



INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL



Cofinanciado pela  
União Europeia

Os Fundos Europeus mais próximos de si.



REPÚBLICA  
PORTUGUESA

TRABALHO, SOLIDARIEDADE  
E SEGURANÇA SOCIAL

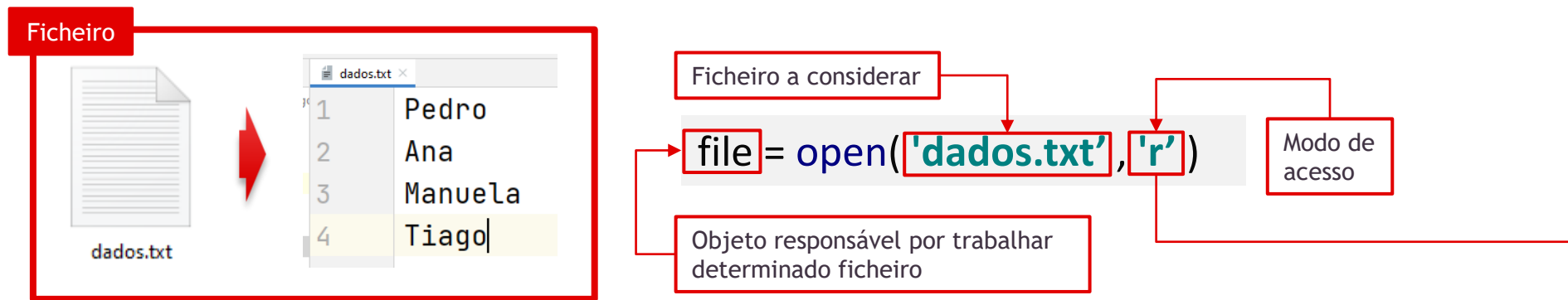


UFCD 10794 - Programação avançada com  
Python

# Ficheiros

# Ficheiros

Trabalhar com ficheiros no Python (abertura de ficheiros):



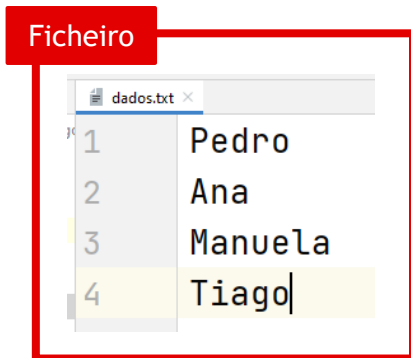
Características:	Modos:					
	r	r+	w	w+	a	a+
Leitura	x	x		x		x
Escrita		x	x	x	x	x
Caso o ficheiro não exista	erro	erro	cria	cria	cria	cria
Caso o ficheiro exista			limpa	limpa		
Ao escrever	erro	sobrepõe	substitui	substitui	acrescenta	acrescenta
Posiciona-se	início	início	início	início	fim	fim

# Ficheiros

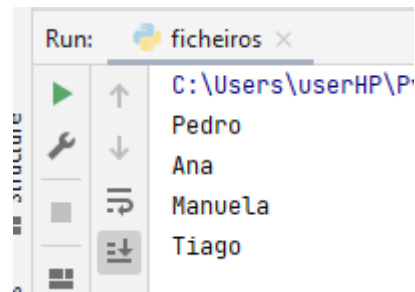
Leitura

# Ficheiros

Ler um ficheiro integralmente (como string) - **cuidado com o tamanho do ficheiro!!!**:



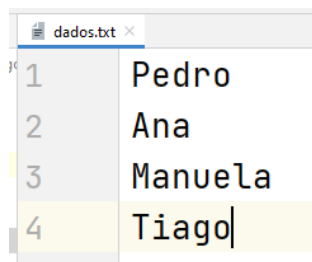
```
file = open('dados.txt', 'r')  
nomes = file.read()  
print(nomes)
```



# Ficheiros

Ler os próximos n carateres:

Ficheiro



1 Pedro  
2 Ana  
3 Manuela  
4 Tiago

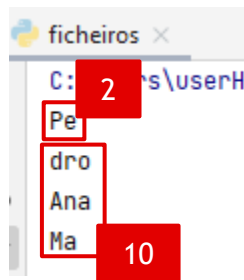
```
file = open('dados.txt', 'r')
```

```
print(file.read(2))
```

Lê os primeiros 2 carateres

```
print(file.read(10))
```

Lê os seguintes 10 carateres



2  
Pe  
dro  
Ana  
Ma

10



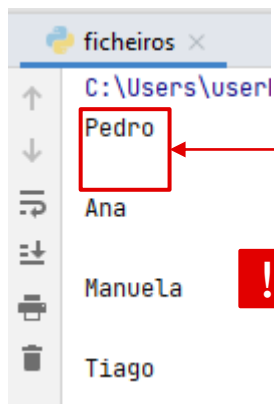
A cada leitura o cursor reposiciona-se para que a próxima leitura parta dele.

# Ficheiros

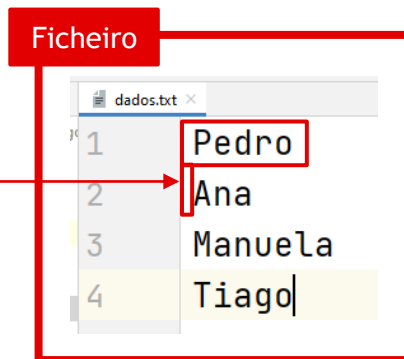
Obter separadamente todas as linhas de um ficheiro (**cuidado com o tamanho do ficheiro!!!**):

```
file = open('dados.txt', 'r')
linhas = file.readlines()

for item in linhas:
    print (item)
```



! No ficheiro original cada linha tem um parágrafo, é normal que esse parágrafo venha em conjunto com os dados finais.

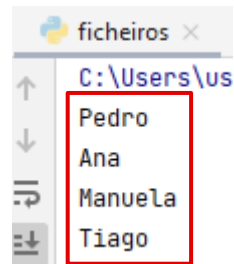


# Ficheiros

Para limparmos o parágrafo que fica a mais:

```
file = open('dados.txt', 'r')  
linhas = file.readlines()  
  
for item in linhas:  
    print ( item.strip() )
```

Remove os espaços em branco do lado esquerdo ou direito da string (inclui os parágrafos)



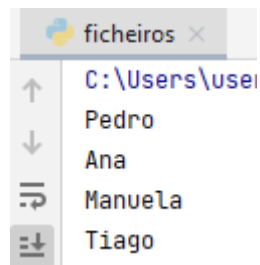


# Ficheiros

Trabalhar a leitura de um ficheiro de forma otimizada.

Permite faseadamente carregar do ficheiro linha a linha.

```
file = open('dados.txt', 'r')  
linha = file.readline()  
  
while linha:  
    print (linha.strip())  
    linha = file.readline()
```



# Ficheiros

Escrita

# Ficheiros

Escrever diretamente no ficheiro:

```
file = open('dados.txt', 'w')  
file.write('Ricardo')  
file.write('Jorge')  
file.write('Francisco')
```

**Antes**

Linha	Conteúdo
1	Pedro
2	Ana
3	Manuela
4	Tiago

**Depois**

Linha	Conteúdo
1	RicardoJorgeFrancisco

**!** Modo “w” apaga todo o conteúdo e começa a escrever do zero!  
Não existem parágrafos porque não foram indicados!

# Ficheiros

Escrever um ficheiro com parágrafos:

```
file = open('dados.txt', 'w')  
file.write('Ricardo\n')  
file.write('Jorge\n')  
file.write('Francisco\n')
```



dados.txt	
1	Ricardo
2	Jorge
3	Francisco

# Ficheiros

Posicionar o ponteiro

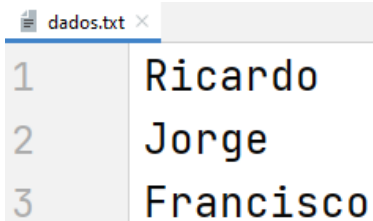
# Ficheiros

Para posicionar o ponteiro:

Serve tanto para leitura como para escrita.

O parágrafo conta como um carater!

Ficheiro

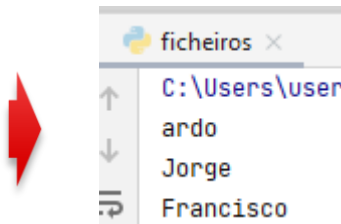


dados.txt x

1	Ricardo
2	Jorge
3	Francisco

```
file = open('dados.txt', 'r')  
file.seek(3)  
print(file.read())
```

Orientar a posição (inclusa) do ponteiro para um local específico.  
0 = início!



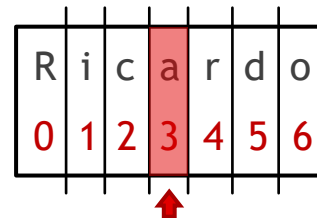
ficheiros x

C:\Users\user

ardo

Jorge

Francisco



R	i	c	a	r	d	o
0	1	2	3	4	5	6

# Ficheiros

Para posicionar o ponteiro em diferentes modos:

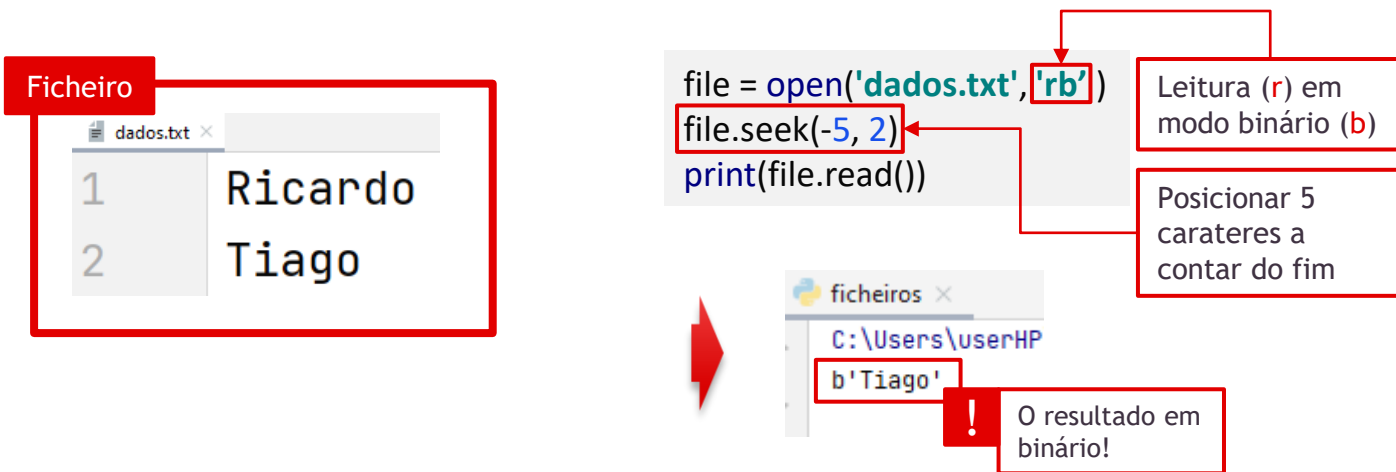
```
file = open('dados.txt', 'r')  
file.seek(0, 2)  
print(file.read())
```

Ponto de referência

Modos	Descrição
0	Início do ficheiro
1	Atual
2	Fim do ficheiro

# Ficheiros

Nota: para posicionar a contar do fim (no modo 2) **é necessário abrir o ficheiro em modo binário!**



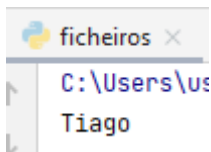


# Ficheiros

Caso necessário: é possível converter um binário para texto:

```
file = open('dados.txt', 'rb')  
file.seek(-5, 2)  
print(file.read().decode("utf-8"))
```

Conversão do  
binário para texto



# Ficheiros

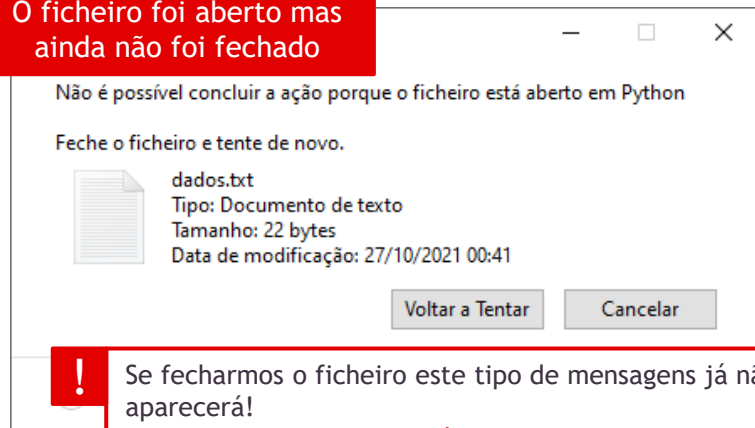
Fechar os ficheiros

# Ficheiros

Se abrirmos um ficheiro sem o fechar, mais nenhum programa lhe poderá ter acesso!

Os ficheiros devem sempre ser fechados quando já não precisamos mais deles!

O ficheiro foi aberto mas  
ainda não foi fechado



Se fecharmos o ficheiro este tipo de mensagens já não aparecerá!

Nota: quando o programa é encerrado, os ficheiros que estavam em uso no mesmo, ficam automaticamente libertos.

```
file = open('dados.txt', 'r')  
print(file.read())  
file.close()
```

Liberta o ficheiro para que outros  
programas o possam utilizar!

# Ficheiros

## Exercícios

# Ficheiros

## Exercício 1

Crie um programa que permita registar convidados num ficheiro.

Importante:

- Os seus convidados deverão estar registados no ficheiro `lista.txt`
- Cada vez que o programa for chamado, os convidados deverão ser acrescentados à lista já existente (separar por parágrafos cada indivíduo)
- O utilizador indica o nome dos convidados que pretende.
- O programa deverá parar de registar novos convidados quando for indicado um espaço em branco.
- No fim de registar, apresente a lista completa de convidados.

► Cábula:

```
file = open('dados.txt', 'w')
file.write('Ricardo')
file.write('Jorge')
file.write('Francisco')
```

```
file = open('dados.txt', 'r')
linha = file.readline()
```

```
while linha:
    print (linha.strip())
    linha = file.readline()
```

Características:	Modos:					
	r	r+	w	w+	a	a+
Leitura	x	x		x		x
Escrita		x	x	x	x	x
Caso não exista o ficheiro indicado	erro	erro	cria	cria	cria	cria
Sobrepõe (caso exista)		x				
Posiciona-se	início	início	início	início	fim	fim

# Ficheiros

## Exercício 2

Aproveitando o ficheiro de convidados gerado pelo exercício anterior:

Crie um novo programa que permita verificar se determinada pessoa (indicada pelo utilizador) existe ou não na lista de convidados.

Caso exista: Indique a posição em que se encontra na lista e a quantidade de convidados total, **exemplo:**

Convidado existe na lista, na posição 23/150

Assuma que “23” = posição e “150” = total de convidados

► Cábula:

```
file = open('dados.txt', 'w')
file.write('Ricardo')
file.write('Jorge')
file.write('Francisco')
```

```
file = open('dados.txt', 'r')
linha = file.readline()
```

```
while linha:
    print (linha.strip())
    linha = file.readline()
```

Características:	Modos:					
	r	r+	w	w+	a	a+
Leitura	x	x		x		x
Escrita		x	x	x	x	x
Caso não exista o ficheiro indicado	erro	erro	cria	cria	cria	cria
Sobrepõe (caso exista)		x				
Posiciona-se	início	início	início	início	fim	fim