

# Computação II - Python

## Aula 6 - Persistência de Dados

Carla A. D. M. Delgado

João C. P. da Silva

Dept. Ciência da Computação - UFRJ

# Persistência de dados

## Persistência de dados - como usar amanhã os dados que você está gerando hoje...

- Quando a execução de um programa termina, todos os dados que ele manipulou e ficaram na memória são perdidos.
- Quanto queremos armazenar os dados para uso futuro, temos que trabalhar com mecanismos de **persistência de dados**.
- *Arquivos* são o mecanismo mais elementar e também o mais tradicional de persistência de dados em programação.

# Persistência de dados - Arquivos

**Arquivos são sequências de dados logicamente relacionados que podem ser usados por programas.**

- Arquivos são armazenados na memória de longa duração (disco, pen drive, etc).
- Seguem um "protocolo" de uso:
  - Para trazer um arquivo para a memória de trabalho do computador, deve ser usado o **comando de abertura do arquivo**;
  - Ao terminar de usar um arquivo, deve ser usado o **comando de fechar arquivo**, indicando que este pode ser copiado para o disco novamente.
- O processo de trazer para a memória de trabalho e copiar para o disco é complexo. Porém isso é gerenciado pelo compilador ou interpretador da linguagem de programação.
- O arquivo pode ser corrompido caso o protocolo não seja seguido, ocasionando perda de dados :-)

# Persistência de dados - Arquivos

## Abrindo um arquivo : `open`

- O método **`open`** recebe como parâmetro o nome do arquivo que se deseja abrir e o modo em que o arquivo deve ser aberto ("`r`" de *read* para leitura, "`w`" de *write* para escrita)
- Caso o arquivo não seja encontrado a partir do diretório corrente onde o interpretador está sendo executado, será criado um novo arquivo vazio com esse nome.

```
1 In [0]: f = open('newfile.txt', 'w')
```

- o arquivo '`newfile.txt`' não existia e foi criado no diretório corrente. Após ser criado, ele foi aberto para escrita e associado à variável `f`.

# Persistência de dados - Arquivos

## Escrevendo no arquivo: write

- Usamos o método **write** para escrever em um arquivo. Será retornado o número de caracteres escrito.
- Só é possível escrever em um arquivo previamente aberto!

```
1 In [2]: f.write('Escrever ou nao escrever\n') # '\n' gera uma quebra de linha
2 Out[2]: 25
3
4 In [3]: f.write('Eis a questao')
5 Out[3]: 13
```

- Só dados do tipo string podem ser escritos em arquivos de texto. Portanto, converta seus dados para string ao escrever.
- As operações de escrita não são salvas no disco imediatamente. Caso o arquivo seja aberto por outro programa (notepad por exemplo), ele ainda pode estar em branco mesmo após uma operação de escrita ter sido feita.
- Quando o arquivo é fechado, todas as operações de escrita que estiverem pendentes serão salvas no disco.

# Persistência de dados - Arquivos

## Escrevendo no arquivo: write

- Usamos o método **close** para fechar um arquivo.

```
1 In [0]: f = open('newfile.txt', 'w')
2
3 In [1]: pwd
4 Out[1]: 'C:\\Users\\Carla\\Documents\\Ipea-Rio\\teste'
5
6 In [2]: f.write('Escrever ou nao escrever\\n') # '\\n' gera uma quebra de linha
7 Out[2]: 25
8
9 In [3]: f.write('Eis a questao')
10 Out[3]: 13
11
12 In [4]: f.close()
```

- Quando o arquivo é fechado, todas as operações de escrita que estiverem pendentes serão salvas no disco.
- Depois de fechado, o arquivo fica indisponível para uso no Python até ser aberto novamente.

# Persistência de dados - Arquivos

## Lendo um arquivo: read

- Usamos o método **read** para ler dados de um arquivo que foi previamente aberto para leitura.
- Para abrir um arquivo para leitura, precisamos dar o nome de um arquivo **existente** e indicar o modo de leitura **r** no momento da abertura.
- Caso o arquivo não esteja no diretório corrente, podemos indicar o diretório passando o caminho completo do diretório corrente até ele.

```
1 In [1]: f = open('caminho_completo/newfile.txt', 'rw')
2
3 In [2]: infos = f.read()
4
5 In [3]: infos
6 Out[3]: 'Escrever ou nao escrever\nEis a questao'
7
8 In [4]: print(infos)
9 Escrever ou nao escrever
10 Eis a questao
```

# Persistência de dados - Arquivos

## Retornando ao início de um arquivo: seek

- O acesso aos dados de um arquivo é sequencial. Após realizar o comando `read`, chegamos ao final do arquivo. Se quisermos ler o arquivo novamente, temos que retornar ao seu início. Para isso, usamos o método **seek**
- Como método `seek` podemos indicar para qual posição do arquivo queremos ir. Para ir ao início, indicamos a posição 0.

```
1 In [5]: f.seek(0)
2 Out[5]: 0
```



# Persistência de dados - Arquivos

## Leitura de arquivo linha a linha

- Podemos usar um loop para ler um arquivo linha a linha:

```
1 In [6]: for line in f:  
2         print(line)  
3 Escrever ou não escrever  
4  
5 Eis a questão
```

- a string correspondente a uma linha será atribuída à variável *line*, a cada iteração do comando *for*

# Persistência de dados - Arquivos

## Fechando arquivos

- Não esqueça de fechar o arquivo quando acabar de utilizá-lo!

```
1 In [0]: f = open('newfile.txt', 'r')
2
3 In [1]: f = open('caminho_completo/newfile.txt', 'rw')
4
5 In [2]: infos = f.read()
6
7 In [3]: infos
8 Out[3]: 'Escrever ou nao escrever\nEis a questao'
9
10 In [4]: print(infos)
11 Escrever ou nao escrever
12 Eis a questao
13
14 In [5]: f.seek(0)
15 Out[5]: 0
16
17 In [6]: for line in f:
18     print(line)
19 Escrever ou nao escrever
20
21 Eis a questao
22
23 In [7]: f.close()
```

# Persistência de dados - Arquivos

## Exercício

Crie como o auxílio de um editor de texto ascii (bloco de notas ou notepad++, por exemplo) um arquivo contendo um número par em cada linha, e outro arquivo contendo um número ímpar em cada linha. Os números nestes arquivos podem estar fora de ordem. Agora, escreva uma função que receba como entrada o nome destes arquivos, e crie um outro arquivo que tenha todas as linhas dos outros arquivos, porém os números devem estar ordenados do menor para o maior. O nome do arquivo a ser criado deve ser uma composição dos nomes dos arquivos de entrada (por exemplo, se os arquivos de entrada foram `arq1.txt` e `arq2.txt`, o nome do arquivo a ser criado deve ser `arq1arq2.txt`). Sua função deve retornar o nome do arquivo criado.

# Persistência de dados - Arquivos

- 1 Faça o programa do jogo: “Acerte os números”. Inicialmente, leia diversos números inteiros entrados pelo dono da casa de aposta, guardando numa lista. A leitura termina quando o dono digitar um número negativo. Em seguida o apostador poderá entrar com vários números para verificar o seu acerto.  
Caso acerte um dos números da lista, informe do acerto e diga quantos números faltam acertar. O programa termina caso o apostador erre 10 vezes ou até ele acertar todos os números.  
Obs.: caso o dono da casa de apostas tenha entrado com um número que já exista, ignore-o.
- 2 Como o programa acima deveria ser modificado para que as informações de entrada pudessem ser fornecidas via um arquivo?

# Computação II - Python

## Aula 6 - Persistência de Dados

Carla A. D. M. Delgado

João C. P. da Silva

Dept. Ciência da Computação - UFRJ