar o método **write** para escrevê-los no arquivo.

# Escrevendo dados numéricos

## :: Exemplo 01

```
# Abre arquivo
arq = open("numeros.txt", "w")

for linha in range(100):
    arq.write(str(linha) + "\n")

# Fecha arquivo
arq.close()
```

Converte números  
em strings



# Escrevendo dados numéricos

## :: Exemplo 02

```
# Abre arquivo
from random import *

arq = open("numeros_rand.txt", "w")

for i in range(20):
    arq.write(str(randint(11, 99)) + " ")

# Fecha arquivo
arq.close()
```



# Lendo dados numéricos

- Quando um arquivo contém diversos números, estes são lidos como uma **string única** de caracteres.
- Antes de converter, é necessário **separar** (*split*, em inglês) os números que estão unidos na string por meio de espaços em branco, como " " ou **\n**.

# Lendo dados numéricos

## :: Exemplo

```
# Abre arquivo
infile = open("numeros_rand.txt", "r")

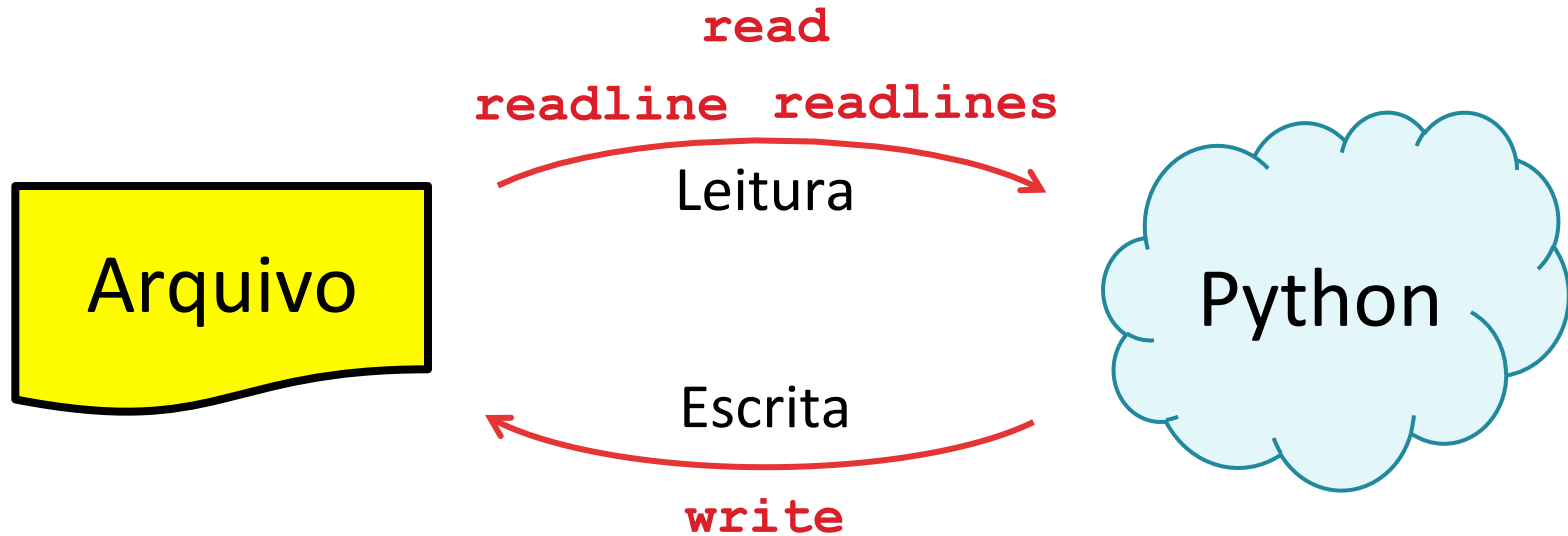
# Guarda todo o arquivo em uma string
s = infile.read()

# Toma cada conjunto de caracteres separados por
# espaços e os transforma em numeros
for x in s.split():
    print(int(x))

# Fecha arquivo
infile.close()
```





# Resumo



`open()` `split` `close`

# Referências bibliográficas

-  □ Menezes, Nilo Ney Coutinho (2010). **Introdução à Programação com Python**. Editora Novatec.
-  □ HETLAND, Magnus Lie (2008). **Beginning Python: From Novice to Professional**. Springer eBooks, 2ª edição. Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0634-7>.
- Horstmann, Cay & Necaise, Rance D. (2013). **Python for Everyone**. John Wiley & Sons.
- Liang, Y. D. (2013). **Introduction to Programming Using Python**. Pearson

Dúvidas?

